Smarter. Greener. Together.

Штаб промислової автоматизації

Delta Electronics, Inc.
Технологічний центр Таоюань
No.18, Xinglong Rd., Taoyuan District,
Taoyuan City 33068, Тайвань
ТЕЛ: 886-3-362-6301 / ФАКС: 886-3-371-6301

Азії

Delta Electronics (Shanghai) Co., Ltd.
No.182 Minyu Rd., Pudong Shanghai, КНР
Поштовий індекс: 201209
ТЕЛ: 86-21-6872-3988 / ФАКС: 86-21-6872-3996
Служба підтримки клієнтів: 400-820-9595

Delta Electronics (Японія), Inc.
Занадто□□□□
Відділ продажів промислової автоматизації 2-
1-14 Shibadaimon, Minato-ku
Токіо, Японія 105-0012
ТЕЛ: 81-3-5733-1155 / ФАКС: 81-3-5733-1255

Delta Electronics (Korea), Inc.
□□
1511, 219, Gasan Digital 1-Ro., Geumcheon-gu, Seoul, 08501
Південна Корея
ТЕЛ: 82-2-515-5305 / ФАКС: 82-2-515-5302

Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, Сінгапур 417939
ТЕЛ: 65-6747-5155 / ФАКС: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No.43, Sector 35, HSIIDC Gurgaon,
PIN 122001, Haryana, India
ТЕЛ: 91-124-4874900 / ФАКС: 91-124-4874945

Delta Electronics (Таїланд) PCL.
909 Soi 9, Moo 4, Bangpoo Industrial Estate (EPZ),
Pattana 1 Rd., T.Phraksa, A.Muang,
Samutprakarn 10280, Таїланд
ТЕЛ: 66-2709-2800/ФАКС: 662-709-2827

Delta Electronics (Австралія) Pty.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168,
Австралія
ТЕЛ.: 61-3-9543-3720

Америки

Delta Electronics (America) Ltd.
0000 □□
PO Box 12173, 5101 Davis Drive, Research
Triangle Park, NC 27709, USA ТЕЛ.: 1-919-
767-3813 / ФАКС: 1-919-767-3969

Delta Greentech (Бразилія) S/A
□□□□□ □□
Rua Itapeva, 26 – 3-й поверх – Bela Vista
CEP: 01332-000 – São Paulo – SP – Brazil
TEL: 55-11-3530-8643 / 55-11-3530-8640

Delta Electronics International Mexico SA de CV
□□□□□□□
000□00000000□000000
Colonia La Loma, CP 54060
Tlalnepantla, State of Mexico
ТЕЛ.: 52-55-3603-9200

RTS UKRAINE

Дистриб'ютор в Україні

Україна: ТОВ "Системи реального часу - Україна"
www.delta-electronics.com.ua
вул. Святослава Хороброго, 29-А, 49001, м.Дніпро
Пошта: sales@rts.ua
ТЕЛ : +38 0562 392223 / +38 068 2392223

EMEA

Штаб-квартира: Delta Electronics (Нідерланди) BV
Продажі: Sales.IA.EMEA@deltaww.com
Маркетинг: Marketing.IA.EMEA@deltaww.com
Технічна підтримка: iatechnicalsupport@deltaww.com
Підтримка клієнтів: Customer-Support@deltaww.com
Сервіс: Service.IA.emea@deltaww.com
ТЕЛ.: +31(0)40 800 3900

BENELUX: Delta Electronics (Нідерланди) BV
De Witbogt 20, 5652 AG Eindhoven, The Netherlands Mail:
Sales.IA.Benelux@deltaww.com
ТЕЛ.: +31(0)40 800 3900

DACH: Delta Electronics (Нідерланди) BV
Coesterweg 45, D-59494 Soest, Німеччина
Пошта: Sales.IA.DACH@deltaww.com ТЕЛ.: +49(0)2921 987 0

Франція: Delta Electronics (France) SA
ZI du bois Challand 2, 15 rue des Pyrénées, Lisses, 91090
Evry Cedex, France
Пошта: Sales.IA.FR@deltaww.com ТЕЛ.:
+33(0)1 69 77 82 60

Iberia: Delta Electronics Solutions (Іспанія) SLU
Ctra. Від Вільяверде до Вальекаса, 265 1-е праворуч ред.
Hormigueras – PI de Vallecas 28031 Madrid TEL: +34(0)91 223 74 20
Carrer Llacuna 166, 08018 Барселона, Іспанія Пошта:
Sales.IA.Iberia@deltaww.com

Італія: Delta Electronics (Italy) Srl
Via Meda 2–22060 Novedrate(CO) Piazza Grazioli 18
00186 Roma Italy Пошта:
Sales.IA.Italy@deltaww.com
ТЕЛ: +39 039 8900365

Туреччина: Delta Greentech Elektronik San. Ltd. Sti. (Туреччина)
Şerifali Mah. Hendem Cad. Kule Сок. No:16-A
34775 Ümraniye – Стамбул
Пошта: Sales.IA.Turkey@deltaww.com ТЕЛ.: + 90 216 499 9910

GCC: Delta Energy Systems AG (Дубай BR)
PO Box 185668, Gate 7, 3rd Floor, Hamarain Center
Дубай, Об'єднані Арабські Емірати
Пошта: Sales.IA.MEA@deltaww.com ТЕЛ.:
+971(0)4 2690148

Єгипет + Північна Африка: Delta Electronics
318, 3-й поверх, БЦ «Тривіум», вул.Північна, 90, вул.
Новий Каїр, Каїр, Єгипет
Пошта: Sales.IA.MEA@deltaww.com

*Ми залишаємо за собою право змінювати інформацію в цьому документі без попереднього повідомлення.

Керівництво з експлуатації контролера руху серії AS500E



Серія AS500E Керівництво по експлуатації контролера руху

AC-0242620-02
25 грудня 2020 р
delta-electronics.com.ua

Smarter. Greener. Together.

Керівництво з експлуатації контролера руху серії AS500E

Зміст

Розділ 1

Передмова..... 1-1
1.1 Explanation of Symbols in This Manual..... 1-2
1.2 Revision History 1-2
Розділ 2 Огляд..... 2-1
2.1 Product Description..... 2-2
2.2 Functions..... 2-2
2.3 Profile and Components 2-3
Розділ 3 Технічні характеристики..... 3-1
3.1 Function Specifications..... 3-2
3.1.1 Технічні характеристики 3-2
3.1.2 Пристрої та типи даних 3-3
3.1.2.1 Пристрої 3-3
3.1.2.2 Дійсні діапазони пристроїв 3-4
3.1.2.3 Блоковані пристрої 3-5
3.1.2.4 Підтримувані типи даних і дійсні діапазони 3-6
3.2 Electrical Specifications..... 3-7
Розділ 4 Архітектура системи 4-1
4.1 System Constitution 4-2
4.2 Introduction to EtherCAT Fieldbus..... 4-2
4.2.1 Особливості EtherCAT Fieldbus 4-2
4.2.2 Зв’язок EtherCAT між контролером і підлеглими 4-3
4.2.3 Ініціалізація мережі EtherCAT 4-4
4.2.4 Як усунути невідповідність кількості між конфігурацією EtherCAT і фактично підключеними пристроями 4-4
4.2.5 Підлеглий EtherCAT в автономному режимі та механізм відновлення 4-4
4.2.6 Сервоприводи, які можна підключити до порту EtherCAT 4-5
4.3 Power Supply 4-7
4.4 Right-side Extension 4-7
4.4.1 Праві модулі розширення, які підключаються 4-7
4.4.2 Призначення адрес правого модуля розширення 4-9
4.5 SD Memory Card..... 4-9

4.5.1 Модель і технічні характеристики 4-9

Розділ 5 Встановлення 5-1

5.1 Dimensions 5-2

5.1.1 Профіль і розміри 5-2

5.1.2 Розміри правого модуля розширення 5-2

5.1.3 Підключення до правого модуля розширення 5-3

5.1.4 Встановлення та виймання SD-карти 5-4

5.1.5 Встановлення та заміна Кнопкова батарея 5-5

5.2 Installing the Module in the Control Cabinet 5-8

5.2.1 Встановлення модуля на DIN-рейку 5-8

5.2.2 Ілюстрація встановлення всередині шафи керування 5-8

5.2.3 Температура навколишнього середовища в шафі керування 5-9

5.2.4 Дії для захисту від перешкод 5-9

5.2.5 Вимоги до розмірів у шафі керування 5-9

Розділ 6 Електропроводка, налаштування COM і побудова мережі.....6-1

6.1 Wiring 6-3

6.1.1 Джерело живлення 6-3

6.1.2 Електропроводка ланцюга безпеки..... 6-4

6.2 Input Point and Output Point Wiring.....6-5

6.2.1 Функції, які підтримують точки введення 6-5

6.2.2 Електропроводка вхідної точки 6-7

6.2.3 Електропроводка вихідної точки 6-9

6.3 RS-485 Communication Port 6-10

6.3.1 Функція, яку підтримує порт RS-485 6-10

6.3.2 Визначення контактів порту RS-485 6-10

6.3.3 Апаратне підключення RS-485 6-10

6.3.4 Підтримувані функціональні коди та виняткові коди 6-11

6.4 RS-232 Communication Port 6-12

6.4.1 Функція, яку підтримує порт RS-232 6-12

6.4.2 Визначення контактів порту RS-232 6-13

6.4.3 Апаратне підключення RS-232 6-13

6.4.4 Підтримувані функціональні коди та коди винятків 6-13

6.5 SSI Absolute Encoder Port 6-14

6.5.1 Функція абсолютного кодера SSI 6-14

6.5.2 Визначення контактів порту SSI 6-15

6.5.3 Апаратне підключення абсолютного кодера SSI 6-15

6.6 Incremental Encoder Port 6-16

6.6.1 Функція інкрементального кодера 6-16

6.6.2 Визначення контактів порту інкрементального енкодера 6-16

6.6.3 Апаратне підключення інкрементального енкодера 6-18

6.7 Ethernet Communication Port..... 6-19
6.7.1 Функція, яку підтримує порт зв'язку Ethernet..... 6-19
6.7.2 контактів порту зв'язку Ethernet 6-20
6.7.3 Підключення до мережі порту зв'язку Ethernet 6-20
6.7.4 Коди функцій, які підтримує порт зв'язку Ethernet 6-20

6.8 EtherCAT Communication Port..... 6-21
6.8.1 Функція, яку підтримує комунікаційний порт EtherCAT 6-21
6.8.2 контактів комунікаційного порту EtherCAT 6-21
6.8.3 Мережеве з'єднання комунікаційного порту EtherCAT 6-22
6.8.4 Відстань зв'язку EtherCAT 6-23

6.9 CANopen Communication Port..... 6-23
6.9.1 Функції, які підтримує комунікаційний порт CANopen 6-23
6.9.2 Контакти комунікаційного порту CANopen 6-24
6.9.3 Відображення PDO на комунікаційному порту CANopen 6-24
6.9.4 Підключення до мережі на Комунікаційний порт CANopen 6-24
6.9.5 Швидкість зв'язку CANopen і відстань зв'язку 6-25

Розділ 7 Принцип виконання 7-1

7.1 Tasks..... 7-2
7.1.1 Типи завдань 7-2
7.1.2 Рівні пріоритету завдань 7-4
7.1.3 Сторожовий пес для завдання 7-6
7.1.4 Інструкції щодо руху та спілкування для кожного типу завдань 7-7

7.2 The Impact of PLC RUN or STOP on Variables and Devices..... 7-10

7.3 Relationship between Motion Program and Motion Bus..... 7-10

7.4 Synchronization Cycle Period Setting 7-11

Розділ 8 Логічні інструкції 8-1

8.1 Table of Logic Instructions..... 8-6

8.2 Explanation of Logic Instructions..... 8-11
8.2.1 EN та ENO 8-11

8.3 Sequence Input and Output Instructions..... 8-11
8.3.1 R_TRIG 8-11
8.3.2 F_TRIG 8-13
8.3.3 RS 8-15
8.3.4 CP 8-17
8.3.5 SEMA 8-19

8.4 Sequence Control Instructions 8-21
8.4.1 JMP..... 8-21

8.5 Data Movement Instructions 8-22
8.5.1 MOVE 8-22
8.5.2 MoveBit 8-24
8.5.3 TransBit 8-26
8.5.4 MoveDigit 8-28
8.5.5 Exchange 8-30
8.5.6 Swap 8-32
8.6 Comparison Instructions 8-34
8.6.1 LT 8-34
8.6.2 LE 8-36
8.6.3 GT 8-38
8.6.4 GE 8-40
8.6.5 EQ 8-42
8.6.6 NE 8-44
8.7 Timer Instructions 8-46
8.7.1 TON 8-46
8.7.2 TOF 8-48
8.7.3 TP 8-50
8.7.4 Sys_ReadTime 8-52
8.7.5 Sys_ReadTotalWorkTime 8-53
8.7.6 Sys_ReadPowerOnTime 8-54
8.7.7 Sys_WdgStatus 8-55
8.8 Counter Instructions 8-57
8.8.1 CTU 8-57
8.8.2 CTD 8-59
8.8.3 CTUD 8-61
8.9 Math Instructions 8-64
8.9.1 ADD 8-64
8.9.2 SUB 8-67
8.9.3 MUL 8-70
8.9.4 DIV 8-73
8.9.5 MOD 8-76
8.9.6 MODREAL 8-78
8.9.7 MODTURNS 8-80
8.9.8 MODABS 8-82
8.9.9 ABS 8-84
8.9.10 DegToRad 8-86
8.9.11 ReadToThem 8-88
8.9.12 SIN 8-90
8.9.13 COS 8-92
8.9.14 TAN 8-94
8.9.15 ASIN 8-96

8.9.16 ACOS 8-99
8.9.17 ATAN 8-101
8.9.18 LN 8-103
8.9.19 LOG 8-105
8.9.20 SQRT 8-107
8.9.21 EXP 8-109
8.9.22 EXPT 8-111
8.9.23 RAND 8-113
8.9.24 TRUNC 8-115
8.9.25 FLOOR 8-117
8.9.26 FRACTION 8-119
8.10 Bit String Instructions 8-121
8.10.1 AND 8-121
8.10.2 OR 8-124
8.10.3 NOT 8-127
8.10.4 XOR 8-129
8.10.5 XORN 8-132
8.11 Shift Instructions 8-135
8.11.1 SHL 8-135
8.11.2 SHR 8-137
8.11.3 ROL 8-139
8.11.4 ROR 8-141
8.12 Selection Instructions 8-143
8.12.1 MAX 8-143
8.12.2 MIN 8-145
8.12.3 SEL 8-147
8.12.4 MUX 8-149
8.12.5 LIMIT 8-152
8.12.6 BAND 8-154
8.12.7 ZONE 8-157
8.13 Data Type Conversion Instructions 8-160
8.13.1 BOOL_TO_*** 8-160
8.13.2 Бітові рядки_TO_*** 8-163
8.13.3 Цілі числа_TO_*** 8-170
8.13.4 Дійсні числа_TO_*** 8-179
8.13.5 Час, дата_TO_*** 8-182
8.13.6 Рядки_TO_*** 8-184
8.14 Communication Instructions 8-187
8.14.1 Інструкції зі зв’язку CANopen 8-187
8.14.1.1 DMC_ReadParameter_CANopen 8-187
8.14.1.2 DMC_WriteParameter_CANopen 8-192

8.14.2 Інструкції з Ethernet 8-197
8.14.2.1 ETH_Link_Config 8-197
8.14.2.2 ETH_Link_Manage 8-202
8.14.2.3 ETH_Link_Status 8-204
8.14.2.4 Приклад обміну даними MODBUS TCP 8-207
8.14.2.5 ETH_Link_Config_Ext 8-213
8.14.2.6 ETH_SetServerlinkkeeptime 8-215
8.14.2.7 ETH_Socket_Manage..... ..8-216
8.14.2.8 ETH_Socket_Config..... 8-218
8.14.2.9 ETH_Socket_Open 8-221
8.14.2.10 ETH_Socket_Send 8-223
8.14.2.11 ETH_Socket_Receive 8-227
8.14.2.12 ETH_Socket_Close 8-231
8.14.2.13 ETH_Socket_Status 8-233
8.14.2.14 Приклад вільного протоколу Ethernet 8-236
8.14.3 Інструкції щодо зв’язку RS485 8-241
8.14.3.1 RS485_Link_Manage 8-241
8.14.3.2 RS485_Link_Config 8-243
8.14.3.3 RS485_Link_Status 8-247
8.14.3.4 Приклад обміну даними RS485 8-250
8.14.3.5 RS485_RS 8-253
8.14.3.6 Приклад безкоштовного протоколу RS485 8-259
8.14.3.7 RS485_SetDelayTime 8-262
8.14.4 Інструкції щодо зв’язку RS232 8-264
8.14.4.1 RS232_Link_Manage 8-264
8.14.4.2 RS232_Link_Config 8-266
8.14.4.3 RS232_Link_Status 8-271
8.14.4.4 Приклад обміну даними RS232 8-274
8.14.4.5 RS232_RS 8-277
8.14.4.6 Приклад безкоштовного протоколу RS232 8-283
8.14.4.7 RS232_SetDelayTime 8-286
8.15 String Processing Instructions.....8-288
8.15.1 CONCAT..... 8-288
8.15.2 DELETE 8-290
8.15.3 INSERT 8-292
8.15.4 LEFT / RIGHT..... 8-294
8.15.5 MID 8-296
8.15.6 REPLACE 8-298
8.15.7 LEN 8-300
8.15.8 FIND 8-301
8.16 Immediate Refresh Instructions 8-303
8.16.1 FROM 8-303

8.16.2 TO 8-307
8.16.3 Негайний вхід 8-311
8.16.4 Миттєвий вихід 8-313
8.17 PID-related Instructions 8-315
8.17.1 PID 8-315
8.17.2 GPWM 8-326
8.18 Address Instruction 8-328
8.18.1 ADR 8-328
8.19 Network Diagnosis 8-330
8.19.1 Діагностика EtherCAT 8-330
8.19.1.1 EtherCAT_SysDiag 8-330
8.19.2 Діагностика CANopen 8-332
8.19.2.1 CANopen_SysDiag 8-332
8.19.2.2 CANopen_NodeDiag 8-334
8.19.2.3 CANopen_State 8-336
8.20 Read and Write Offset Bit Value 8-338
8.20.1 SetBitOffsetValue 8-338
8.20.2 GetBitOffsetValue 8-340
8.21 FCS Instructions 8-342
8.21.1 CRC16 8-342
8.21.2 LRC 8-344
8.22 Right-Side Extension Module Instructions 8-346
8.22.1 Інструкції, пов'язані з AS02PU та AS04PU 8-346
8.22.1.1 Умови застосування 8-346
8.22.1.2 Інструкції, пов'язані з AS02PU та AS04PU 8-347
Розділ 9 Вступ до параметрів осі 9-1
9.1 Description of Axis Parameters 9-2
Розділ 10 Функція керування рухом 10-1
10.1 EN and ENO 10-2
10.2 Relation among Velocity, Acceleration and Jerk 10-3
10.3 Introduction of BufferMode 10-6
10.4 The State Machine 10-32
Розділ 11 Інструкції з керування рухом 11-1
11.1 Table of Motion Control Instructions 11-4
11.2 About Motion Control Instructions 11-6
11.2.1 Склад інструкції керування рухом 11-6
11.2.2 Мови програмування, які підтримують інструкції керування рухом . 11-6

11.2.3 Конфігурація інструкцій керування рухом 11-6
11.3 Single-axis Instructions.....11-7
11.3.1 MC_Power 11-7
11.3.2 MC_Home 11-17
11.3.3 MC_MoveVelocity 11-22
11.3.4 MC_Halt 11-30
11.3.5 MC_Stop 11-35
11.3.6 MC_MoveRelative 11-40
11.3.7 MC_MoveAdditive 11-48
11.3.8 MC_MoveAbsolute 11-56
11.3.9 MC_MoveSuperimposed 11-65
11.3.10 MC_HaltSuperimposed 11-73
11.3.11 MC_SetPosition 11-78
11.3.12 MC_SetOverride 11-89
11.3.13 MC_Reset 11-93
11.3.14 DMC_SetTorque 11-96
11.3.15 MC_ReadAxisError 11-100
11.3.16 MC_ReadActualPosition 11-102
11.3.17 MC_ReadStatus 11-107
11.3.18 MC_ReadMotionState 11-112
11.3.19 DMC_ReadParameter_Motion 11-117
11.3.20 DMC_WriteParameter_Motion 11-122
11.3.21 DMC_TouchProbe 11-126
11.3.22 DMC_TouchProbeCyclically..... 11- 135
11.3.23 DMC_Jog 11-139
11.3.24 DMC_MoveVelocityStopByPos 11-142
11.3.25 DMC_MoveVelocityStopByLinePos 11-146
11.3.26 DMC_ReadPositionLagStatus 11-150
11.3.27 DMC_WritePositionLagSetting 11-153
11.3.28 DMC_ChangeMechanismGearRatio 11-155
11.3.29 DMC_TorqueControl 11-157
11.3.30 DMC_MoveVelocity 11-162
11.3.31 DMC_SwitchSoftLimit 11-163
11.4 Coupling Instructions.....11-165
11.4.1 MC_GearIn 11-165
11.4.2 MC_GearOut..... 11-172
11.4.3 MC_CombineAxes 11-177
11.4.4 Впровадження електронної камери 11-185
11.4.5 MC_CamIn 11-186
11.4.6 MC_CamOut 11-207
11.4.7 DMC_CamReadPoint 11-213
11.4.8 DMC_CamWritePoint 11-217

11.4.9 DMC_CamSet..... 11-219
11.4.10 DMC_CamReadTappetStatus 11-223
11.4.11 DMC_CamReadTappetValue..... 11-228
11.4.12 DMC_CamWriteTappetValue 11-231
11.4.13 DMC_CamAddTappet 11-235
11.4.14 DMC_CamDeleteTappet 11-239

11.5 Application Instructions 11-242
11.5.1 Технологія ротаційного різання 11-242
11.5.2 Параметри ротаційного різання 11-243
11.5.3 Функція керування функцією поворотного різання 11-244
11.5.4 Вступ до кривої кулачка з функцією обертового різання 11-245
11.5.5 Інструкції щодо ротаційного різання 11-249
11.5.5.1 APF_RotaryCut_Init 11-249
11.5.5.2 APF_RotaryCut_In 11-252
11.5.5.3 APF_RotaryCut_Out 11-254
11.5.6 Приклад застосування інструкцій з ротаційного різання 11-256

11.6 G Code Instructions..... 11-260
11.6.1 Введення в ЧПК 11-260
11.6.2 Формат введення коду G 11-260
11.6.3 Пояснення форматів коду G 11-261
11.6.4 G Функції коду 11-262
11.6.4.1 G90 (абсолютний режим) 11-262
11.6.4.2 G91 (відносний режим)..... 11-263
11.6.4.3 G0 (Швидке позиціонування) 11-264
11.6.4.4 G1 (лінійна інтерполяція) 11-267
11.6.4.5 G2 (Кругова/спіральна інтерполяція за годинниковою стрілкою)... 11-271
11.6.4.6 G3 (Кругова/спіральна інтерполяція проти годинникової стрілки).. 11-278
11.6.4.7 G17/G18/G19 (Укажіть кругову площину інтерполяції) 11-284
11.6.4.8 G4 (Інструкція про зупинку) 11-285
11.6.4.9 G50 (точна зупинка) 11-285
11.6.4.10 G51 (перехід по круговій траєкторії) 11-286
11.6.4.11 G52 (Плавний перехід траєкторії) 11-287
11.6.4.12 M Код 11-289
11.6.5 Інструкції щодо коду G 11-291
11.6.5.1 DMC_CartesianCoordinate 11-291
11.6.5.2 DMC_ReadMFunction ..... 11 -297
11.6.5.3 DMC_ResetMFunction 11-300
11.6.5.4 DMC_SetG0Para..... 11-303
11.6.5.5 DMC_SetG1Para..... 11-306
11.6.5.6 DMC_SetStartPosition 11-309

11.7 Axes Group Instructions..... 11-313
11.7.1 DMC_AddAxisToGroup..... 11-313

11.7.2 DMC_RemoveAxisFromGroup 11-315
11.7.3 DMC_UngroupAllAxes 11-317
11.7.4 DMC_GroupEnable 11-319
11.7.5 DMC_ GroupStop 11-322
11.7.6 DMC_GroupInterrupt 11-328
11.7.7 DMC_GroupContinue 11-334
11.7.8 DMC_MoveDirectAbsolute..... 11-336
11.7.9 DMC_MoveDirectRelative 11-341
11.7.10 DMC_MoveLinearAbsolute ..... 11 -346
11.7.11 DMC_MoveLinearRelative..... 11-360
11.7.12 DMC_MoveCircularAbsolute 11-373
11.7.13 DMC_MoveCircularRelative 11-384
11.7.14 DMC_GroupSetOverride..... .. 11-395
11.7.15 DMC_GroupReadActualPosition 11-400
11.8 Coordination Instructions 11-402
11.8.1 DMC_ControlAxisByPos 11-402
11.8.2 DMC_NC 11-407
Розділ 12 Усунення несправностей 12-1
12.1 Explanation of LED Indicators.....12-2
12.2 Table of Error IDs in Motion Instructions.....12-7
12.3 System Trouble Diagnosis through System Error Codes.....12-19
Додаток А Зв'язок Modbus A-1
A.1 Message Format in ASCII Mode..... A-2
A.2 Message Format in RTU Mode A-5
A.3 Modbus Function Codes Supported A-7
A.4 Modbus Exception Response Code Supported A-7
A.5 Introduction to Modbus Function Codes..... A-8
A.6 Table of Registers and Corresponding Modbus addresses..... A-15
Додаток Б Зв'язок Modbus TCP B-1
B.1 Modbus TCP Message Structure B-2
B.2 Modbus Function Codes Supported in Modbus TCP.....B-2
B.3 Exception Response Code in Modbus TCP.....B-3
B.4 Modbus Function Codes in Modbus TCP.....B-3
B.5 Registers in PLC and Corresponding Modbus Addresses B-12
Додаток С Протокол CANopen C-1

C.1 Node States C-4
C.2 Network Management (NMT) C-6
C.3 PDO (Process Data Object) C-6
C.4 SDO (Service Data Object) C-8
Додаток D Пояснення режимів наведення D-1
D.1 Explanation of Homing Modes D-2
Додаток E Список аксесуарів E-1
E.1 Accessories for EtherCAT Communication E-2
E.2 Accessories for CANopen Communication E-2
E.3 Accessories for DeviceNet Communication E-5

⚠ Caution

- У цьому посібнику міститься ознайомлення з функціями продукту, специфікаціями, установкою, основними операціями та налаштуваннями.
- Цей продукт є пристроєм ВІДКРИТОГО ТИПУ, тому його слід встановлювати в корпусі, вільному від пилу, вологи, ударів електричним струмом і вібрації. Корпус повинен запобігти управлінню пристроєм необслуговуючим персоналом (наприклад, для роботи з корпусом потрібні ключ або спеціальні інструменти) у разі виникнення небезпеки та пошкодження пристрою.
- Обов’язково уважно прочитайте інструкцію та дотримуйтесь інструкцій, що містяться в ній, щоб уникнути травмування персоналу та пошкодження виробів.

Розділ 1 Передмова

Зміст

1.1 Explanation of Symbols in This Manual..... 1-2

1.2 Revision History 1-2

Дякуємо за придбання контролера руху серії AS500E, створеного на основі керування рухом, і ми надаємо вам високоякісну систему керування рухом.

1

У цьому посібнику описано технічні характеристики продукту, функції, архітектуру системи, установку, підключення, принцип виконання, логічні інструкції та інструкції з керування рухом, усунення несправностей, протоколи зв'язку, режими наведення та іншу відповідну інформацію.

Перед використанням контролера руху серії AS500E переконайтеся, що ви добре знаєте конфігурацію системи керування рухом і роботу продукту.

1.1 Пояснення символів у цьому посібнику

- **Запобіжні заходи перед операцією**

Перед початком роботи уважно прочитайте відповідні інструкції з техніки безпеки, щоб запобігти травмуванню персоналу та пошкодженню виробів.

⚠ Небезпека	вказує на потенційну небезпеку. Тяжкі тілесні ушкодження або навіть **смерть** результат, якщо ви не дотримуєтесь інструкцій.
⚠ УВАГА	вказує на потенційну небезпеку. Недотримання інструкцій **може призвести до** незначних тілесних ушкоджень або навіть смерті .
⚠ Обережно	вказує на те, що слід приділяти велику увагу. Недотримання інструкцій може статися нещасним випадком.

1.2 Історія версій

Версія	Ревізія	Дата випуску
1 - й	Вийшла перша версія.	6 грудня 2019 р
2- й	1. Додано нові моделі AS532EST-B і AS564EST-B і відповідний опис. 2. Додано входи Write_FunCode інструкцій конфігурації в розділі 8.14 і відповідний опис. 3. Додано розділ 8.22 Інструкції для правого модуля розширення.	25 грудня 2020 р

2

Розділ 2 Огляд

Зміст

2.1 Product Description 2-2
2.2 Functions 2-2
2.3 Profile and Components 2-3

2.1 Опис продукту

Контролери руху серії AS500E — це багатоосьові контролери руху, розроблені та вироблені Delta автономно на основі польової шини EtherCAT. Швидкість передачі по шині становить 100 Мбіт/с, застосовуючи розподільний годинник для високошвидкісної, точної та високоефективної передачі даних і зручної синхронізації кількох осей.

Крім того, він також підтримує стандартні бібліотеки інструкцій керування рухом, визначені міжнародними організаціями. Це забезпечує користувачам велику зручність для швидкого вивчення та розробки проектів. Декількома осями можна керувати через порт EtherCAT. Підтримуються інструкції щодо руху по одній осі, включаючи інструкції щодо швидкості, положення, крутного моменту та самонаведення, а також інструкції по кількох осях, такі як електронний механізм, електронний кулачок, поворотне різання та код G.

У контролері руху вбудовано декілька комунікаційних портів. Таким чином, різні комунікаційні функції можуть бути реалізовані без додавання модулів. Контролер руху має правий розширювальний порт для розширення модулів серії AS з правого боку.

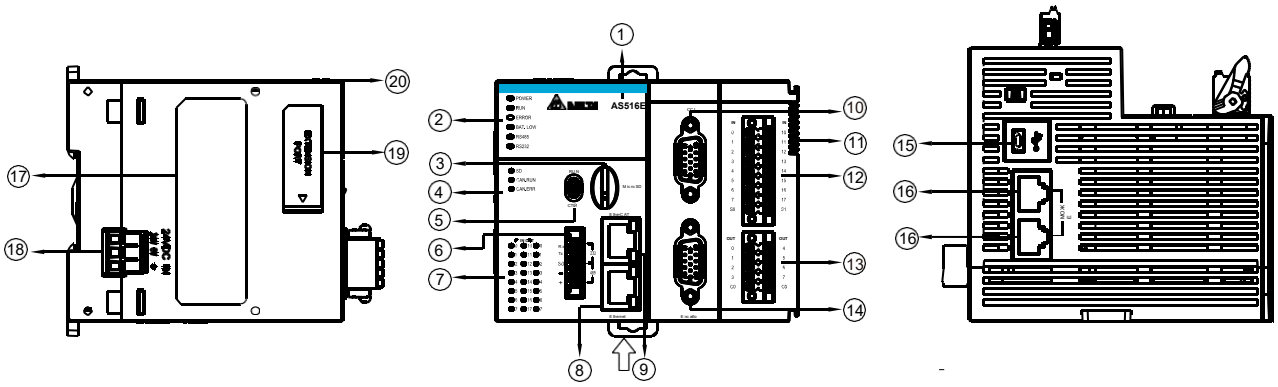
Контролер руху також має вбудований головний комунікаційний порт EtherCAT. Електропроводка проста і зручна. Завдяки високошвидкісній надійній шині EtherCAT і високошвидкісному процесору з плаваючою комою 1 ГГц контролер руху може широко застосовуватися в широкому діапазоні секторів автоматизованого керування, таких як пакування, друк, інкапсуляція, різання дроту, виробництво ліків тощо.

Контролери руху серії AS500E включають три моделі AS516E-B, AS523EST-B і AS564EST-B. Макс. кількість осей і багатоосьових інструкцій, які ці моделі можуть контролювати та підтримувати, відрізняються (додаткову інформацію див. у специфікаціях у розділі 3). Але вони мають однакові специфікації щодо портів зв'язку, лівих і правих розширень і програмної ємності.

2.2 Функції

- Швидкість передачі становить 100 Мбіт/с у мережі шини EtherCAT для високошвидкісної, точної та високоефективної передачі даних.
- Для різних моделей макс. кількість осей, якими вони можуть керувати, і багатоосьові інструкції, які вони можуть підтримувати, відрізняються. Щоб отримати докладнішу інформацію, зверніться до технічних характеристик у розділі 3.
- Віртуальна вісь і вісь кодера можуть бути побудовані всередині контролера (з номером осі в діапазоні від 1 до 32, який не може збігатися з номером реальної осі).
- Оснащений високошвидкісним процесором операцій з плаваючою комою 1 ГГц; підтримує 64-розрядне число з плаваючою крапкою (LREAL) і здатне відповідати різноманітним складним керуванням рухом.
- З двома вбудованими портами інкрементного кодера та одним портом абсолютного кодера SSI.
- Контролер має один порт EtherCAT, один порт Ethernet, один порт USB, один порт RS-232 і один порт RS-485.
- З одним вбудованим портом CAN, який служить головним або підлеглим CANopen.
- Підтримує потужну польову мережу (як провідний або підлеглий пристрій Ethernet і провідний або підлеглий пристрій CANopen) для побудови функціонально складної системи керування.
- З різними розширеннями вводу/виводу (цифрові модулі, аналогові модулі, модулі температури тощо).
- Загальна кількість цифрових модулів і спеціальних модулів, які підключаються до правої сторони контролера руху, становить до 32, серед яких кількість спеціальних модулів, таких як аналогові модулі, модулі температури тощо, становить до 16.
- Загальна кількість точок цифрового входу та виходу становить до 1024, включаючи цифрові точки, вбудовані в контролер.
- Використання простого у використанні інтерфейсу програмного забезпечення з функціями повної функції та зручного застосування.
- Надання стандартних шинних кабелів для легкого та зручного підключення проводки «підключи і працюй».

2.3 Профіль і компоненти



◯ 1	Назва моделі	◯ 11	Слот для модуля розширення
◯ 2	Показники стану	◯ 12	Вхідні клеми
◯ 3	Слот для карт SD	◯ 13	Вихідні клеми
◯ 4	Індикатори стану SD/CAN	◯ 14	Комунікаційний порт кодера
◯ 5	Перемикач RUN/STOP	◯ 15	Порт зв'язку USB
◯ 6	Порт RS-232/RS-485	◯ 16	Комунікаційний порт CAN
◯ 7	Індикатори введення/виведення	◯ 17	Табличка
◯ 8	Комунікаційний порт Ethernet	◯ 18	Вхідний порт живлення
◯ 9	Комунікаційний порт EtherCAT	◯ 19	Перемикач функцій
◯ 10	Комунікаційний порт SSI	◯ 20	Тримач батареї (батарея не входить до продукту, який ви купуєте)

Пам'ятка

3

Розділ 3 Технічні характеристики

Зміст

3.1 Function Specifications..... 3-2
3.1.1 Технічні характеристики 3-2
3.1.2 Пристрої та типи даних 3-4
3.1.2.1 Пристрої 3-4
3.1.2.2 Дійсні діапазони пристроїв 3-5
3.1.2.3 Блоковані пристрої 3-5
3.1.2.4 Підтримувані типи даних і дійсні діапазони 3-6
3.2 Electrical Specifications..... 3-6

3.1 Характеристики функцій

3.1.1 Технічні характеристики

Пункт				Специфікація		
				AS516E-B	AS532EST-B	AS564EST-B
Програмування	Ємність програми	Розмір		20M		
		Кількість	Кількість ПОУ визначення	1024		
	Ємність пам'яті для змінних	Зберігається	Розмір	128 тис		
		Не зберігається	Розмір	20M		
	код G	Одна єдина програма G-коду	Розмір	256 тис		
		Програми G коду	Кількість	64		
Контроль руху	Кількість керованих осей	Кількість реальних осей		16	32	64
		Кількість віртуальних осей + кодери		32	64	64
		Кількість реальних осей + віртуальних осей + кодери		32	64	64
		Діапазон номерів осей		1~32	1~64	1~64
	Макс. кількість осей, що підтримують багатоосьові інструкції (*1)			32	4 (*2)	8 (*2)
	Макс. кількість осей для лінійної інтерполяції			8	4	8
	Макс. кількість осей для кругової інтерполяції			3	3	3
	Кількість кулачків	Розмір	Кількість	64		
	Ключові моменти камери	Ключові моменти однієї камери	Кількість	2048		
Вбудовані порти		EtherCAT	Кількість	1, зі швидкістю передачі даних 100 Мбіт/с; Для отримання додаткової інформації зверніться до порту зв'язку EtherCAT у розділі 6 Електропроводка, зв'язок і конструкція мережі.		
		Ethernet	Кількість	1, використовується як головний або підлеглий Щоб отримати докладніші відомості, зверніться до Комунікаційного порту Ethernet у розділі 6 Електропроводка, зв'язок і конструкція мережі.		
		USB	Кількість	1 стандартний інтерфейс USB2.0, який використовується як підлеглий або для завантаження та завантаження програм.		
		CAN	Кількість	1, використовується як головний або підлеглий CANopen; Як головний, він може підключити до 32 підлеглих.		
		RS-232	Кількість	1, використовується як головний або ведений		
		RS-485	Кількість	1, використовується як головний або ведений Як головний, він може підключити до 24 підлеглих.		

Пункт			Специфікація		
			AS516E-B	AS532EST-B	AS564EST-B
	Інкрементний кодер	Кількість	2, можуть бути побудовані як осі кодера. Сигнал Z може викликати програму переривання.		
	Абсолютний кодер SSI	Кількість	1, може бути побудований як вісь кодера		
	Вхідні точки	Кількість	16 точок, підтримує тригер зовнішнього переривання		
	Вихідні точки	Кількість	8 балів		
	Правий розширювальний порт	Кількість	1, підключає спеціальний модуль серії AS		
Розширення вправо	Спеціальні модулі	Кількість	До 16 спеціальних модулів серії AS		
	Цифрові точки	Кількість балів	Всього 1024 точки входу плюс точка виведення, включаючи локальні цифрові точки вводу-виводу		
	Спеціальні модулі та цифрові модулі	Кількість	Макс. 32 модулі серії AS		
RTC (годинник реального часу)	Акумулятор	Кількість балів	1, користувачі повинні мати одну батарею CR1620 3 В, яку можна придбати на ринку всередині контролера руху. RTC не може працювати, якщо батарея не подає живлення нормально,		

Примітка:

(*1): Багатоосьові інструкції, згадані тут, включають інструкції щодо зчеплення, інструкції із застосування, Інструкції щодо координації, інструкції G Code та інструкції Axis Group. Для наборів інструкцій зверніться до таблиці інструкцій керування рухом у розділі 11.1.

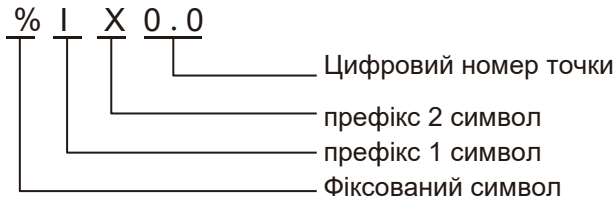
(*2): кількість осей, що підтримують багатоосьові інструкції, перераховується з 0 після того, як контролер увімкнено. Коли кількість осей, які виконали багатоосьові інструкції, перевищить дозволену максимальну кількість, інші осі не зможуть виконувати багатоосьові інструкції. Коли виконуються інструкції з'єднання та застосування, головна вісь не буде зарахована до кількості осей, які виконують багатоосьові інструкції. Коли виконуються інші багатоосьові інструкції, крім інструкцій з'єднання та застосування, усі використані осі зараховуються до кількості осей, які виконують багатоосьові інструкції.

Для прикладу AS532EST-B кількість осей, що підтримують багатоосьові інструкції, становить 4. Вісь 1 (головна) і вісь 2 (підпорядкована) виконали інструкцію MC_GearIn, вісь 1 (головна) і вісь 3 (підлегла) виконали інструкцію MC_GearIn і Вісь 1 (головна) і вісь 4 (підпорядкована) виконали інструкцію MC_GearIn. Тоді інші осі не зможуть виконати жодну багатоосьову інструкцію після того, як вісь 1 (головна) і вісь 5 (підпорядкована) виконали інструкцію MC_GearIn. Навіть якщо осі 2 ~ 5 перервано одноосьовою інструкцією, інші осі все одно не зможуть виконати будь-яку багатоосьову інструкцію. Наприклад, інші осі не можуть виконувати жодну багатоосьову інструкцію, а також після того, як осі 1 ~ 4 виконують інструкцію DMC_MoveLinearRelative.

3.1.2 Пристрої та типи даних

3.1.2.1 Пристрої

- **Пояснення назви пристрою**



- **Відповідні пристрої контролера руху, які використовуються в програмному забезпеченні**

№	Пункт	Зміст				
1	Префікс 1 символ	I	Q	M		
2	Префікс 1 імені	Пристрій введення	Вивідний пристрій	Проміжний пристрій		
3	Префікс 2 символ	X	B	W	D	L
4	Тип даних префікса 2	BIT	BYTE	WORD	DWORD	QWORD
5	Приклад пристрою	%IX0,0	%IB0	%IW0	%ID0	%IL0
6		%QX0,0	%QB0	%QW0	%QD0	%QL0
7		%MX0,0	%MB0	%MW0	%MD0	%ML0

- **Відповідні зв'язки пристроїв**

%ML0 включає %MB0~%MB7, %MD0 включає %MB0~%MB3 і %MW0 включає %MB0~%MB1, як показано в наступній таблиці.

Ім'І пристрою	Відповідні відносини																							
	1- е WORD						2 -е WORD						3 -тє WORD						4- е WORD					
	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15
%MX	%MX0,0~0,7			%MX1,0~1,7			%MX2.0~2.7			%MX3.0~3.7			%MX4.0~4.7			%MX5.0~5.7			%MX6.0~6.7			%MX7,0~7,7		
%MB	%MB0			%MB1			%MB2			%MB3			%MB4			%MB5			%MB6			%MB7		
%MW	%MW0						%MW1						%MW2						%MW3					
%MD	%MD0												%MD1											
%ML	%ML0																							

%ML1 включає %MB8~%MB15, %MD2 включає %MB8~%MB11, %MW4 включає %MB8~%MB9 і %MB8 включає %MX8.0~8.7, як показано в наступній таблиці.

Ім'І пристрою	Відповідні відносини																							
	5- е WORD						6- е WORD						7- е WORD						8- е WORD					
	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15	Біт 0	…	Біт 7	Біт 8	…	Біт 15
%MX	%MX8,0~8,7			%MX9,0~9,7			%MX10,0~10.7			%MX11.0~11.7			%MX12.0~12.7			%MX13.0~13.7			%MX14.0~14.7			%MX15,0~15.7		
%MB	%MB8			%MB9			%MB10			%MB11			%MB12			%MB13			%MB14			%MB15		
%MW	%MW4						%MW5						%MW6						%MW7					
%MD	%MD2												%MD3											
%ML	%ML1																							

3.1.2.2 Дійсні діапазони пристроїв

- **Таблиця допустимих діапазонів пристроїв у контролері руху**

Ім'І пристрою	експресія	Діапазон
%IX	%IX0.0~%IX0.7	%IX0.0~%IX127.7
%QX	%QX0.0~%QX0.7	%QX0.0~%QX127.7
%MX	%MX0,0	%MX0.0~%MX131071.7
%IB	%IB0	%IB0~%IB127
%QB	%QB0	%QB0~%QB127
% MB	%MB0	%MB0~%MB131071
%IW	%IW0	%IW0~%IW63
%QW	%QW0	%QW0~%QW63
%MW	%MW0	%MW0~%MW65535
%ID	%ID0	%ID0~%ID31
%QD	%QD0	%QD0~%QD31
%MD	%MD0	%MD0~%MD32767
%IL	%IL0	%IL0~%IL15
%QL	%QL0	%QL0~%QL15
%ML	%ML0	%ML0~%ML16383

- **Таблиця адрес пристроїв Modbus**

Доступ до адрес Modbus, які знаходяться в діапазоні 16#0000~16#FFFF, можна отримати через стандартний код функції MODBUS 01, 02, 03, 05, 06, 16#0F і 16#10. Якщо код функції MODBUS використовується для доступу до бітових пристроїв, будь ласка, отримайте доступ до бітових пристроїв QX та IX. Якщо потрібно отримати доступ до пристроїв з бітами MX, здійснюється доступ до пристрою MW, а потім значення в пристроях з бітами MX можна отримати через пристрій MW.

Дивіться Розділ 3.1.2.1, щоб дізнатися більше про відповідні відносини між пристроями MW і MX.

Область пристрою	Тип пристрою	Діапазон	Адреса Modbus	Тип адреси Modbus
I (Вхід)	біт	%IX0.0~%IX0.7	16#6000~16#6007	Стандартна адреса Modbus
		%IX1.0~%IX1.7	16#6008~16#600F	
		……	……	
		%IX127.0~%IX127.7	16#63F8~16#63FF	
	Слово	%IW0~%IW63	16#8000~16#803F	
Q (Вихід)	біт	%QX0.0~%QX0.7	16#A000~16#A007	
		%QX1.0~%QX1.7	16#A008~16#A00F	
		……	……	
		%QX127.0~%QX127.7	16#A3F8~16#A3FF	
	Слово	%QW0~%QW63	16#A000~16#A03F	
M (Регістр)	Слово	%MW0~%MW32767	16#0000~16#7FFF	

3.1.2.3 Блоковані пристрої

Пристрої %MW0~%MW999 є пристроями з фіксацією, у яких дані зберігаються після вимкнення живлення. Крім того, змінні, визначені в програмному забезпеченні, можуть вибрати Retain як свою властивість. Ємність заблокованих пристроїв становить 128 Кбайт.

3.1.2.4 Підтримуються типи даних і допустимі діапазони

Типи даних і допустимі діапазони змінних у програмному забезпеченні, яке використовує контролер руху, наведено в наступній таблиці.

№	Тип даних	Допустимий діапазон	Початкове значення
1	BOOL	TRUE or FALSE	FALSE
2	BYTE	16#00 ~ FF	16#00
3	WORD	16#0000 ~ FFFF	16#0000
4	DWORD	16#00000000 ~ FFFFFFFF	16#00000000
5	LWORD	16#0000000000000000 ~ FFFFFFFFFFFFFFFFFFFF	16#0000000000000000
6	USINT	0 ~ +255	0
7	UINT	0 ~ +65535	0
8	UDINT	0 ~ +4294967295	0
9	ULINT	0 ~ +18446744073709551615	0
10	SINT	−128 ~ +127	0
11	INT	−32768 ~ +32767	0
12	DINT	−2147483648 ~ +2147483647	0
13	LINT	−9223372036854775808 ~ +9223372036854775807	0
14	REAL	−3,402823e+38 ~ −1,175495e-38, 0, +1,175495e-38 ~ +3,402823e+38	0,0
15	LREAL	−1,79769313486231e+308 ~ −2,22507385850721e- 308, 0, +2,22507385850721e−308 ~ +1.79769313486231e+308,	0,0
16	TIME	T#XXXXXXdXXhXXmXXsXXXms, одиниця: ns. Діапазон: T#0ms~213503d23h34m33s709,551ms	T#0ms
17	DATE	D#YMD. Діапазон: D#1970-01-01~D#2106-02-07. одиниця:с.	D#1970-01-01
18	TOD	TOD#H:M:S:MS, діапазон:TOD#00:00:00~23:59:59.999. Одиниця: мс. Якщо вказано 0, відображається TOD#00:00:00. Якщо вказано 1, відображається TOD#00:00:00.001. Якщо вказано 86399999, відображається TOD#23:59:59.999. Якщо вказано 86400000, відображається TOD#00:00:00. Якщо вказано 4294967295, відображається TOD#17:2:47.295.	TOD#00:00:00
19	DT	DT#Y-M-D-H-M-S. Діапазон: DT#1970-01-01-0:0:0~2106-02-07-6:28:15. Одиниця: с.	DT#1970-01-01-0:0:0
20	STRING	0~32000 символів	''

3.2 Електричні характеристики

- Електричні характеристики

Пункт	Зміст
Напруга живлення	24 В постійного струму
Ємність запобіжника	3 А/30 VDC, поліперемикач
Напруга ізоляції	500 В змінного струму（ вторинний PE ）
Потужність споживання	Макс. 8 Вт

Пункт	Зміст			
Потужність споживання на правому порту розширення	Макс. вихідна потужність 36 Вт на правому порту розширення			
Робоча температура	-20~55°C			
Температура зберігання	-40~80°C			
Вимоги до акумулятора	Дозволена температура навколишнього середовища батареї має бути не менше 100 ℃ , а зворотний струм має бути не менше 4 мА.			
Робоча вологість	5~95% Відсутність конденсату			
Вологість при зберіганні	5~95% Відсутність конденсату			
Робоче середовище	Відсутність корозійного газу			
Ступінь забруднення	2			
Стійкість до вібрації	Випробувано з: 5 Гц ≦ f ≦ 8,4 Гц , постійна амплітуда 3,5 мм ; 8,4 Гц ≦ f ≦ 150 Гц , постійне прискорення 1g Тривалість коливань : 10 циклів розгортки на вісь у кожному напрямку 3 взаємно перпендикулярних осей Міжнародний стандарт: IEC 61131-2 & IEC 60068-2-6 （ TECT Fc ）			
Ударостійкість	Перевірено за допомогою : напівсинусоїда : Сила удару 15 g пікове значення , тривалість 11 мс ; Напрямок удару : удари в кожному напрямку на вісь , на 3 взаємно перпендикулярних осях (всього 18 ударів) Міжнародний стандарт: IEC 61131-2 & IEC 60068-2-27 （ TECT Ea ）			
EMI	Порт	Діапазон частот	Рівень (нормативний)	Еталонний стандарт
	Порт корпусу (випромінюється) (вимірюється на відстані 10 метрів)	30-230 МГц	40 дБ (мкВ/м) квазіпіковий	IEC 61000-6-4
		230-1000 МГц	47 дБ (мкВ/м) квазіпіковий	
	Порт живлення змінного струму (проведений)	0,15-0,5 МГц	79 дБ (мкВ) квазіпіковий	IEC 61000-6-4
			66 дБ (мкВ) в середньому	
		0,5-30 МГц	73 дБ (мкВ) квазіпіковий	
			60 дБ (мкВ) в середньому	
EMS	Екологічний феномен	Еталонний стандарт	Тест	Тестовий рівень
	Електростатичний	IEC 61000-4-2	контакт	±4кВ

Пункт	Зміст				
	розрядка		повітря		±8 кВ
	Радіочастотне електромагнітне поле з амплітудною модуляцією	IEC 61000-4-3	80% ранку , 1 кГц синусоїдальний	2,0-2,7 ГГц	1 V/m
				1,4-2,0 ГГц	3 V/m
				80-1000 МГц	10 V/m
	Магнітне поле потужної частоти	IEC 61000-4-8	60 Гц		30 A/m
			50 Гц		30 A/m
вага	405г				

- Електричні характеристики для вхідних точок

Пункт	Зміст
Кількість вхідних каналів	16 каналів
Тип каналу	Високошвидкісний цифровий вхід для 16 каналів
Вхідні клеми	Термінал I0~I7 , I10~I17
Загальний термінал для точок введення	Термінал S0/S1
Тип введення	Режим прийому або джерела
Затримка введення	2,5 мкс （ OFF -> ON ）, 5 мкс （ ON -> OFF ）
Вхідний струм	24 В постійного струму, 5 мА
Макс. довжина кабелю	Екранований кабель: 500 м ;
	Неекранований кабель: 300м

- Електричні характеристики для вихідних точок

Пункт	Зміст
Кількість вихідних каналів	8 транзисторів для виходу (N-MOS)
Тип каналу	Високошвидкісний цифровий вихід на 8 каналів
Вихідні клеми	Термінал Q0~Q7
Загальний термінал для вихідних точок	Клема C0/C0 (для підключення до катода живлення)
Напруга живлення для вихідних точок	24 В постійного струму
Затримка виведення	2 мкс （ OFF -> ON ）, 3 мкс （ ON -> OFF ）
Макс. частота перемикання	1 кГц
Макс. завантаження	Опір: 0,2 A / 1 точка
	Індуктивність: 13 Вт （ 24 В постійного струму ）
	Лампа: 2,5 Вт （ 24 В постійного струму ）
Макс. довжина кабелю	Екранований кабель: 500м
	Неекранований кабель: 300м

Пункт	Зміст
	Індуктивність: 13 Вт （ 24 В постійного струму ）
	Лампа: 2,5 Вт （ 24 В постійного струму ）
Макс. довжина кабелю	Екранований кабель: 500м
	Неекранований кабель: 300м

Пам'ятка

4

Розділ 4 Архітектура системи

Зміст

4.1 System Constitution 4-2

4.2 Introduction to EtherCAT Fieldbus..... 4-2

4.2.1 Особливості EtherCAT Fieldbus 4-2

4.2.2 Зв’язок EtherCAT між контролером і підлеглими.....4-3

4.2.3 Ініціалізація мережі EtherCAT 4-4

4.2.4 Як усунути невідповідність кількості між EtherCAT Конфігурація та фактично підключені пристрої 4-4

4.2.5 Підлеглий EtherCAT в автономному режимі та механізм відновлення ..4-4

4.2.6 Сервоприводи, які підключаються до порту EtherCAT 4-5

4.3 Power Supply 4-7

4.4 Right-side Extension 4-7

4.4.1 Правосторонні модулі розширення, що підключаються 4-7

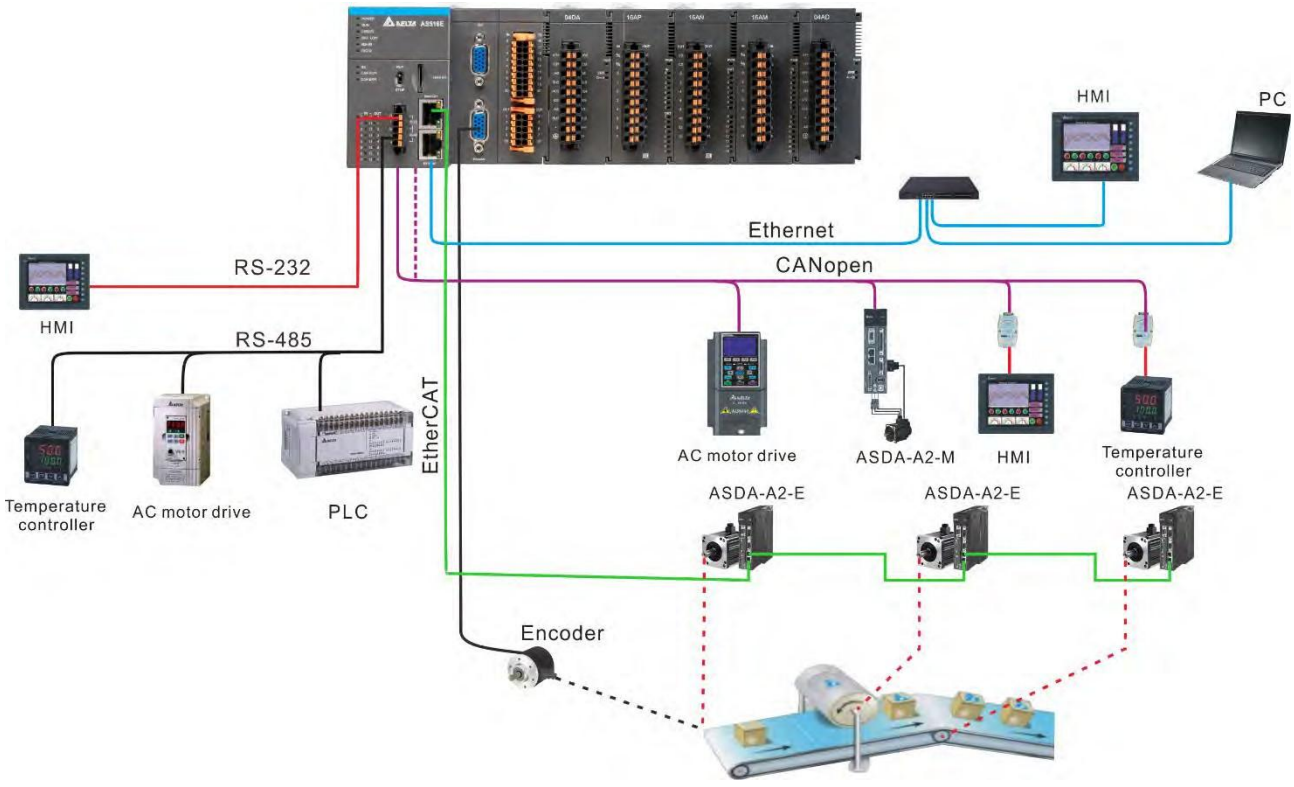
4.4.2 Розподіл адрес правого модуля розширення 4-9

4.5 SD Memory Card..... 4-9

4.5.1 Модель і технічні характеристики 4-9

4.1 Конституція системи

За допомогою контролера руху можна побудувати багатошарову промислову мережу. За допомогою контролера руху мережа може складатися з Ethernet верхнього рівня, CANopen середнього рівня, а також шини RS-485 нижнього рівня, що підтримує Modbus, як описано нижче.



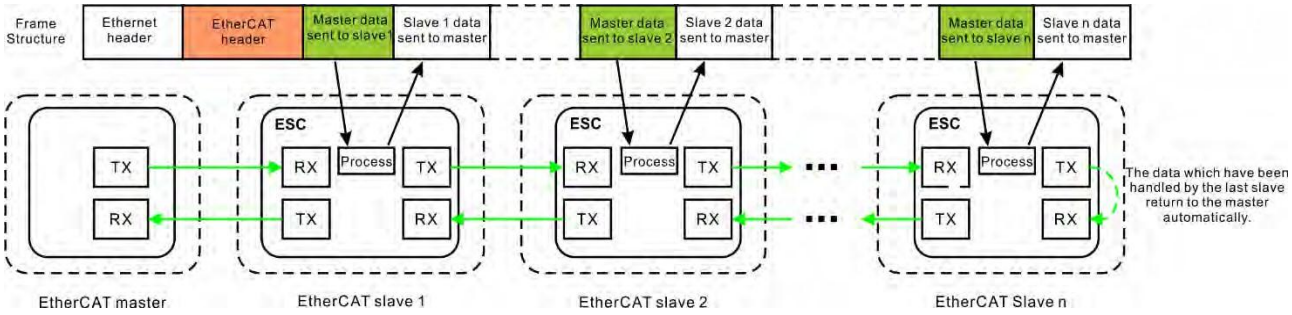
На малюнку вище показано периферійні пристрої, підключені до різних портів контролера руху в усій системі. Зверніться до розділу 6, щоб дізнатися більше про функції комунікаційних портів.

4.2 Введення в EtherCAT Fieldbus

4.2.1 Особливості EtherCAT Fieldbus

Шина EtherCAT — це польова шина на основі Ethernet. Швидкість зв'язку в мережі EtherCAT становить 100 Мбіт/с, а відстань між двома сусідніми вузлами не перевищує 50 метрів. Очевидно, на відміну від загальної мережі Ethernet, одна мережа EtherCAT має лише один головний EtherCAT, а підлеглі EtherCAT містять мікросхеми ESC (підлеглий контролер EtherCAT), які спеціально використовуються для обробки даних зв'язку EtherCAT і вставки даних, які підлеглі повинні передати головному, у кадр EtherCAT. Останній підлеглий пристрій EtherCAT у мережі поверне дані, які були оброблені головному. Ілюстрація передачі даних показана нижче.

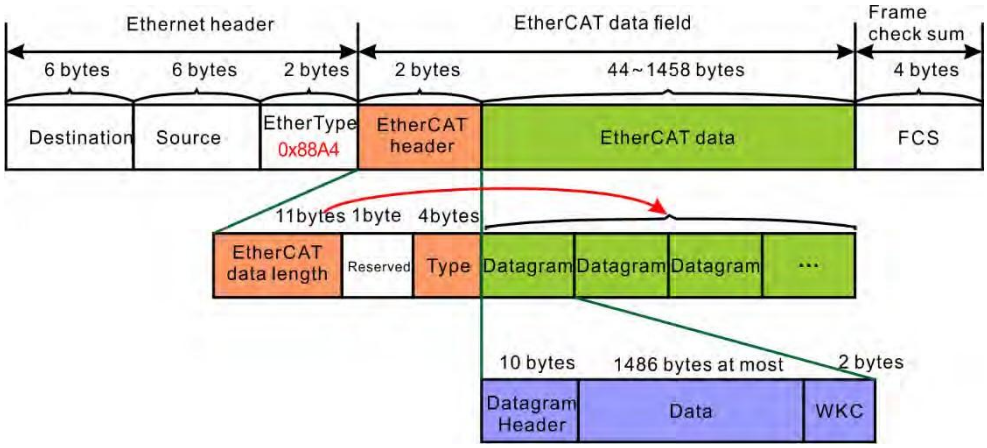
Під час обміну даними між головним і підлеглими підлеглими обробляються дані шини EtherCAT через чіпи ESC. Завдяки цьому ефективність обробки даних шини була значно покращена. Завдяки чіпам ESC у підлеглих пристроях головний пристрій може здійснювати зв'язок із усіма підлеглими у кадрі даних EtherCAT, що підвищує ефективність зв'язку.



4.2.2 Зв'язок EtherCAT між контролером і підлеглими

Оскільки шина EtherCAT є польовою шиною на основі EtherNet, кадр даних EtherCAT все ще приймає структуру кадру даних UDP/IP Ethernet.

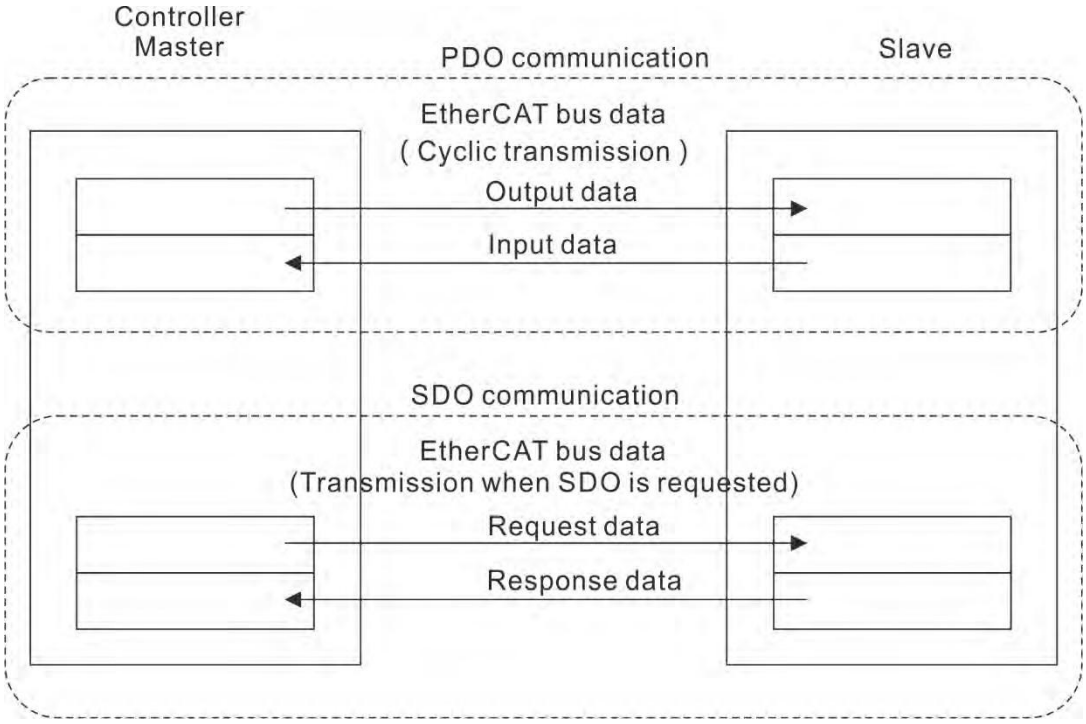
Структура кадру даних EtherCAT показана нижче. Поле даних EtherCAT містить 2 байти заголовка даних EtherCAT і 44~1498 байт даних EtherCAT. Поле даних EtherCAT складається з однієї або кількох дейтаграм EtherCAT. Дані EtherCAT можуть бути визначені та проаналізовані в протоколі, якщо головний і підлеглі пристрої відповідають протоколу. Наразі найчастіше використовуються два протоколи: COE (CANopen через EtherCAT) і SOE (Sercos через EtherCAT). Контролер руху серії AS500E використовує протокол COE.



Порт EtherCAT контролера обмінюється даними з підлеглими EtherCAT згідно з протоколом COE （ CANopen через EtherCAT ） . Існує два види методів передачі для контролера та підлеглих пристроїв. Один є циклічний обмін даними на основі заданого часового циклу, який називається PDO (Process Data Object). Інший — це обмін даними запит-відповідь, який називається SDO (Service Data Object).

Передача даних PDO використовується для швидкого циклічного обміну даними. У той час як головний і підлеглі обмінюються даними через PDO, іншій стороні не потрібно давати жодної відповіді після того, як одна сторона надсилає дані. Якщо контролер керує підлеглим EtherCAT за допомогою інструкцій руху, обмін даними між контролером і підлеглими здійснюється через PDO.

Дані SDO надсилаються лише тоді, коли ведучому потрібно прочитати або записати дані в ведений. Метод передачі SDO може бути використаний лише для того, щоб головний пристрій зчитував або записував дані на підлеглому пристрої, а підлеглий пристрій повинен відповідати головному. Читання або запис даних через SDO можна досягти за допомогою інструкцій DMC_ReadParameter_Motion і DMC_WriteParameter_Motion.



4.2.3 Ініціалізація мережі EtherCAT

Підлеглий пристрій EtherCAT, яким керує головний EtherCAT, має бути налаштований на головний у програмному забезпеченні. Як головний EtherCAT, контролер ініціалізує підлеглий пристрій, налаштований у програмному забезпеченні, після УВІМКНЕННЯ живлення. Ініціалізація включає наступні процедури.

1. Команда ініціалізації мережі надсилається через трансляцію, і починається ініціалізація всіх підлеглих пристроїв. Усі підлеглі переходять у передопераційний стан.
2. Підлеглі дані для циклічного обміну налаштовуються на основі конфігураційної інформації в програмному забезпеченні.
3. Підлеглий пристрій, який успішно налаштований, переходить у робочий стан і встановлює з'єднання з головним.
4. Після того, як налаштований підлеглий пристрій переходить у робочий стан, головний і підлеглий пристрій починають виконувати циклічний обмін даними.

Наведений вище процес ініціалізації завершується контролером без дії користувача.

4.2.4 Як усунути невідповідність кількості між конфігурацією EtherCAT і фактично підключеними пристроями

Після того, як головний ініціалізує підлеглі пристрої, усі підлеглі пристрої, які успішно налаштовані, можуть контролюватися головним, незалежно від того, чи є кількість підлеглих пристроїв, налаштованих у програмному забезпеченні, такою ж, як кількість пристроїв, фактично підключених до порту EtherCAT, чи ні. Наприклад, якщо в програмному забезпеченні налаштовано два підлеглих пристрої EtherCAT, але фактично підключено лише одне з них, фактично підключеним можна керувати за допомогою інструкцій руху. Підлеглий пристрій, який було налаштовано в програмному забезпеченні, підключається до мережі після того, як провідний встановив з'єднання з підлеглим, який був підключений до мережі. У цьому випадку ведучий не буде з'єднуватися з веденим, який підключається до мережі пізніше.

4.2.5 EtherCAT Slave Offline і механізм відновлення

Ведучий не встановлюватиме з'єднання з офлайн-підпорядкованими системами знову, якщо деякі з підлеглих пристроїв EtherCAT перебувають у режимі офлайн через видалення кабелю зв'язку після того, як підлеглі EtherCAT і головний пристрій встановлять з'єднання. Офлайн-підпорядковані пристрої не можна контролювати за допомогою інструкцій керування рухом, а решта підлеглих пристроїв, які не працюють у режимі офлайн, не вплинуть.

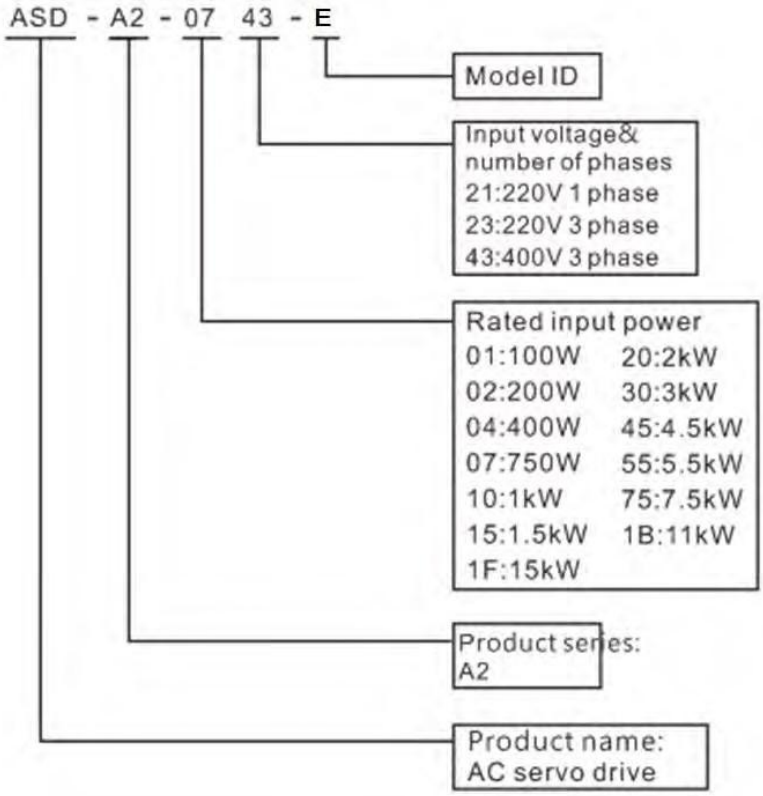
Якщо всі підлеглі пристрої, налаштовані на головному пристрої, перебувають у режимі офлайн, головний і всі підлеглі пристрої встановлюють з’єднання знову. Після того, як підключення буде встановлено знову, MC_Reset потрібно виконати на офлайн-підпорядкованих пристроях, і тоді підлеглими можна буде керувати.

Якщо офлайн-підпорядковані пристрої повинні знову встановити з’єднання з головним, кабель EtherCAT між контролером і першим сервоприводом слід повторно під’єднати після видалення або повторного увімкнення контролера. Якщо нормально працюючі підпорядковані пристрої постраждали через операції, згадані вище, нормально працюючі підлеглі та головний встановлять з’єднання знову. Якщо якась вісь працює, її рух буде негайно припинено в цій ситуації.

4.2.6 Сервоприводи підключаються до порту EtherCAT

Існує багато моделей для сервоприводів серій ASDA-A2, ASDA-B3 і ASDA-A3. Моделі ASDA-A2-XXXX-E, ASDA-A2-XXXX-EN, ASDA-B3-XXXX-E та ASDA-A3-XXXX-E підтримують зв’язок EtherCAT. Тільки ці моделі сервоприводів можна використовувати для побудови мережі керування рухом EtherCAT через підключення до порту EtherCAT серії AS500E.

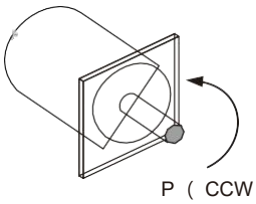
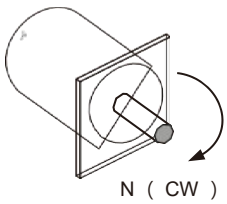
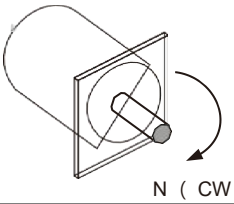
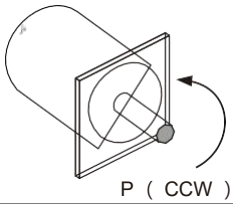
- Ілюстрація моделі сервоприводу



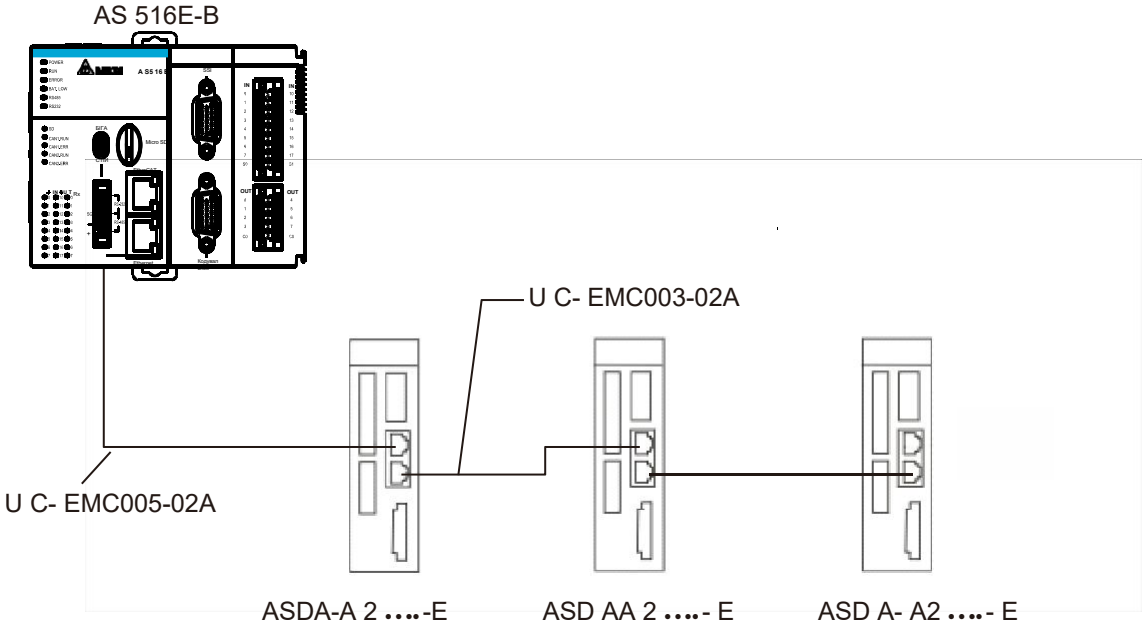
- Відповідні налаштування параметрів сервоприводу показані в наступній таблиці, коли контролер руху та сервопривод підключено.

Параметр	Пояснення	Значення налаштування	Пояснення
П1-01	Налаштуйте режим керування сервоприводом	X0C* [1]	Встановити як режим EtherCAT

* [1]: вихідні напрямки крутного моменту показано нижче, коли значення X дорівнює 0 і 1 відповідно.

	0	1
Позитивний напрямок		
Негативний напрямок		

- Схема підключення контролера руху та сервоприводів серії ASDA-A2-XXXX-E



Примітка:

1. Будь ласка, зверніться до посібника з експлуатації сервоприводу для підключення сервоприводів серії ASDA-A2-XXXX-E, серводвигунів і кодерів.
2. Виберіть кабель зв'язку UC-EMC003-02A, UC-EMC005-02A або UC-EMC010-02A відповідно до польової ситуації.
3. Зверніться до розділу E.1 для пояснення моделей кабелю зв'язку EtherCAT.

4.3 Джерело живлення

Як джерело живлення для контролера руху рекомендовані силові модулі Delta. Інформація про модулі живлення Delta наведена в наступній таблиці.

№	Назва модуля	Вхідна напруга	Вихідна напруга	потужність	Вихідний струм	Міжнародний стандарт
1	AS-PS02	100~240 В змінного струму 50/60 Гц	24 В постійного струму (Для внутрішнього використання в ПЛК)	48 Вт	2A	
2	AS-PS02A		24 В постійного струму (Для внутрішнього використання в ПЛК)	36 Вт	1,5 A	
			24 В постійного струму (Для зовнішнього застосування)	12 Вт	0,5 A	

4.4 Правостороннє розширення

4.4.1 Підключаються праві модулі розширення

Модулі розширення серії AS, включаючи цифрові модулі, аналогові модулі та модулі температури, можна підключити до правої сторони контролера руху. Загальна кількість цифрових модулів і спеціальних модулів, які підключаються до правої сторони контролера, може становити до 32, серед яких кількість спеціальних модулів, таких як аналогові модулі, модулі температури тощо, становить до 16, а загальна кількість їх вхідних і вихідних точок до 992.

Дивіться підключені праві модулі розширення в наступній таблиці.

№	Назва модуля	Вхідна довжина	Вихідна довжина	Тип розширення
1	AS08AM10N-A	8 біт	-	Розширення точки введення
2	AS16AM10N-A	16 біт		
3	AS32AM10N-A	32 біти		
4	AS64AM10N-A	64 біти	-	
5	AS08AN01T-A	-	8 біт	Розширення вихідної точки
6	AS08AN01P-A		8 біт	
7	AS08AN01R-A	-	8 біт	
8	AS16AN01T-A		16 біт	
9	AS16AN01P-A		16 біт	
10	AS16AN01R-A		16 біт	
11	AS32AN02T-A		32 біти	
12	AS64AN02T-A	-	64 біти	
13	AS16AP11T-A	8 біт	8 біт	Вхідне розширення та вихідне розширення
14	AS16AP11P-A	8 біт	8 біт	
15	AS16AP11R-A	8 біт	8 біт	
16	AS04AD-A	10 слів	-	4-канальний модуль аналогового введення Апаратна роздільна здатність: 16 біт Номінальна вхідна напруга: 0–10 В, 0/1–5 В, від -5 до +5 В, -10 ~ +10 В, 0/4–20 мА, -20 ~ +20 мА Час перетворення: 2 мс/канал
17	AS08AD-B	18 слів	-	8-канальний модуль аналогового введення Апаратна роздільна здатність: 16 біт Номінальна вхідна напруга: від 0 до +10 В, 0/1–5 В, від -5 В до +5 В, -10 В ~

№	Назва модуля	Вхідна довжина	Вихідна довжина	Тип розширення
				+10В Час перетворення: 2 мс/канал
18	AS08AD-C	18 слів	-	8-канальний модуль аналогового введення Апаратна роздільна здатність: 16 біт Номінальний вхід: 0/4–20 мА, -20 мА ~ +20 мА Час перетворення: 2 мс/канал
19	AS04DA-A	2 слова	8 слів	4-канальний модуль аналогового виведення Апаратна роздільна здатність: 12 біт Номінальний вихід: -10 ~ +10 В, 0 ~ 20 мА, 4 ~ 20 мА Час перетворення: 2 мс/канал
20	AS06XA-A	10 слів	4 слова	4-канальний аналоговий вхід Апаратна роздільна здатність: 16 біт 0 ~10 В, 0/1 ~ 5 В, -5 ~ +5 В, -10 ~ +10 В, 0/4 ~ 20 мА, -20 ~ +20 мА Час перетворення: 2 мс/канал 2-канальний аналоговий вихід Апаратна роздільна здатність: 12 біт -10 ~ +10 В, 0 ~ 20 мА, 4 ~ 20 мА Час перетворення: 2 мс/канал
21	AS04RTD-A	10 слів	-	4-канальний, 2-/3-провідний RTD Тип датчика: Pt100 / Ni100 / Pt1000 / Ni1000 / JPt100 / LG-Ni1000 / Cu50 / Cu100 / 0-300 Ом / 0-3000 Ом вхідний опір Роздільна здатність: 0,1°C/0,1°F (16 біт) Час перетворення: 200 мс/канал
22	AS06RTD-A	14 слів	-	6-канальний, 2-/3-провідний RTD Тип датчика: Pt100 / Ni100 / Pt1000 / Ni1000 / JPt100 / LG-Ni1000 / Cu50 / Cu100 / 0-300 Ом / 0-3000 Ом вхідний опір, Роздільна здатність: 0,1°C/0,1°F (16 біт) Час перетворення: 200 мс/канал
23	AS04TC-A	10 слів	-	4-канальна термопара Тип датчика: J, K, R, S, T, E, N, B і -100 ~ +100 мВ Роздільна здатність: 0,1°C/0,1°F (24 біти) Час перетворення: 200 мс/канал
24	AS08TC-A	18 слів	-	8-канальна термопара Тип датчика: J, K, R, S, T, E, N, B і -100 ~ +100 мВ Роздільна здатність: 0,1°C/0,1°F (24 біти) Час перетворення: 200 мс/канал
25	AS02LC-A	7 слів	1 слово	2-канальний, 4-провідний/6-провідний датчик навантаження Власні значення датчика навантаження: 1, 2, 4, 6, 20, 40, 80 мВ/В Найвища точність 1/10000 при 50 мс часу перетворення Роздільна здатність АЦП: 4 біти Час перетворення: 2,5 ~ 400 мс (дев'ять варіантів на вибір)
26	AS02PU-A	15 слів	-	2-осьове керування позиціонуванням 5-24 В постійного струму, 1 (фаза A/B/Z) диференціальний вхід, апаратна максимальна пропускна здатність для входу: 200 кГц

№	Назва модуля	Вхідна довжина	Вихідна довжина	Тип розширення
				24 В постійного струму, 5 мА, 5 зовнішніх входів, апаратна максимальна смуга пропускання для входу: 1 кГц; 5 В постійного струму, 2-осьові (4 точки) високошвидкісні диференціальні виходи, максимальна смуга пропускання для виходу: 200 кГц;
27	AS04PU-A	20 слів	-	4-осьове керування позиціонуванням 24 В постійного струму, 5 мА, 6 входів, апаратна максимальна смуга пропускання для входу: 1 кГц; 5-30 В постійного струму, 0,1 А, 4-осьовий (8 точок) вихід NPN, максимальна смуга пропускання для виходу: 100 кГц

4.4.2 Розподіл адрес правого модуля розширення

- **Номери цифрових точок правих цифрових модулів розширення**
  1. Праві цифрові модулі можна налаштувати в програмному забезпеченні, і програмне забезпечення автоматично призначить пристрої введення та виведення для цифрових модулів.
  2. Номер цифрової точки цифрових модулів розширення, підключених праворуч від контролера руху, починається з 2.0. Наприклад, вхідна точка для першого цифрового модуля починається з %IX2.0, а вихідна — з %QX2.0. Це зараховується як 16 очок, якщо число менше 16.
  3. Цифрові вхідні та вихідні точки пронумеровані в десятковій системі, як показано нижче.
     Номер точки введення: %IX2.0 ~%IX2.7,....., %IX8.0 ~%IX8.7,....., %IX127.0 ~ %IX127.7
     Номер вихідної точки: %QX2.0 ~ %QX2.7,....., %QX8.0 ~ %QX8.7,....., %QX127.0 ~ %QX127.7
- **Про правий спеціальний модуль і серійний номер**
  1. Праві модулі розширення, такі як аналогові модулі та модулі температури, вважаються спеціальними модулями.
  2. Праві цифрові модулі можна налаштувати в програмному забезпеченні, і програмне забезпечення автоматично призначить пристрої введення та виведення для цифрових модулів.
  3. Серійний номер першого спецмодуля праворуч від контролера руху – 1; порядковий номер другого — 2 і так далі. Можна підключити до 16 спеціальних модулів. Початкова адреса для введення правого спеціального модуля — %MW10000, а початкова адреса для виведення правого спеціального модуля — %MW10500.

4.5 Карта пам'яті SD

4.5.1 Модель і специфікація

- **Модель і зовнішній вигляд**

Контролер руху підтримує лише такі карти Micro SD.



- **Специфікація**

У наведеній нижче таблиці показано члени родини SD-карт. На поточних ринках існує багато специфікацій SD-карт. Контролер підтримує лише категорію карт SDHC із файловою системою FAT32 і розмір Micro SDHC (15 мм X 11 мм X 1 мм). Будь ласка, переконайтеся, що ви придбали правильну специфікацію SD-карти, яку підтримує контролер.

- **Класифікація карт SD**

Категорія	SD	SDHC			SDXC	
Ємність	32 МБ~2 ГБ	4 ГБ ~ 32 ГБ			32 ГБ ~ 2 ТБ	
Файлова система	FAT16/FAT32	FAT32			exFAT （ FAT64 ）	
Розмір	SD	SDHC	Міні SDHC	Micro SDHC	SDXC	Micro SDXC
Рівень швидкості SD	N/A	КЛАС 2 (мін. 2 МБ/сек.) КЛАС 4 (мін. 4 МБ/сек.) КЛАС 6 (мін. 6 МБ/сек.) КЛАС 10 (мін. 10 МБ/сек.)			КЛАС 2 (мін. 2 МБ/с.) КЛАС 4 (мін. 4 Мб/с.) КЛАС 6 (Мін. 6 МБ/с.) КЛАС 10 (Мін. 10 МБ/с.)	

5

Розділ 5 Встановлення

Зміст

5.1 Dimensions..... 5-2

5.1.1 Профіль і розміри 5-2

5.1.2 Розміри правого модуля розширення 5-2

5.1.3 Підключення до правого модуля розширення 5-3

5.1.4 Встановлення та видалення SD-карти 5-4

5.1.5 Встановлення та заміна батареї кнопкового елемента 5-5

5.2 Installing the Module in the Control Cabinet 5-8

5.2.1 Встановлення модуля на DIN-рейку 5-8

5.2.2 Ілюстрація встановлення всередині шафи керування 5-8

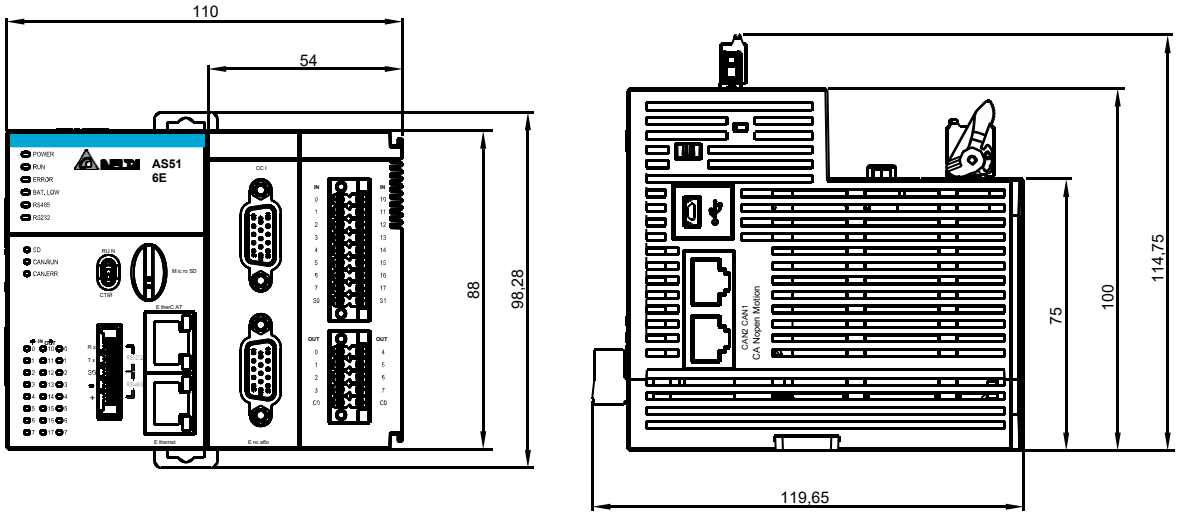
5.2.3 Температура навколишнього середовища в шафі керування 5-9

5.2.4 Дії щодо захисту від перешкод 5-9

5.2.5 Вимоги до розмірів у шафі керування 5-9

5.1 Розміри

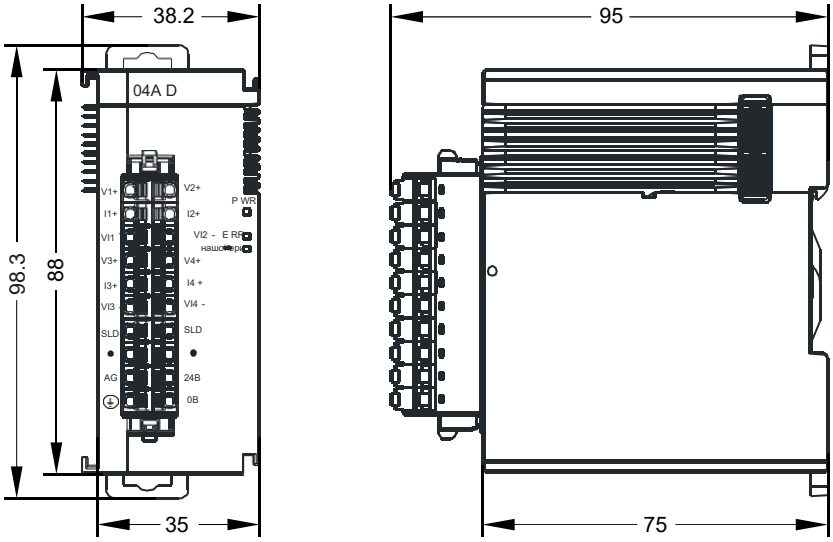
5.1.1 Профіль і розміри



Одиниця: мм

5.1.2 Розміри правого модуля розширення

- Подивіться наступний розмірний малюнок правого модуля розширення, взявши, наприклад, AS04AD-A. Довжина, ширина та висота всіх правих модулів такі ж, як у AS04AD-A.



Одиниця: мм

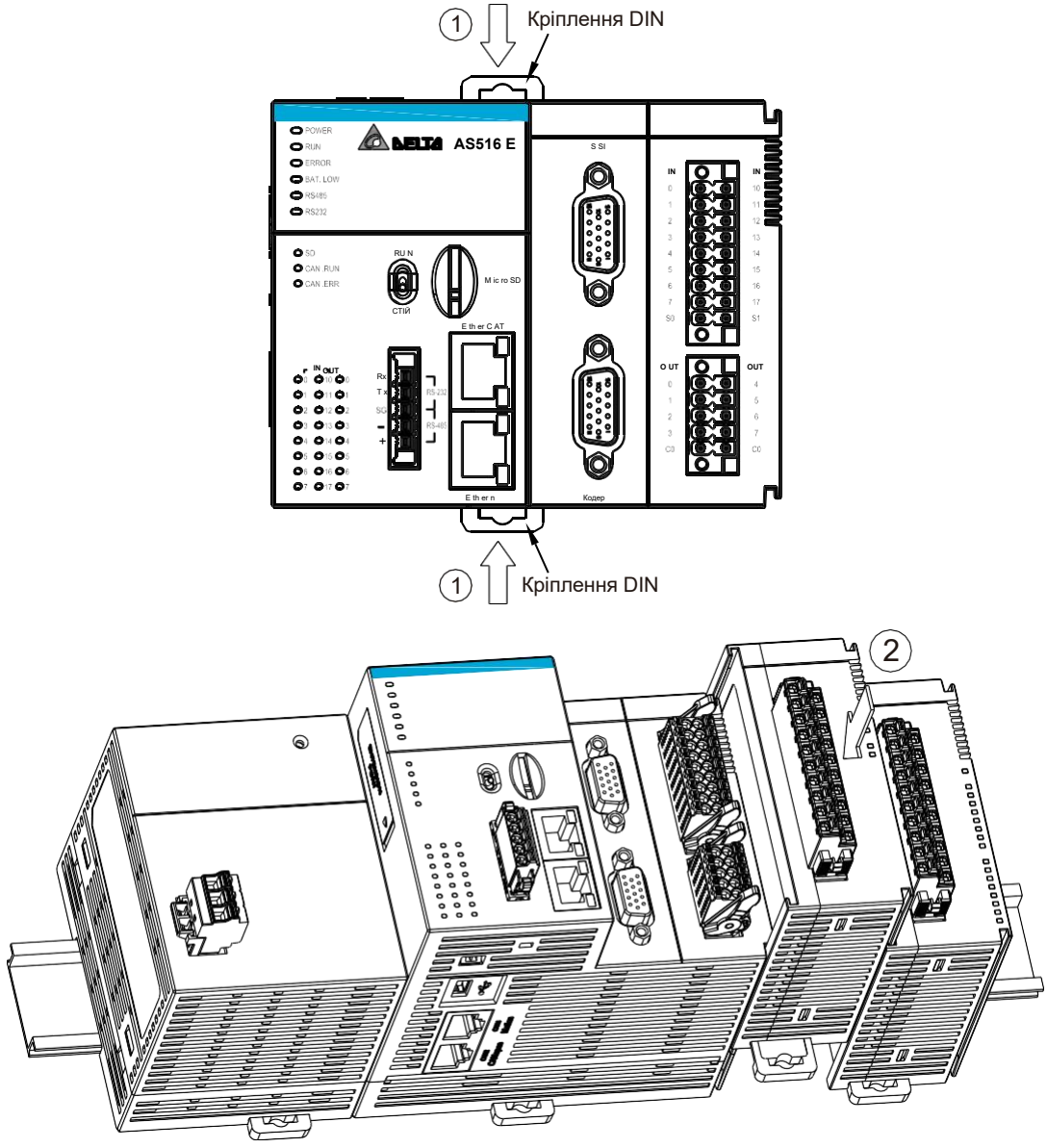
5.1.3 Підключення до правого модуля розширення

- Установка контролера руху та AS04AD-A на DIN-рейку

■ Натисніть на кріпильний затискач DIN на задній панелі серії AS500E у напрямку, указаному стрілкою ① , доки не чути клацання. Потім зробіть паз на задній частині модуля, спрямований на рейку, і натисніть на модуль, доки не почуєте клацання. Це означає, що модуль підключено до рейки.

■ Для встановлення другого модуля AS04AD-A натисніть на кріпильний затискач DIN AS04AD-A у напрямку, указаному стрілкою ① . Потім спрямуйте ліву канавку AS04AD-A на праву канавку серії AS500E та вставте AS04AD-A в серію AS500E. Після цього натисніть на модуль у напрямку стрілки ② , щоб зафіксувати його на рейці. Коли лунає звук клацання, це означає, що модуль закріплено на рейці та виконано з'єднання з контролером.

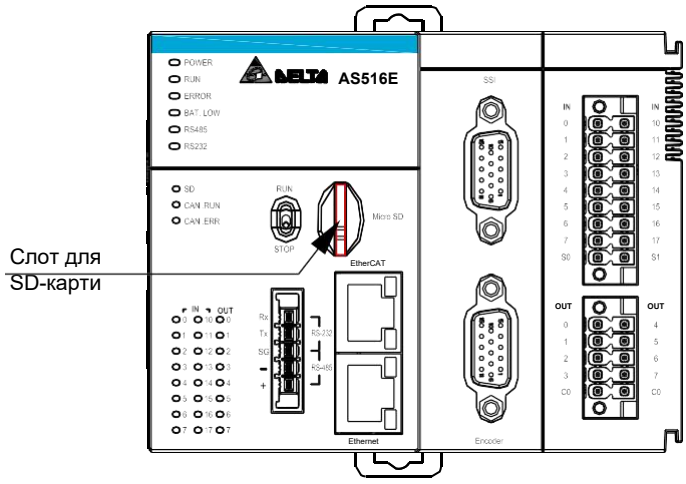
Таким же чином підключіть один за одним модулі вводу/виводу з правого боку контролера руху та закріпіть їх на рейці.



5.1.4 Встановлення та видалення SD-карти

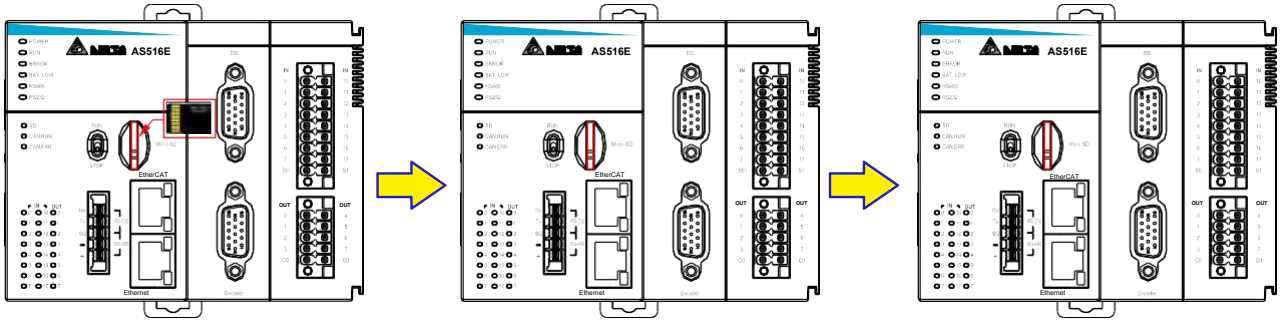
- Слот для карти пам'яті SD контролера руху

Гніздо для картки пам’яті розташовано на правій стороні передньої панелі контролера руху, як показано нижче.



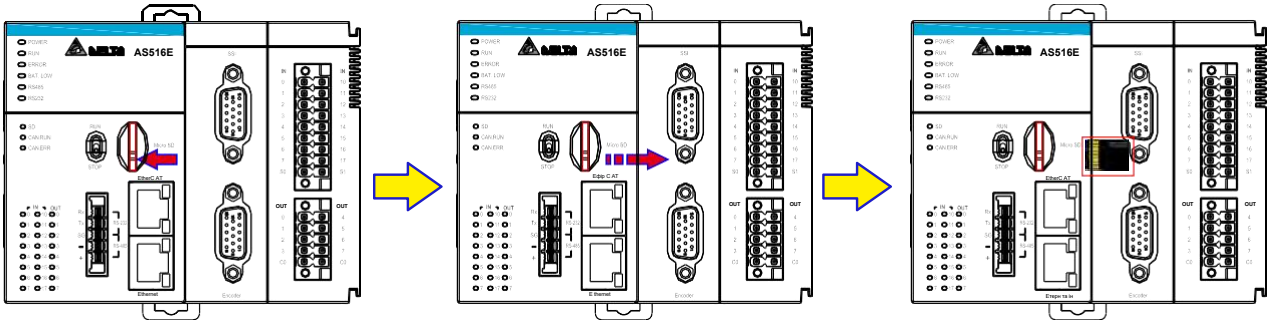
- Встановлення SD-карти

Вставте SD-карту безпосередньо в гніздо для карти пам’яті та натисніть її до кінця гнізда, доки не почуєте клацання. Після завершення встановлення SD-карту необхідно щільно закріпити. Якщо SD-карта, вставлена в слот, не закріплена, інсталяція не вдається. Крім того, SD-карта має захист від помилок. Якщо SD-карту вставлено неправильно, вона не досягне кінця слота. У цьому випадку не намагайтеся штовхати SD-карту до кінця слота, щоб уникнути пошкодження модуля та SD-карти.
Дотримуйтесь інструкцій на малюнках нижче, щоб вставити SD-карту в правильному напрямку.



- Видалення SD-карти

Просто проштовхніть SD-карту до кінця слота, щоб SD-карта послабилася та відскочила зсередини слота.
А потім легко вийміть SD-карту з гнізда.



5.1.5 Встановлення та заміна батареї

- монтаж

⚠ Warning

RTC (годинник реального часу) контролера руху не може працювати, якщо живлення батареї не подається належним чином. Контролер руху виключає акумулятор, коли він залишає завод. Таким чином, користувачі повинні встановити акумулятор CR1620 3V, який можна придбати на ринку самостійно.

Першу установку батареї можна виконати незалежно від того, увімкнено чи вимкнено контролер. Перед установкою батареї необхідно позбутися статичної електрики в корпусі, торкнувшись заземленого металу, або уникнути цього, одягнувши антистатичні рукавички. Після цього встановіть батарею. Після встановлення батареї RTC можна встановити за допомогою програмного забезпечення. Процедура встановлення акумулятора така.

1. Витягніть тримач батареї в контролері руху кінчиком викрутки з увігнутої частини відсіку для батареї, як показано на малюнку 5.1.5.1.

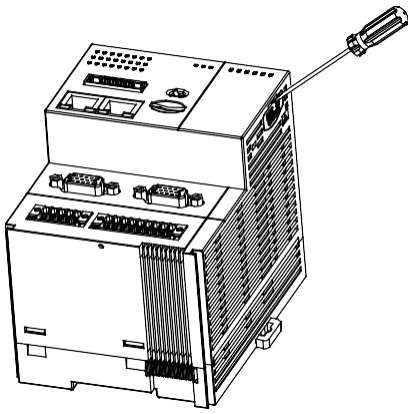
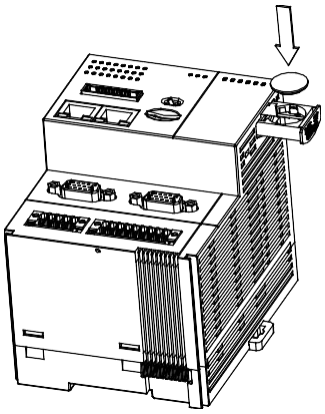


Рисунок 5.1.5.1

2. Вставте батарею CR1620 3 В у тримач батареї в напрямку, указаному стрілкою, як показано нижче.



Малюнок 5.1.5.2

3. Після того, як батарея покладена в тримач батареї, вставте тримач батареї назад у контролер руху, як показано на малюнку 5.1.5.3.

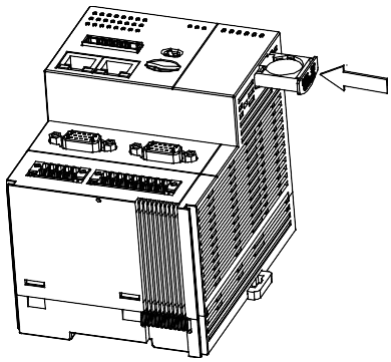


Рисунок 5.1.5.3

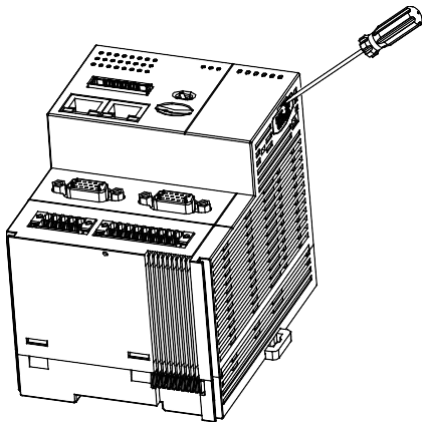
- Заміна

⚠ Warning

Якщо батарея ще не встановлена або напруга батареї низька, індикатор BAT LOW контролера руху світиться червоним, що вказує на те, що в контролер потрібно встановити батарею або стару батарею потрібно замінити на нову.

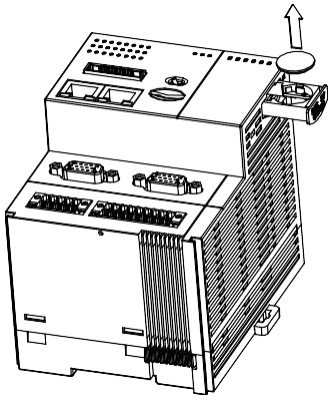
Ми рекомендуємо замінити батарею, коли контролер увімкнено, оскільки дані годинника реального часу будуть втрачені, якщо замінити батарею, коли контролер вимкнено. Перед заміною батареї необхідно позбутися статичної електрики в тілі, торкнувшись заземленого металу, або уникнути цього, одягнувши антистатичні рукавички. Після цього замініть батарею.

1. Витягніть тримач батареї в контролері руху кінчиком викрутки з увігнутої частини відсіку для батареї, як показано на малюнку 5.1.5.4.



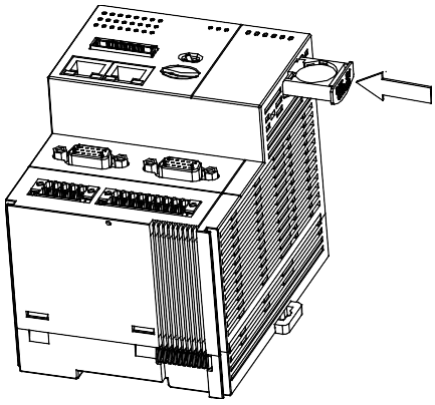
Малюнок 5.1.5.4

2. Вийміть стару батарею CR1620 3 В із тримача батареї в напрямку, указаному стрілкою нижче.



Малюнок 5.1.5.5

3. Вийнявши батарею, вставте нову та вставте тримач батареї назад у контролер руху, як показано на малюнку 5.1.5.6.

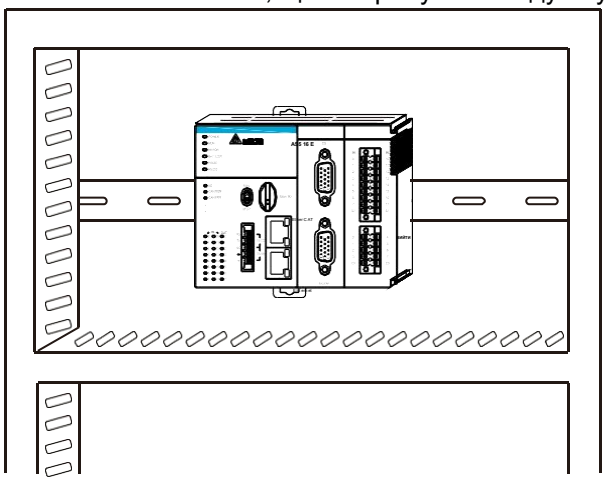


Малюнок 5.1.5.6

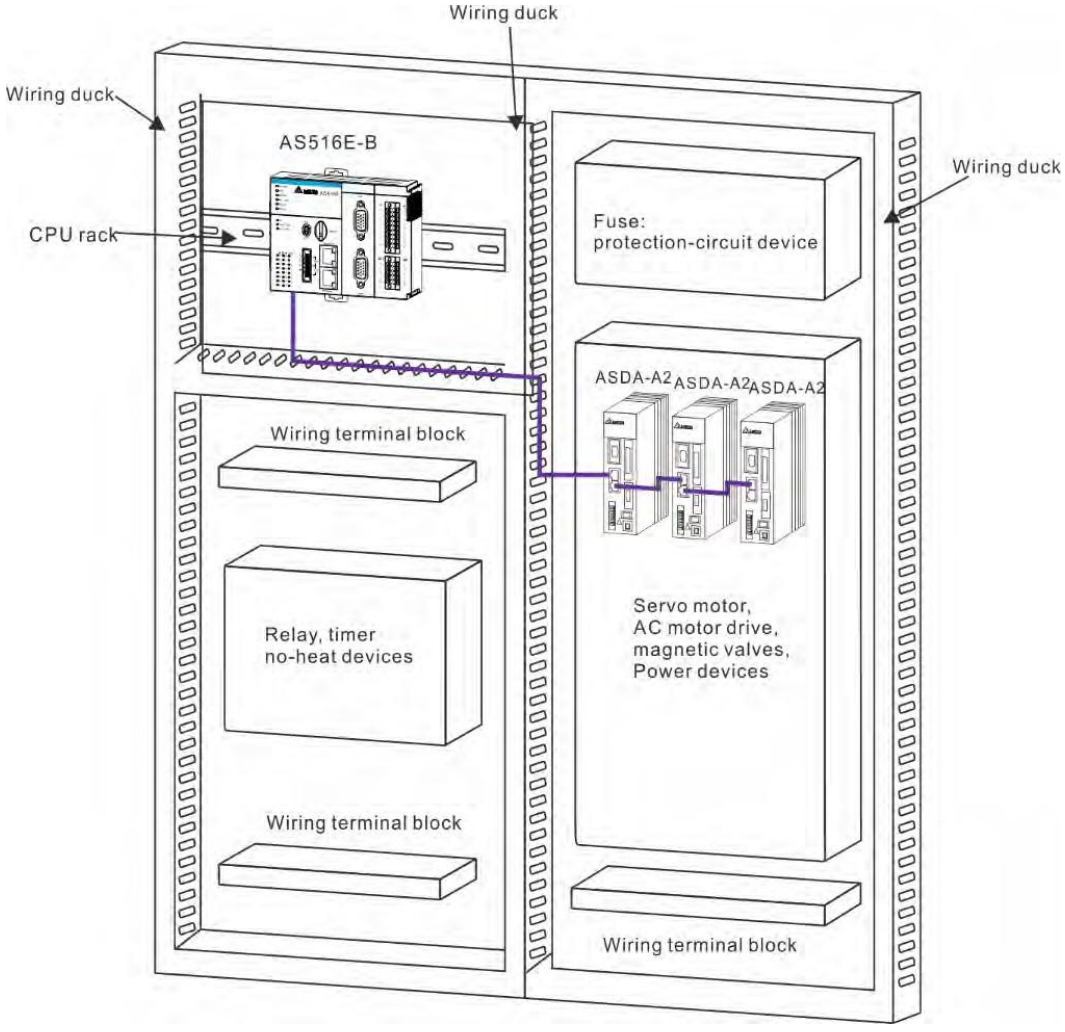
5.2 Встановлення модуля в шафу керування

5.2.1 Встановлення модуля на DIN-рейку

Витягніть затискачі на задній панелі контролера руху. Потім приклейте горизонтальні прорізи на задній частині модуля на DIN-рейку. Нарешті вставте затискачі, щоб зафіксувати модуль усередині шафи керування.



5.2.2 Ілюстрація встановлення всередині шафи керування



5.2.3 Температура навколишнього середовища в шафі керування

- **Вимоги**

⚠ Warning

- Температура навколишнього середовища контролера руху всередині шафи керування становить -20~55°C, а вологість — 5~95%.
- Будь ласка, уникайте установки поблизу високотемпературного обладнання.
- Залиште достатньо місця для вентиляції.
- Вентилятор або кондиціонер необхідно встановлювати, якщо температура навколишнього середовища перевищує 60°C.
- Контролер руху призначений лише для використання в приміщенні.
- Шафа керування висотою 1,0–2,0 м легша для встановлення та експлуатації.
- Тримайте установку подалі від високовольтного обладнання та силового обладнання.
- Перед установкою необхідно відключити живлення в шафі керування.

5.2.4 Дії щодо захисту від перешкод

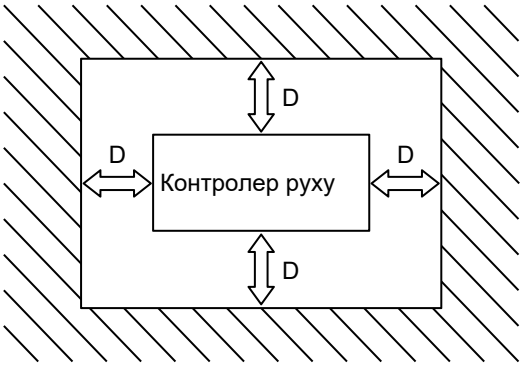
⚠ Warning

- Не встановлюйте контролер у шафі керування, яка містить високовольтне обладнання.
- Будь ласка, тримайте щонайменше 200 мм від проводу живлення для встановлення.
- Для шафи управління повинен бути провід заземлення.
- Якщо обладнання використовується у спосіб, не визначений виробником, захист, який забезпечує обладнання, може бути порушений.

5.2.5 Вимоги до розмірів у шафі керування

- **Установчий малюнок**

Контролер руху повинен бути встановлений у корпусі. Щоб контролер нормально випромінював тепло, простір між контролером і корпусом має бути більше 50 міліметрів. (D > 50 мм)



ПАМ'ЯТКА

6

частина 6 Електропроводка, налаштування зв'язку та побудова мережі

Зміст

6.1 Wiring 6-3
6.1.1 Джерело живлення 6-3
6.1.2 Електропроводка ланцюга безпеки 6-4
6.2 Input Point and Output Point Wiring 6-5
6.2.1 Функція, яку підтримують точки введення 6-5
6.2.2 Електропроводка вхідної точки 6-7
6.2.3 Електропроводка вихідної точки 6-9
6.3 RS-485 Communication Port..... 6-10
6.3.1 Функція, яку підтримує порт RS-485 6-10
6.3.2 Визначення контактів порту RS-485 6-10
6.3.3 Апаратне підключення RS-485 6-10
6.3.4 Підтримувані функціональні коди та коди винятків 6-11
6.4 RS-232 Communication Port..... 6-12
6.4.1 Функція, яку підтримує порт RS-232 6-12
6.4.2 Визначення контактів порту RS-232 6-13
6.4.3 Апаратне підключення RS-232 6-13
6.4.4 Підтримувані функціональні коди та коди винятків 6-13
6.5 SSI Absolute Encoder Port..... 6-14
6.5.1 Функція абсолютного кодера SSI 6-14
6.5.2 Визначення контактів порту SSI 6-15
6.5.3 Підключення апаратного забезпечення абсолютного кодера SSI 6-15
6.6 Incremental Encoder Port..... 6-16
6.6.1 Функція інкрементального кодера 6-16
6.6.2 Визначення контактів порту інкрементального кодера 6-16
6.6.3 Апаратне підключення інкрементального енкодера 6-18
6.7 Ethernet Communication Port..... 6-19
6.7.1 Функція, яку підтримує комунікаційний порт Ethernet 6-19
6.7.2 Контакти комунікаційного порту Ethernet 6-20
6.7.3 Мережеве з'єднання комунікаційного порту Ethernet 6-20
6.7.4 Коди функцій, які підтримує комунікаційний порт Ethernet 6-20

6.8 EtherCAT Communication Port 6-21
6.8.1 Функція, яку підтримує комунікаційний порт EtherCAT 6-21
6.8.2 Контакти комунікаційного порту EtherCAT 6-21
6.8.3 Підключення до мережі комунікаційного порту EtherCAT 6-22
6.8.4 Відстань зв'язку EtherCAT 6-23
6.9 CANopen Communication Port 6-23
6.9.1 Функції, які підтримує комунікаційний порт CANopen 6-23
6.9.2 Контакти комунікаційного порту CANopen 6-24
6.9.3 Відображення PDO на комунікаційному порту CANopen 6-24
6.9.4 Підключення до мережі через комунікаційний порт CANopen 6-24
6.9.5 Швидкість зв'язку CANopen і відстань зв'язку 6-25

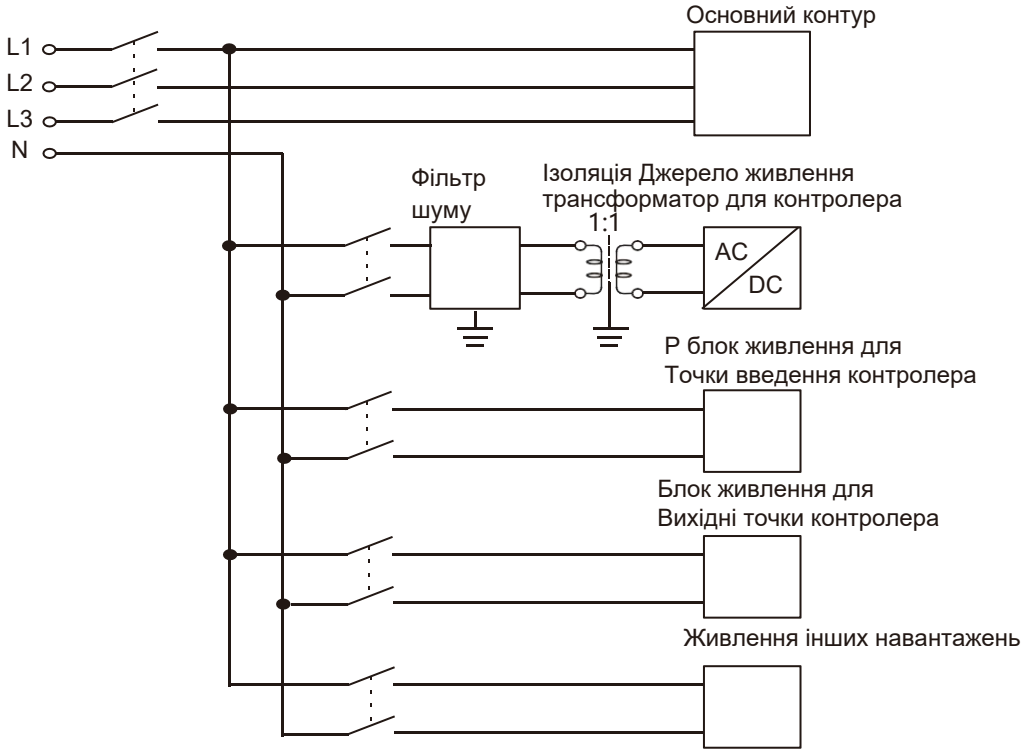
6.1 Електропроводка

6.1.1 Джерело живлення

Вхідна напруга контролера руху становить 24 В постійного струму. Під час використання зверніть увагу на наступні моменти.

⚠ Warning

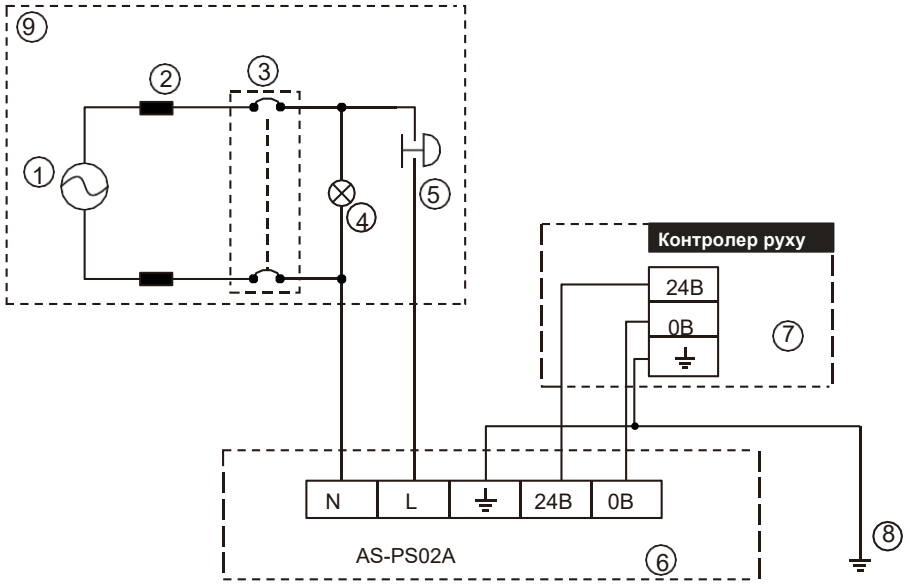
- Підключіть джерело живлення до двох клем, 24 В і 0 В, і клему заземлення до землі. Будьте обережні, контролер руху може бути пошкоджений, якщо позитивна та негативна полярності джерела живлення підключені навпаки.
- Обов’язково використовуйте сертифіковане джерело живлення з виходом SELV або сертифіковане джерело живлення з подвійною ізоляцією, оцінене за стандартами UL60950 або UL61010-1 і UL61010-2-201
- В якості проводів живлення використовуйте тільки мідні провідники. Діаметр проводів живлення має бути від 12 до 28 AWG, а номінальна температура має бути вище 70°C. Момент затягування проводки вилки клемної колодки живлення становить 4,5 дюйм-фунт.
- Розділіть джерело живлення для контролера, зовнішнє джерело живлення для точок входу та виходу контролера та джерело живлення для інших навантажень. Контролер використовує індивідуальне джерело живлення.
- Додайте фільтр шуму та ізоляційний трансформатор перед живленням контролера. Ізоляційний трансформатор встановлюється між джерелом живлення та фільтром перешкод. Вторинну обмотку трансформатора не потрібно заземлювати. Дивіться схему підключення нижче.
- Ізоляційний трансформатор ефективно зменшує перешкоди від землі, електричний шум тощо. Силові дроти між ізоляційним трансформатором і контролером повинні бути щільно скручені. Чим менша відстань між ними, тим кращий ефект. Обов’язково тримайтеся подалі від ліній електропередач і ліній високої напруги.
- Шумовий фільтр може ефективно зменшити перешкоди. Вхідний кабель і вихідний кабель фільтра слід розташовувати окремо, щоб уникнути перешкод перед шумовим фільтром і кабелем після шумового фільтра.
- Тривале відключення живлення або падіння напруги призупинить роботу контролера руху, і контролер припинить обмін даними з сервоприводом, коли всі виходи стануть FALSE. Контролер руху відновить зв'язок із сервоприводом, коли джерело живлення повернеться до нормального стану.



6.1.2 Електропроводка ланцюга безпеки

⚠ Warning

Дія будь-якого пристрою всередині контролера руху може вплинути на поведінку зовнішнього обладнання під керуванням контролера руху над сервоприводом. Тому будь-яка несправність пристрою може призвести до втрати керування всією системою автоматичного керування і навіть до травм і загибелі персоналу. З цих причин ми пропонуємо включити в ланцюг вхідного живлення наступний пристрій безпеки.



Малюнок 3.2.1

① Джерело живлення змінного струму: 100 ~ 240 В змінного струму; 50/60 Гц.
② Запобіжник захисту ланцюга живлення
③ Пристрій ізоляції ланцюга системи: електромагнітний контактор, реле та інший перемикач можна використовувати як ізоляційний пристрій, щоб запобігти нестабільності системи, коли джерело живлення переривається.
④ Індикатор живлення
⑤ Кнопка аварійної зупинки: кнопка відключає живлення системи, коли виникає випадкова надзвичайна ситуація.
⑥ Модуль живлення Delta AS-PS02/24VDC (AS-PS02 рекомендовано для контролера руху)
⑦ Контролер руху
⑧ Земля
⑨ Схема безпеки

6.2 Електропроводка точки входу та точки виходу

6.2.1 Функція, яку підтримують точки введення

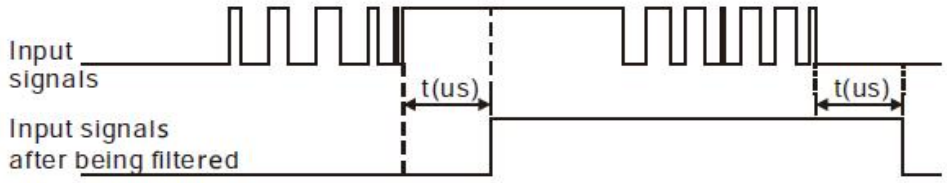
Є 16 вхідних точок, які підтримують функції зовнішнього переривання та фільтрації в контролері руху. Крім того, вхідні точки можна використовувати для фіксації положення кодера. Зверніться до пояснення до інструкції DMC_TouchProbe, щоб отримати докладні відомості про захоплення позиції.

- **Принцип роботи вхідного фільтра**

  Вхідний фільтр фільтрує короткі імпульсні сигнали через 16 точок I0~I7 та I10~I17, щоб зменшити вплив вхідних сигналів перешкод. Збільшення значення фільтра може зменшити вібрацію вхідних сигналів або вплив зовнішніх перешкод.

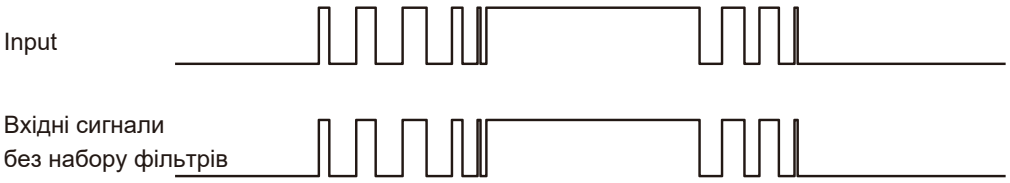
Час вхідного фільтра: t=31us * （ 0~255 ） . Таким чином, час фільтрації є кратним 31 мкс, і 0 є значенням за замовчуванням. Час вхідного фільтра можна встановити за допомогою програмного забезпечення.

- **Коли є встановлений фільтр:**
  Коли час фільтра встановлено на t (us), сигнал є дійсним, якщо час увімкнення або вимкнення вхідного сигналу перевищує t (us). Якщо час увімкнення або вимкнення вхідного сигналу менше t (us), сигнал буде усунено. Вхідний сигнал, який залишився після фільтрації, буде введений після затримки на t (us).



Малюнок 6.2.1.1

- **Якщо немає набору фільтрів:**
  Вхідні сигнали не змінюються, якщо не встановлено час фільтра.



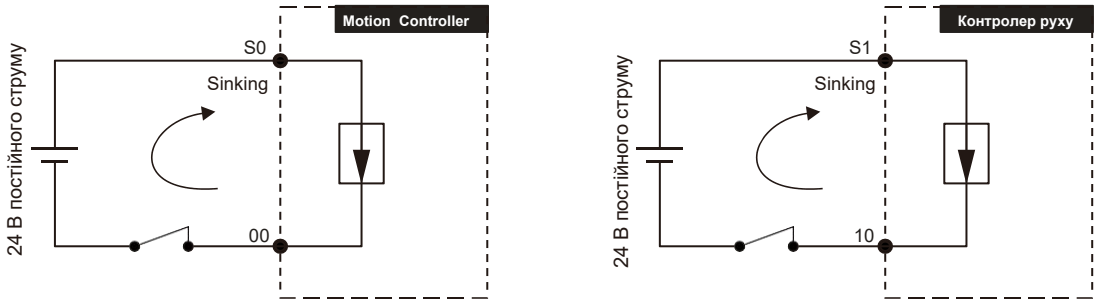
Малюнок 6.2.1.2

6.2.2 Розводка вхідної точки

Існує два типи входів постійного струму: SINK і SOURCE. Перегляньте деталі проводки в наступних двох режимах.

● **Режим раковини**

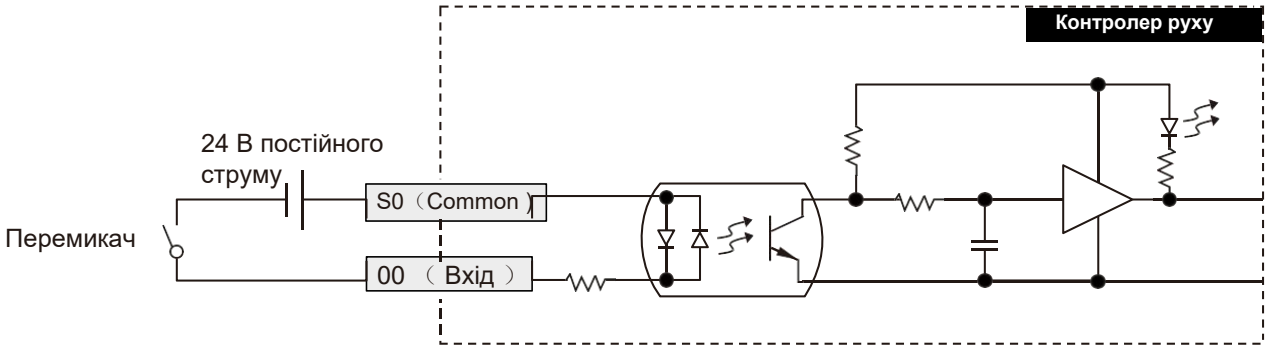
У режимі приймача спрощена модель показана нижче, і струм тече в загальні порти S0 і S1.



Малюнок 6.2.2.1

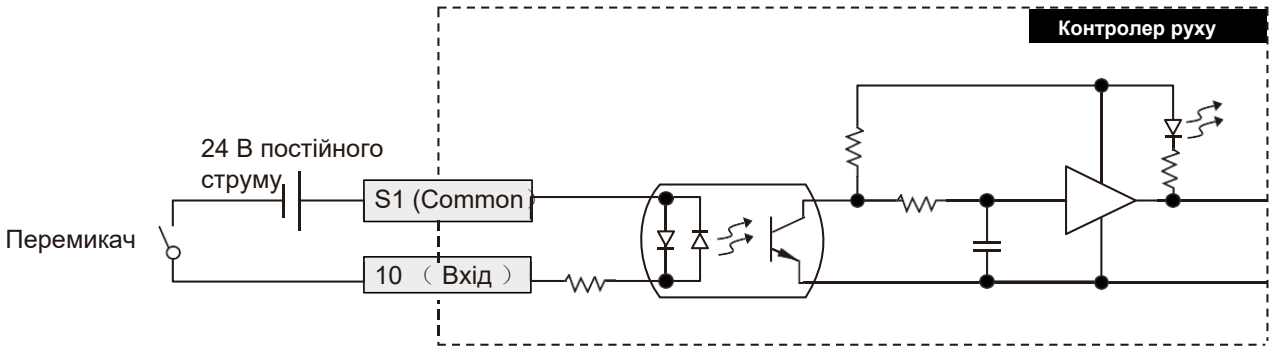
Дивіться відповідну схему електропроводки на наступних малюнках.

■ Вхідні точки контролера руху 00~07 відповідають S0, як показано нижче.



Малюнок 6.2.2.2

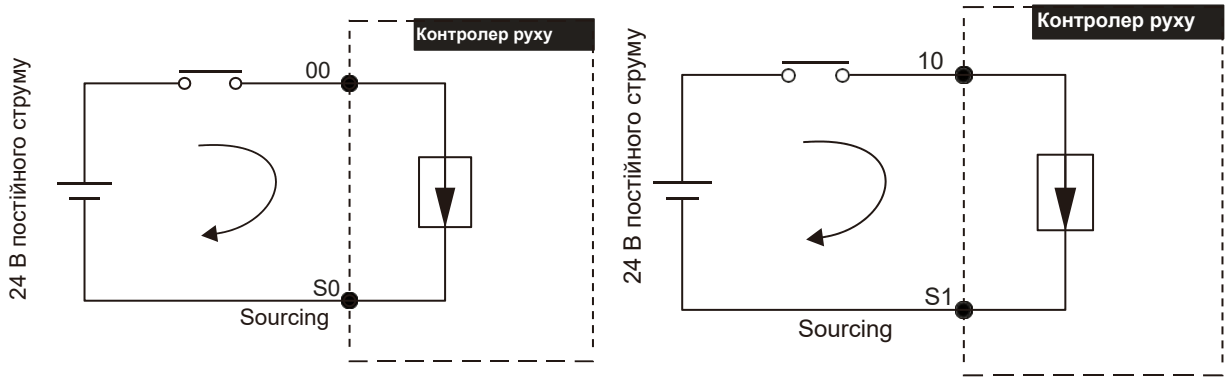
■ Вхідні точки контролера руху 10~17 відповідають S1, як показано нижче.



Малюнок 6.2.2.3

● **Режим Source**

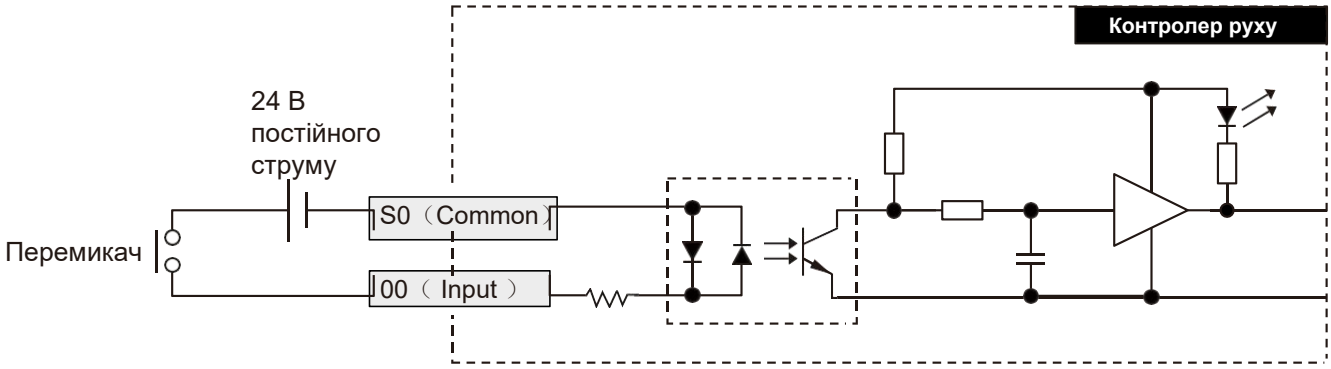
У режимі джерела спрощена модель показана нижче, і струм тече в загальні порти S0 і S1.



Малюнок 6.2.2.4

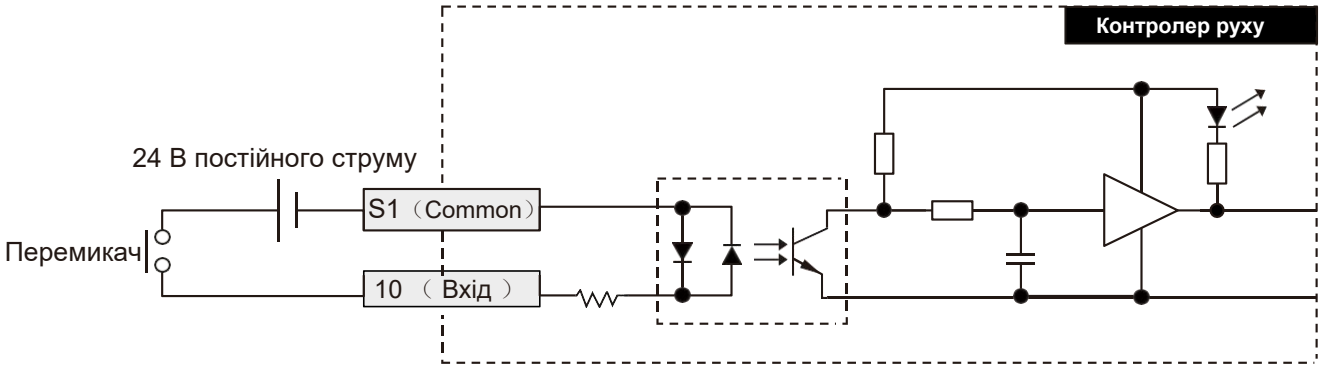
Дивіться схему підключення нижче

■ Вхідні точки контролера руху 00~07 відповідають S0, як показано нижче.



Малюнок 6.2.2.5

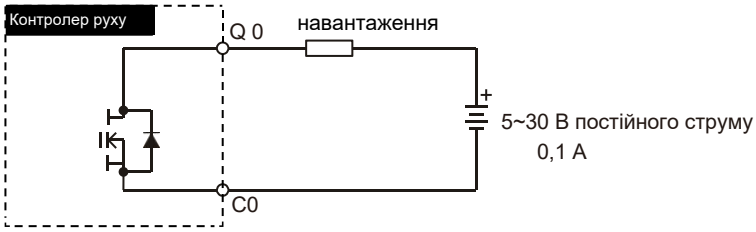
■ Вхідні точки контролера руху 10~17 відповідають S1, як показано нижче.



Малюнок 6.2.2.6

6.2.3 Електропроводка вихідної точки

Усі локальні вихідні точки контролера руху є транзисторними. Схеми підключення показані нижче.



6.3 Комунікаційний порт RS-485

6.3.1 Функція, яку підтримує порт RS-485

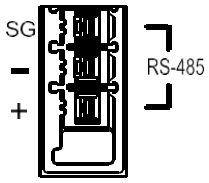
Комунікаційний порт RS-485 контролера руху може функціонувати як провідний або ведений Modbus. HMI, PLC або інший головний пристрій Modbus може зчитувати та записувати дані в пристрої всередині контролера руху. Інтервал часу, коли головний модуль Modbus отримує доступ до контролера руху, повинен перевищувати 5 мс.

Програму неможливо завантажити через порт RS-485. RS-485 підтримує протокол Modbus, ASCII, а також режим RTU. Коди функцій, які підтримує порт RS-485, включають 16#01, 16#02, 16#03, 16#05, 16#06, 16#0F і 16#10. Адреси станцій, які підтримує порт RS-485: 1~255. Функція трансляції не підтримується.

Зверніться до додатку A, щоб дізнатися більше про зв'язок Modbus і адреси пристроїв Modbus.

6.3.2 Визначення контактів порту RS-485

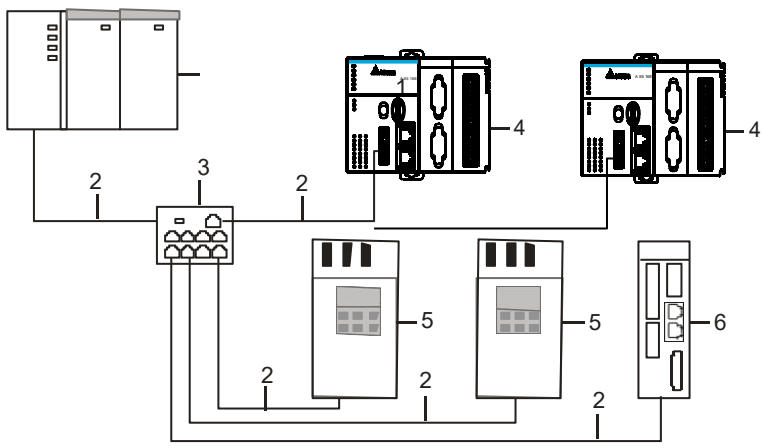
Див. таблицю нижче для визначення відповідних контактів порту зв'язку RS-485.

PIN-№	Сигнал	Визначення	
+	D+	Позитивний полюс RS-485	
-	D-	Мінусовий полюс RS-485	
SG	SG	Заземлення сигналу RS-485	

6.3.3 Апаратне підключення RS-485

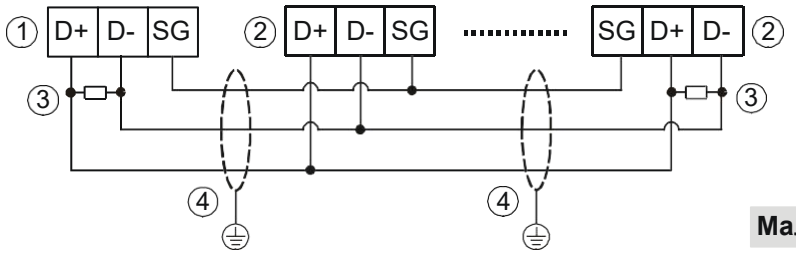
- **Приклад підключення контролера руху до мережі Modbus**

Контролер підключений до мережі Modbus через RS-485, як показано нижче.



№ пристрою	1	2	3	4	5	6
Ім'я пристрою	Майстер Modbus	Кабель зв'язку	VFD-CM08	Контролер	AC моторний привід	Серво привід

● **Електропроводка RS-485:**



Малюнок 19

Пояснення чисел

①	②	③	④
майстер	підлеглий	Кінцевий резистор	Екранований кабель

Примітки:

1. На обох кінцях шини рекомендується підключати кінцеві резистори номіналом 120 Ом.
2. Для забезпечення високої якості зв’язку використовуйте екрановану виту пару (20AWG).
3. Якщо внутрішні напруги двох пристроїв відрізняються, зробіть SG (сигнальне заземлення) двох пристроїв, з’єднаних один з одним, щоб збалансувати їхні напруги SG і зробити зв’язок більш стабільним.

● **Формат зв'язку, який підтримує RS-485**

Комунікаційний порт RS-485 підтримує формати зв’язку ASCII або RTU, а підтримувана швидкість передачі може становити до 115200 біт/с.

Швидкість передачі даних	9600, 19200, 38400, 57600, 115200					
Режим	ASCII				RTU	
Формат спілкування	7,E,1	7,E,2	7,N,1	7,N,2	8,E,1	8,E,2
	7,O,1	7,O,2	8,E,1	8,E,2	8,N,1	8,N,2
	8,N,1	8,N,2	8,O,1	8,O,2	8,O,1	8,O,2

6.3.4 Підтримувані коди функцій і коди винятків

● **Коди функцій Modbus:**

1. Коди функцій, які підтримує порт RS-485 контролера руху, перераховані в наступній таблиці.

Код функції	Індикація	Чи транслювати (Так/Ні)	Макс. кількість доступних для запису / читання регістрів	Наявний реєстр
16#01	Читання вихідних значень бітового регістра.	N	256	Бітовий регістр
16#02	Читання значень бітового регістра.	N	256	Бітовий регістр
16#03	Читання одного або кількох значень регістра слів.	N	100	Реєстр слів
16#05	Записати одне значення регістра одного біта.	Y	1	Бітовий регістр
16#06	Запишіть одне значення регістра слова.	Y	1	Реєстр слів
16#0F	Записати кілька значень бітового регістра.	Y	256	Бітовий регістр

Код функції	Індикація	Чи транслювати (Так/Ні)	Макс. кількість доступних для запису/читання регістрів	Наявний реєстр
16#10	Записати кілька значень реєстру слів.	Y	100	Реєстр слів

2. Коди винятків, які підтримує порт RS-485 контролера руху, перераховані в наступній таблиці.

Код відповіді винятку	Індикація
16#01	Непідтримуваний код функції
16#02	Непідтримувана адреса Modbus
16#03	Довжина даних виходить за допустимий діапазон.

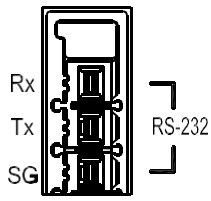
6.4 Комунікаційний порт RS-232

6.4.1 Функція, яку підтримує порт RS-232

Комунікаційний порт RS-232 контролера руху може функціонувати як провідний або ведений Modbus. HMI, PLC або інший пристрій Modbus може зчитувати та записувати дані в пристрої всередині контролера руху. Програму неможливо завантажити через порт RS-232. RS-232 підтримує протокол Modbus, режим ASCII, а також режим RTU. Коди функцій, які підтримує порт RS-232, включають 16#01, 16#02, 16#03, 16#05, 16#06, 16#0F і 16#10. Адреси станцій, які підтримує порт RS-232: 1~255. Функція трансляції не підтримується. Зверніться до додатку A, щоб дізнатися більше про зв'язок Modbus і адреси пристроїв Modbus.

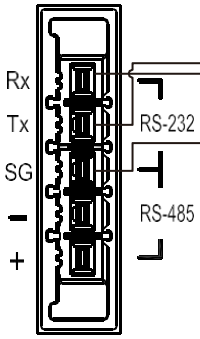
6.4.2 Визначення контактів порту RS-232

Порт SSI контролера руху складається з 15 контактів. Див. таблицю нижче для визначення відповідних контактів порту зв’язку RS-232.

PIN-№	Сигнал	Визначення	
Rx	Rx	Отримання даних	
Tx	Tx	Передача даних	
SG	GND	Сигнальна земля	

6.4.3 Апаратне підключення RS-232

AS516E-B
(Проводка порту R S-232)

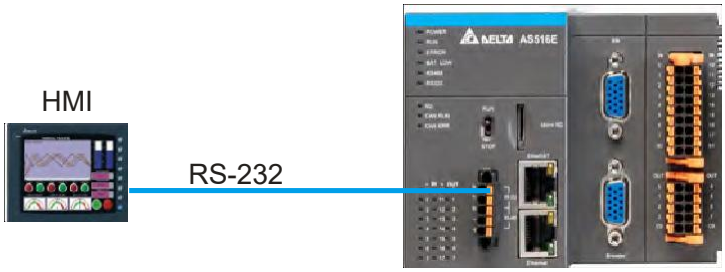


Виріб RS-232, підключений до AS516

Функція	Pin
1	
2	RXD
3	TX D
4	DTR
5	GND
4	DSR
5	RTS
6	CTS
9	

- Порт RS-232 підключається до HMI, коли контролер руху працює як ведений.

AS516E-B



- **Формат зв’язку, який підтримує RS-232**

Швидкість передачі даних	9600 , 19200 , 38400 , 57600 , 115200					
Режим	ASCII				RTU	
Формат спілкування	7,E,1	7,E,2	7,N,1	7,N,2	8,E,1	8,E,2
	7,O,1	7,O,2	8,E,1	8,E,2	8,N,1	8,N,2
	8,N,1	8,N,2	8,O,1	8,O,2	8,O,1	8,O,2

6.4.4 Підтримувані коди функцій і коди винятків

- **Коди функцій Modbus:**

1. Функціональні коди, які підтримує порт RS-232 контролера руху, перераховані в наступній таблиці.

Код функції	Індикація	Макс. кількість доступних для запису / читання регістрів	Наявний реєстр
16#01	Читання вихідних значень бітового регістра.	256	Бітовий регістр
16#02	Читання значень бітового регістра.	256	Бітовий регістр
16#03	Читання одного або кількох значень регістра слів.	100	Реєстр слів
16#05	Записати одне значення регістра одного біта.	1	Бітовий регістр
16#06	Запишіть одне значення регістра слова.	1	Реєстр слів
16#0F	Записати кілька значень бітового регістра.	256	Бітовий регістр
16#10	Записати кілька значень реєстру слів.	100	Реєстр слів

2. Коди винятків, які підтримує порт RS-232 контролера руху, перераховані в наступній таблиці.

Код винятку	Індикація
16#01	Непідтримуваний код функції
16#02	Непідтримувана адреса Modbus
16#03	Довжина даних виходить за допустимий діапазон.

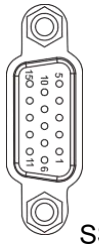
6.5 Порт абсолютного кодера SSI

6.5.1 Функція абсолютного кодера SSI

Порт SSI контролера руху є 15-контактним інтерфейсом D-SUB, який можна використовувати для підключення кодера SSI. Крім того, порт також включає вихід живлення 5 В (400 мА), який забезпечує живлення кодера. Ви можете використовувати вісь кодера SSI для керування рухом підлеглих осей відповідно до кількості імпульсів, отриманих через порт кодера.

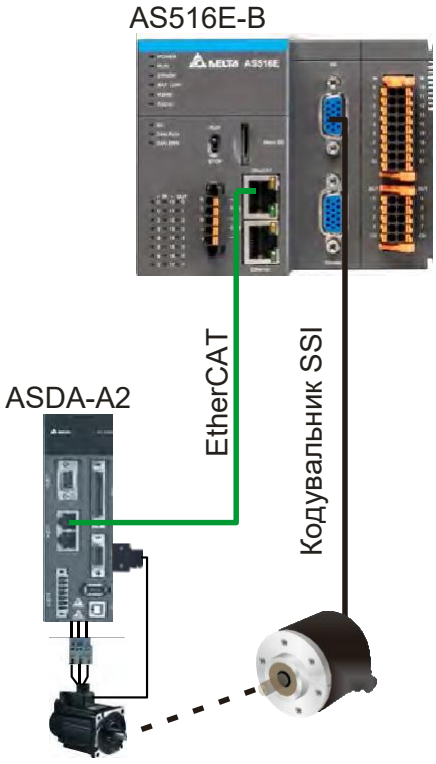
6.5.2 Визначення контактів порту SSI

Порт SSI контролера руху є 15-контактним інтерфейсом D-SUB. Дивіться таблицю нижче для визначення відповідних контактів комунікаційного порту SSI.

PIN-№	Сигнал	Визначення	
1	DATA+	Позитивний полюс даних абсолютного кодера	 SSI
2	DATA-	Негативний полюс даних абсолютного кодера	
6	CLK+	Позитивний полюс годинника абсолютного кодера	
14	CLK-	Негативний полюс годинника абсолютного кодера	
8	GND	Заземлення живлення абсолютного датчика	
15	5B	Живлення абсолютног кодера	

6.5.3Підключення апаратного забезпечення абсолютного кодера SSI

- Ілюстрація підключення датчика абсолютного значення SSI



- Специфікація підключення інтерфейсу абсолютного кодера SSI
  Інтерфейс кодера SSI контролера руху та спосіб підключення показані нижче.



AS516E-B
SSI



SSI кодер

Pin	функція
1	DATA+
2	DATA-
6	CLK +
14	CLK -
15	5B
8	G ND

функція
DATA+
DATA-
CLK +
CLK -
VCC
0B

Примітка. Джерело живлення для порту SSI контролера руху становить 5 В.

Коли VCC = 5 В, підключіть напругу живлення VCC кодера SSI до контакту 15 інтерфейсу SSI і 0 В кодера SSI до контакту 8 інтерфейсу SSI.

Коли VCC ≠ 5 В, подайте живлення на кодер SSI окремо відповідно до фактичної напруги живлення підключеного кодера SSI.

- Специфікація для кабелю зв'язку абсолютного кодера SSI
  Для передачі сигналу CLK+, CLK-, DATA+ і DATA- використовуйте екранований кабель зі скрученою парою.

6.6 Інкрементальний порт кодера

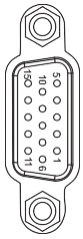
6.6.1 Функція інкрементального кодера

Порт інкрементного кодера контролера руху є 15-контактним інтерфейсом D-SUB, який може підключати два незалежні інкрементні кодери. Обидва порти кодера підтримують вхід диференціального сигналу з максимальною робочою частотою 1 МГц (250Kx 4 = 1 МГц) на один. Крім того, порт інтегрує два виходи живлення 5 В (400 мА) для живлення двох кодерів. Ви можете створити інкрементну вісь кодера для будь-якого з двох кодерів, щоб керувати рухом підлеглих осей відповідно до кількості імпульсів, отриманих на порт кодера.

6.6.2 Визначення контактів порту інкрементального кодера

Порт інкрементного кодера контролера руху є 15-контактним інтерфейсом. Перегляньте таблицю нижче для визначення відповідних контактів комунікаційного порту кодера.

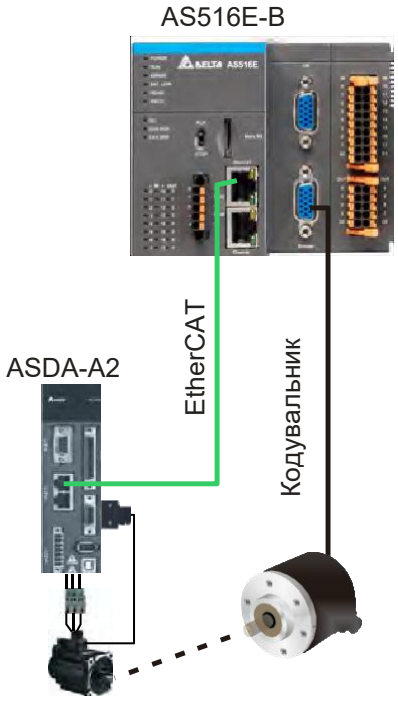
PIN-№	Сигнал	Визначення	
1	A1+	Диференціальні сигнали першого інкрементальний кодер	
2	A1-		
10	B1+		
11	B1-		
4	Z1+		
5	Z1-		
15	+5B	Блок живлення для першого кодера	
3	A2+	Диференціальні сигнали другого інкрементного кодера	
9	A2-		
6	B2+		
12	B2-		
13	Z2+		
14	Z2-		
7	+5B	Блок живлення для другого кодера	
8	0B	0 В, спільний між двома кодерами	
Зовнішня оболонка		Екрануючий шар	



Кодувальник

6.6.3 Інкрементне підключення апаратного кодера

- Ілюстрація підключення інкрементного кодера



- Специфікація підключення порту інкрементального кодера
  Інтерфейс інкрементального кодера контролера руху та спосіб підключення показані нижче.



A S516E- B
Порт кодера



Кодувальник

Pin	Функція
1	A1+
2	A1-
10	B1+
11	B1-
4	Z1+
5	Z1-
15	+5B
8	GND

Функція
A
$\overline{A}$
B
$\overline{B}$
Z
$\overline{Z}$
Vcc
0B

Примітка. Джерело живлення порту кодера контролера руху становить 5 В.

Коли VCC = 5 В, підключіть напругу живлення VCC кодера до контакту 15 інтерфейсу кодера контролера руху та 0 В кодера до контакту 8 інтерфейсу кодера.

Коли VCC ≠ 5 В, живлення подається тільки на кодер відповідно до фактичної напруги живлення підключеного кодера.

6.7 Комунікаційний порт Ethernet

6.7.1 Функція, яку підтримує комунікаційний порт Ethernet

Комунікаційний порт Ethernet у контролері руху підтримує протокол Modbus TCP і може працювати як головний, так і підлеглий.

Порт Ethernet можна використовувати для завантаження файлу конфігурації, файлу виконання та файлу CAM. Він також підтримує функцію автоматичної перемички, і користувачам не потрібно спеціально вибирати дротову перемичку, коли порт Ethernet підключено до комп’ютера або комутатора. Крім того, він може автоматично визначати швидкість передачі 10 Мбіт/с і 100 Мбіт/с.

HMI, PLC або інший провідний пристрій Modbus TCP може читати та записувати дані в пристрої всередині контролера руху через порт Ethernet. Докладніше про зв’язок Modbus TCP див. у додатку B.

Порт Ethernet підтримує функцію EtherNet/IP slave і функцію Socket. Подробиці щодо специфікації порту див. нижче.

Пункт			Порт Ethernet
Протокол зв'язку			MODBUS TCP, Socket, EtherNet/IP
MODBUS TCP	Підключення (сервер)		16
	Підключення (клієнт)		
Розетка	TCP з'єднання		8
	UDP з'єднання		
EtherNet/IP	Тип пристрою		Перехідник
	CIP_IO Підключення	CIP з'єднання	8
		TCP з'єднання	16
		Інтервал надсилання повідомлень (RPI)	5 мс ~ 1000 мс
		Максимальний розмір даних на одне повідомлення	200 байт
	CIP_Відверте повідомлення	Клас 3 (підключений тип)	8
		UCMM (Непідключений тип)	16
		Підтримує об'єкти CIP	Ідентифікація, маршрутизатор повідомлень, збірка, диспетчер з’єднань, порт, інтерфейс TCP/IP, канал Ethernet, залежно від постачальника

Коли контролер служить підлеглим EtherNet/IP, з’єднання вводу/виводу відповідають початковим пристроям за замовчуванням, як показано в наступній таблиці.

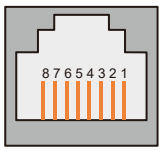
Запустіть пристрій / Підключення IO	Пристрій запуску за замовчуванням для отримання даних від головного	Пристрій запуску за замовчуванням для надсилання даних головному
Підключення 1	%MW11000	%MW12000
Підключення 2	%MW11100	%MW12100
Підключення 3	%MW11200	%MW12200
Підключення 4	%MW11300	%MW12300

Запустіть пристрій / Підключення IO	Пристрій запуску за замовчуванням для отримання даних від головного	Пристрій запуску за замовчуванням для надсилання даних головному
Підключення 5	%MW11400	%MW12400
Підключення 6	%MW11500	%MW12500
Підключення 7	%MW11600	%MW12600
Підключення 8	%MW11700	%MW12700

6.7.2 Контакти комунікаційного порту Ethernet

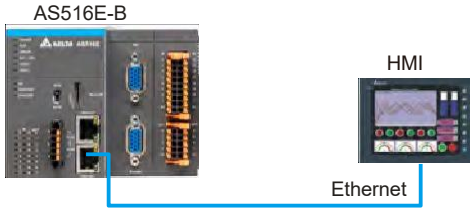
IP-адреса за замовчуванням для порту зв'язку Ethernet – 192.168.1.1. Визначення контактів порту зв'язку Ethernet див. у таблиці нижче.

PIN-№	Сигнал	Визначення
1	Tx+	Позитивний полюс для передачі даних
2	Tx-	Негативний полюс для передачі даних
3	Rx+	Позитивний полюс для отримання даних
4	Зарезервований	Зарезервований
5	Зарезервований	Зарезервований
6	Rx-	Негативний полюс для отримання даних
7	Зарезервований	Зарезервований
8	Зарезервований	Зарезервований



Ethernet

6.7.3 Мережеве з'єднання комунікаційного порту Ethernet



6.7.4 Коди функцій, які підтримує комунікаційний порт Ethernet

Нижче наведено список функціональних кодів і кодів відповіді на виняткові ситуації, які підтримуються, коли комунікаційний порт Ethernet контролера руху використовує протокол Modbus TCP.

Код функції	Індикація	Макс. кількість доступних для запису/читання регістрів	Наявний реєстр
16#01	Зчитування вихідних значень бітового регістра.	256	Бітовий регістр
16#02	Читання значень бітового регістра.	256	Бітовий регістр
16#03	Читання одного або кількох значень регістра слів.	100	Реєстр слів
16#05	Записати одне значення регістра одного біта.	1	Бітовий регістр
16#06	Запишіть одне значення регістра слова.	1	Реєстр слів

Код функції	Індикація	Макс. кількість доступних для запису/читання регістрів	Наявний реєстр
16#0F	Записати кілька значень бітового регістра.	256	Бітовий регістр
16#10	Записати кілька значень реєстру слів.	100	Реєстр слів

Код відповіді винятку	Індикація
16#01	Непідтримуваний код функції
16#02	Непідтримувана адреса Modbus
16#03	Довжина даних виходить за допустимий діапазон.

6.8 Комунікаційний порт EtherCAT

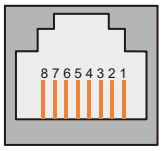
6.8.1 Функція, яку підтримує комунікаційний порт EtherCAT

Для керування рухом використовується порт EtherCAT. Інструкції керування рухом керують сервоприводами через порт EtherCAT.

6.8.2 Контакти комунікаційного порту EtherCAT

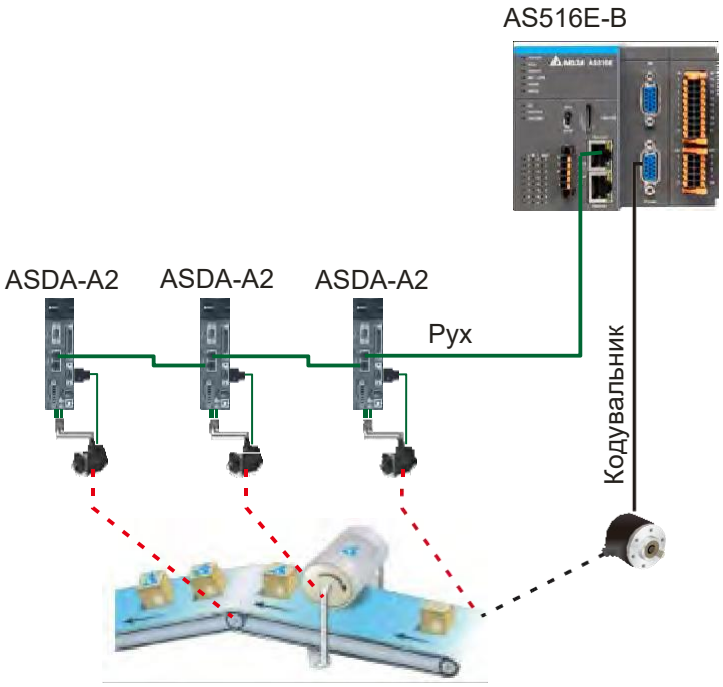
Порт EtherCAT використовується для керування рухом.
Дивіться таблицю нижче для визначення контактів комунікаційного порту EtherCAT.

PIN-№	Сигнал	Визначення
1	Tx+	Позитивний полюс для передачі даних
2	Tx-	Негативний полюс для передачі даних
3	Rx+	Позитивний полюс для отримання даних
4	Зарезервований	Зарезервований
5	Зарезервований	Зарезервований
6	Rx-	Негативний полюс для отримання даних
7	Зарезервований	Зарезервований
8	Зарезервований	Зарезервований



EtherCAT

6.8.3 Підключення до мережі комунікаційного порту EtherCAT



1. Для мережі EtherCAT існують суворі вимоги до топології мережі. Мережа повинна дотримуватися правила, згідно з яким вхідний порт наступного сервоприводу має бути підключений до вихідного порту попереднього сервоприводу.
2. Будь ласка, використовуйте кабелі Delta як кабелі EtherCAT. Технічні характеристики кабелів Delta див. у Додатку E.

6.8.4 Відстань зв'язку EtherCAT

Відстань між двома сусідніми вузлами EtherCAT не повинна перевищувати 50 м.

6.9 Комунікаційний порт CANopen

6.9.1 Функції, які підтримує комунікаційний порт CANopen

Комунікаційний порт CANopen можна використовувати як головний мережевий CANopen або як підлеглий інший головний.

- Як головний комунікаційний порт підтримує такі функції.
  - Стандартний протокол CANopen DS301V4.02;
  - Головний сервіс NMT (Network Management Object);
  - NMT Контроль помилок;
    Контроль помилок NMT використовується, щоб спостерігати за тим, чи підпорядкований пристрій перебуває в автономному режимі. Контроль помилок NMT включає Heartbeat і Node Guarding. Модуль підтримує функцію Heartbeat.
  - Підключає макс. 32 підлеглих.
  - Служба PDO (Process Data Object).

    Кількість RxPDO: макс. 200, довжина даних: макс. 1000 байт

    Кількість TxPDO: макс. 200, довжина даних: макс. 1000 байт Максимум 8 TxPDO та 8 RxPDO налаштовано для кожного підлеглого пристрою.

    Тип передачі PDO: тригер із підтримкою події, тригер за часом, синхронний і циклічний, синхронний і ациклічний
    Відображення PDO: кожен PDO може відображати не більше 32 параметрів. Тип даних, який підтримує порт зв’язку CAN

Ємність для зберігання	Тип даних
1 біт	BOOL
8 біт	SINT, USINT, BYTE
16 біт	INT, UINT, WORD
32 біт	DINT, UDINT, REAL, DWORD
64 біт	LINT, ULINT, LREAL, LWORD

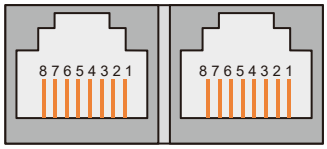
  - Підтримує службу SDO
    Підтримує стандартний прискорений режим передачі SDO;
    Підтримує функцію Auto SDO; здатність надсилати максимум 30 автоматичних SDO кожному підпорядкованому; Підтримує читання та запис підлеглих даних за допомогою служби SDO у програмі сходової діаграми ПЛК.
  - Підтримує виробника SYNC, діапазон 0-65535 мс
    Кілька пристроїв виконують дію синхронно через повідомлення SYNC.
  - Підтримує швидкість зв'язку CANopen: 20K, 50K, 125K, 250K, 500K, 1Mbps

- Як підлеглий порт зв’язку підтримує такі функції.
  - Стандартний протокол CANopen DS301V4.02
  - NMT slave service
  - Контроль помилок NMT
    Підтримка контролю помилок Heartbeat Protocol замість контролю помилок Node Guarding
  - Сервіс PDO
    Кількість RxPDO: макс. 8, довжина даних: макс. 64 байти
    Кількість TxPDO: макс. 0,8, довжина даних: макс. 64 байти
  - Тип передачі PDO: тригер події, тригер часу, синхронний і циклічний, синхронний і ациклічний
  - Сервіс SDO
    Підтримка стандартного прискореного режиму передачі SDO.

6.9.2 Виводи комунікаційного порту CANopen

Комунікаційні порти CANopen контролера руху використовуються в стандартному зв'язку CANopen, а опис контактів наведено в наступній таблиці.

PIN-№	Сигнал	Визначення
1	CAN_H	Сигнал+
2	CAN_L	Сигнал-
3	CAN_GND	0 В постійного струму
4	Зарезервований	Зарезервований
5	Зарезервований	Зарезервований
6	Зарезервований	Зарезервований
7	CAN_GND	0 В постійного струму
8	Зарезервований	Зарезервований



CANopen

6.9.3 Відображення PDO на комунікаційному порту CANopen

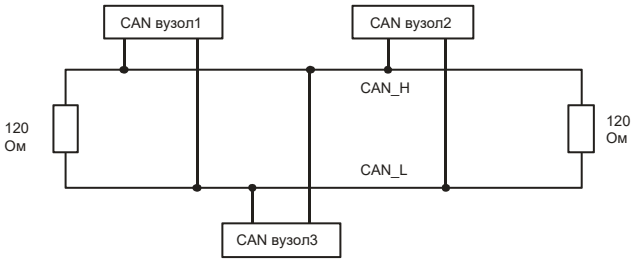
Область відображення вхідних даних становить %MW5000~%MW5499, а область відображення вихідних даних — %MW5500~%MW5999, коли контролер руху працює як головний CANopen.

Область відображення вхідних даних становить %MW5000~%MW5031, а область відображення вихідних даних — %MW5500~%MW5531, коли контролер руху працює як підлеглий пристрій CANopen.

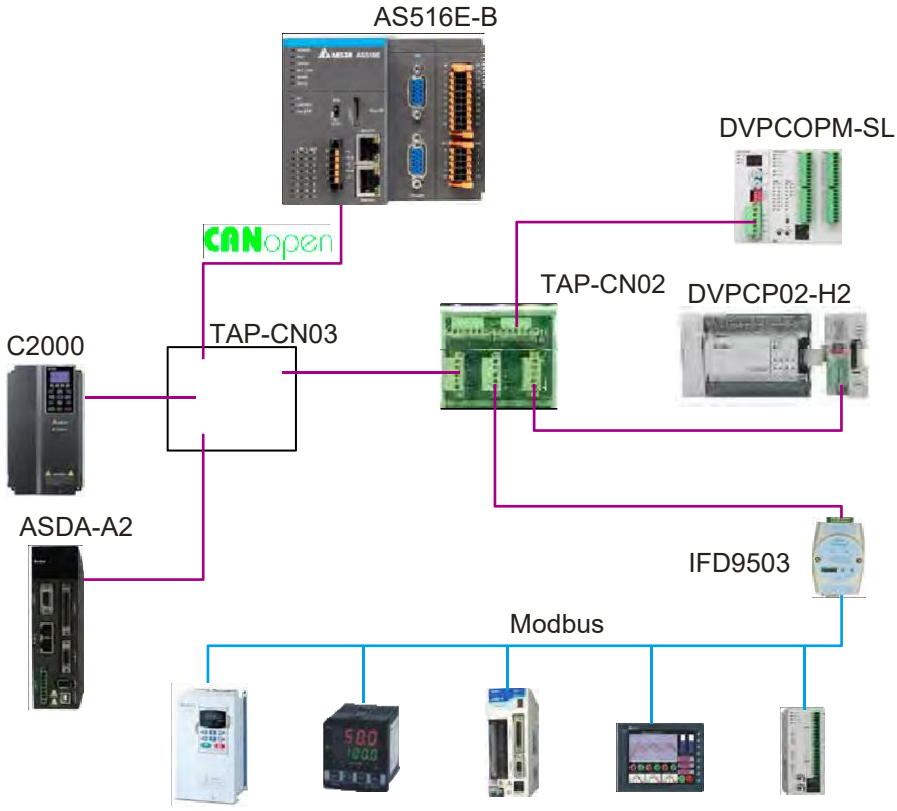
6.9.4 Підключення до мережі через комунікаційний порт CANopen

- **Шинні термінали CANopen і топологія мережі**

Обидва кінці мережі CANopen мають бути з'єднані кінцевими резисторами 120 Ом для підвищення стабільності зв'язку CANopen. Див. ілюстрацію базової топології мережі CANopen нижче.



● **Топологія шинної мережі CANopen**



1> Стандартні кабелі Delta, такі як товстий кабель UC-DN01Z-01A, тонкий кабель UC-DN01Z-02A та тонкий кабель UC-CMC010-01A, рекомендовано використовувати для побудови мережі CANopen. Кабель зв'язку повинен триматися подалі від кабелю живлення.

2> Кінцевий резистор 120 Ом слід підключити між CAN_H і CAN_L кожного кінця мережі. Ви можете придбати кінцевий резистор Delta, TAP-TR01.

6.9.5 Швидкість зв’язку CANopen і відстань зв’язку

Відстань передачі шини CANopen залежить від швидкості передачі шини CANopen. Нижче наведено таблицю, де максимальні відстані зв’язку відповідають різним швидкостям передачі.

Швидкість передачі (біт/с)	20 тис	50 тис	125 тис	250 тис	500 тис	1M
Макс. відстань зв'язку (метри)	2500	1000	500	250	100	25

Пам'ятка

7

Розділ 7 Принцип виконання

Table of Contents

7.1 Tasks 7-2

7.1.1 Типи завдань 7-2

7.1.2 Рівні пріоритету завдань 7-4

7.1.3 Сторожовий пес для завдання 7-6

7.1.4 Інструкції щодо руху та спілкування для кожного типу завдань 7-7

7.2 The Impact of PLC RUN or STOP on Variables and Devices 7-10

7.3 Relationship between Motion Program and Motion Bus 7-10

7.4 Synchronization Cycle Period Setting 7-11

7.1 завдання

- Завдання — це ряд функцій обробки заданих умов виконання та послідовності виконання для оновлення вводу/виводу та виконання програми користувача.
- Завдання визначається назвою, рівнем пріоритету та типом. Завдання можна класифікувати на три типи: циклічні завдання, вільні завдання та завдання, викликані подією.
- Для кожного завдання можна вказати групу POU, які запускаються завданням. Якщо завдання виконується в поточний період, POU буде оброблено протягом певного періоду часу.
- Рівень пріоритету та тип завдання визначають послідовність виконання завдання.
- Для кожного завдання можна призначити сторожового песа.

7.1.1 Типи завдань

- **Три типи завдань, які підтримує контролер руху**

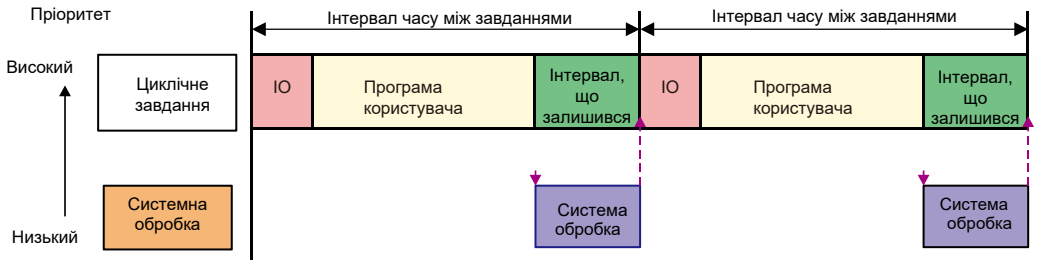
  1. Циклічний
  2. Вільний хід
  3. Ініціюється подією

- **Нижче описано максимум 24 завдання, які підтримує контролер руху.**

  - **Циклічне завдання**

    Цикличне завдання буде виконуватися циклічно відповідно до встановленого інтервалу часу.

    ➢ **Спосіб виконання циклічного завдання**



***IO* : IO** означає оновлення вводу/виводу. Вхід/вивід включає локальні точки вводу/виводу та дані правого модуля розширення та дані CANopen. Дані можна вказати для оновлення перед виконанням поставленого завдання. Якщо не вказано, дані будуть оновлені під час обробки системи.

Програма користувача: Програма користувача означає виконання програми користувача, яке базується на послідовності виконання програм, призначених у завданні.

Інтервал, що залишився:

Коли контролер повинен виконати системну обробку, спочатку виконується завдання з низьким пріоритетом, якщо таке є, а потім виконується системна обробка.

Системна обробка:

Контролер виконуватиме системну обробку, яка включає обробку зв'язку Ethernet, RS232 і RS485 після виконання всіх запитів завдань.

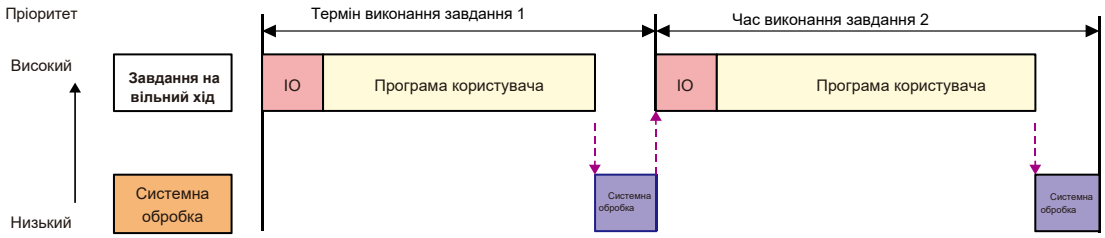
Чотири терміни, згадані вище, мають ті самі значення, що й у наступних розділах.

Примітка. Якщо цикл, встановлений для циклічного завдання, надто короткий, після завершення виконання програми користувача виконання завдання буде негайно повторено, а завдання з низьким пріоритетом або системна обробка не виконуватимуться. У цьому випадку це вплине на виконання всіх завдань. Якщо для завдання встановлено сторожовий тайм-аут, відбудеться тайм-аут сторожового таймера, контролер перейде в статус помилки, і виконання програми користувача буде припинено. Якщо для завдання не встановлено сторожовий таймер, контролер не зможе виконувати системну обробку, і виникнуть такі проблеми, як тайм-аут зв'язку.

■ **Завдання на вільний хід**

Завдання вільного ходу **:** завдання буде оброблено, щойно запущена програма. Завдання буде автоматично перезапущено в наступному циклі після завершення одного циклу виконання.

➢ **Спосіб виконання завдання вільного ходу**

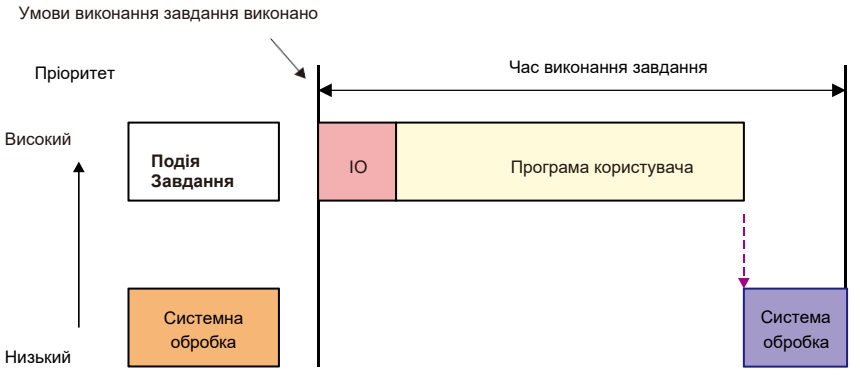


Примітка. Немає фіксованого часу виконання для завдання вільного ходу. Тому значення часу виконання завдання 1 і часу виконання завдання 2 можуть не збігатися на наведеному вище малюнку.

■ **Завдання, викликане подією**

Подійне завдання **:** подійне завдання виконується один раз, коли відбувається зазначена подія. Час для виконання подійного завдання залежить від часу виникнення події та рівня пріоритету подійного завдання.

➢ **Спосіб виконання завдання події**



➢ **Подійні завдання для опції містять наступні кілька типів.**

- Подія руху (завдання керування рухом)
- Наростаючий або спадаючий фронт локальних вхідних точок (I0~I7 та I10~I17)
- Сигнал CANopen SYNC
- Наростаючий фронт імпульсу Z інкрементального кодера 1 або кодера 2

Умова для повторного виконання ігнорується, коли умова, необхідна для виконання завдання події, виконується знову до завершення завдання події. Період до завершення завдання події — це час, поки завдання події виконується або очікує на виконання.

● **Подія руху**

Порт руху контролера надсилає сигнал SYNC, і завдання запускається.

Примітка. За замовчуванням для завдання руху встановлено пріоритет 1. Рівень пріоритету можна змінити. Однак переконайтеся, що є достатньо часу для виконання завдання руху протягом періоду CANopen SYNC.

■ **Налаштування циклу SYNC має відповідати наступним умовам.**

➢ Має бути достатньо часу для виконання програми, визначеної в руховому завданні.

➢ Має бути достатньо часу для обміну даними PDO і SDO між контролером і сервоприводом.

Недостатній час періоду SYNC призведе до того, що керований пристрій не зможе отримати сигнал SYNC і непередбачуваних операцій. Зверніться до розділу 7.3 щодо встановлення періоду SYNC.

- **Наростаючий або спадаючий фронт локальних вхідних точок (I0~I7 , I10~I17)**

  Завдання запускається, коли виявляється наростаючий або спадаючий фронт сигналу вхідної точки. Час відгуку вхідних точок можна встановити за допомогою функції фільтра.

- **Повідомлення CANopen шини SYNC**

  Завдання запускається, коли сигнал SYNC створюється на порту CANopen контролера.

- **Наростаючий фронт імпульсу Z для інкрементального кодера 1**

  Завдання запускається, коли передній фронт сигналу Z першого кодера виявляється на порту кодера контролера.

- **Наростаючий фронт імпульсу Z для інкрементального кодера 2**

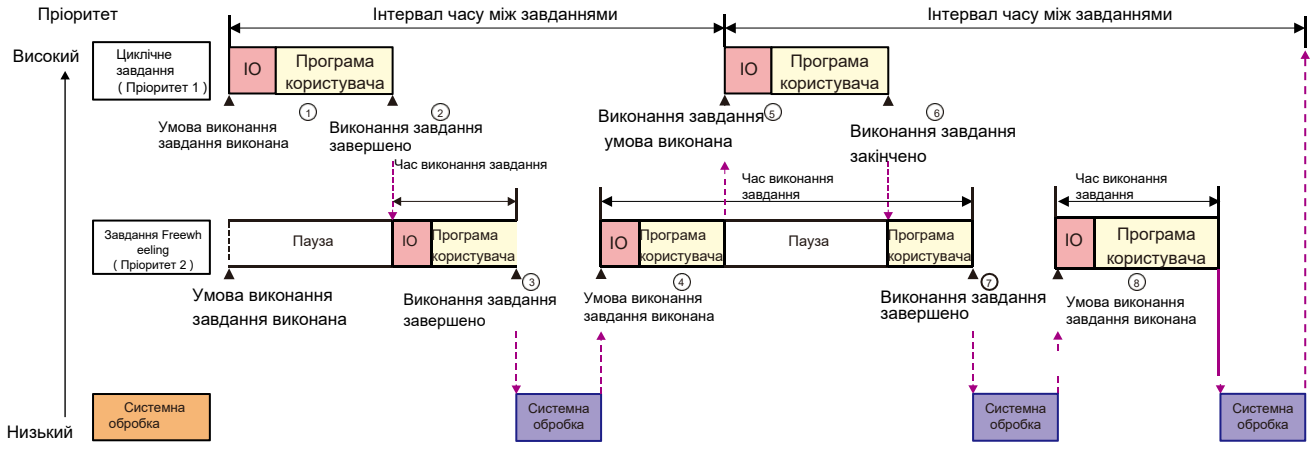
  Завдання запускається, коли передній фронт сигналу Z другого кодера виявляється на порту кодера контролера.

7.1.2 Рівні пріоритету завдань

Контролер не може виконувати кілька завдань одночасно. Кожному завданню має бути наданий рівень пріоритету, і вони виконуються відповідно до попередньо встановлених пріоритетів. Рівень пріоритету можна встановити в діапазоні від 1 до 24. (1 — найвищий пріоритет, а 24 — найнижчий). Рівень пріоритету кожного завдання має бути унікальним. Завдання з вищим пріоритетом має пріоритет для виконання. Завдання з високим пріоритетом може перервати завдання з низьким пріоритетом.

Ми рекомендуємо, щоб завдання, яке має високі вимоги до реального часу, мали мати високий пріоритет, а завдання, яке має низькі вимоги до реального часу, мали мати низький пріоритет. Пріоритет завдання керування рухом за замовчуванням, створеного в програмному забезпеченні CANopen Builder, за замовчуванням становить 1.

- **Принцип багатозадачності**
- **При одночасному виконанні умов виконання двох завдань (Циклічне завдання та Вивільне завдання)**



◯ 1 Умови виконання для циклічного завдання та завдання вільного ходу виконуються одночасно. Циклічне завдання виконується першим через його вищий пріоритет.

◯ 2 Після завершення циклічного виконання завдання починається вільне виконання завдання.

○ 3 Контролер виконає обробку системи, якщо після виконання завдання вільного ходу не буде іншого завдання.

○ 4 Виконання вільного завдання продовжується, оскільки запит циклічного завдання з високим пріоритетом не надійшов.

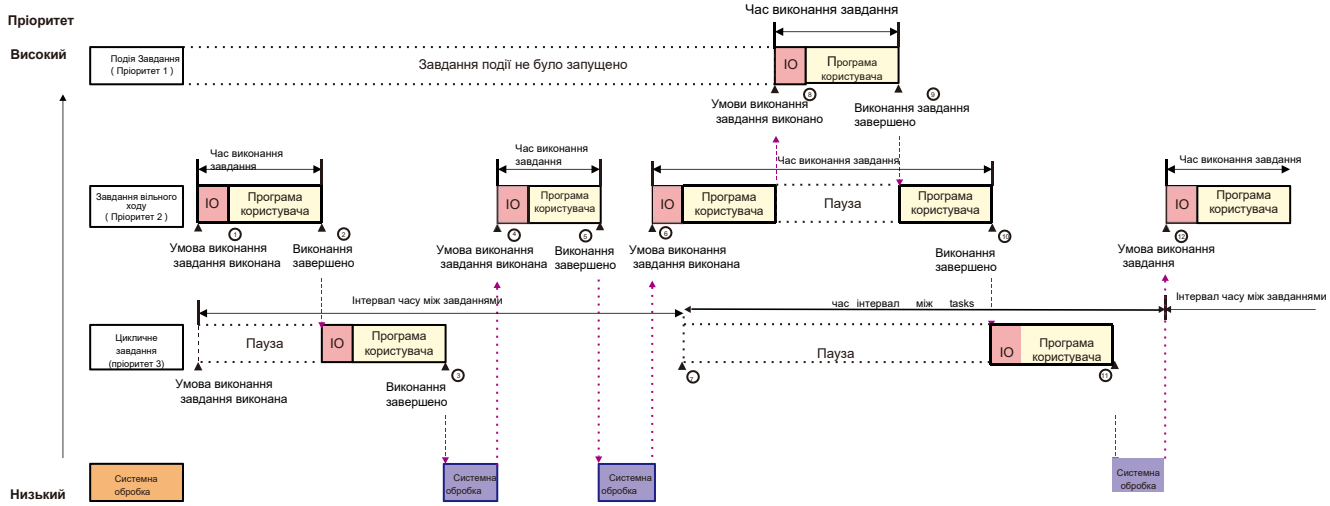
○ 5 Циклічне завдання переривається виконання вільного завдання, а контролер виконує циклічне через надходження високопріоритетного запиту циклічного завдання під час виконання вільного завдання.

○ 6 Контролер продовжує виконувати частину низькопріоритетного завдання вільного ходу, яке ще не було виконано, коли виконання циклічного завдання завершено.

○ 7 Коли виконання завдання вільного ходу завершено, контролер виконує системну обробку через відсутність іншого запиту завдання.

○ 8 Коли системна обробка завершена, виконання вільного завдання продовжується через відсутність високопріоритетного запиту циклічного завдання.

■ **Коли три завдання виконуються разом (завдання подій, циклічне завдання та завдання вільного ходу)**



○ 1 Коли виконуються умови для виконання завдання вільного ходу та циклічного завдання, завдання вільного ходу виконується першим, оскільки пріоритет завдання вільного ходу вищий.

○ 2 Виконання циклічного завдання починається після завершення вільного виконання завдання.

○ 3 Коли циклічне виконання завдання завершено, контролер виконує системну обробку через відсутність іншого запиту завдання.

○ 4 Завдання вільного ходу виконується після завершення системної обробки.

○ 5 Коли виконання вільного завдання завершено, контролер виконує системну обробку через відсутність іншого запиту завдання.

○ 6 Завдання вільного ходу виконується після завершення системної обробки.

○ 7 Виконання завдання вільного ходу продовжується, оскільки завдання вільного ходу має вищий пріоритет ніж циклічне завдання, хоча умова виконання для циклічного завдання виконується. І циклічне завдання чекає на виконання.

○ 8 Завдання події переривається виконання завдання вільного ходу, оскільки завдання події має найвищий пріоритет і умова виконання для завдання події виконується.

◯ 9 Контролер продовжує виконувати частину завдання вільного ходу з низьким пріоритетом, яка ще не була виконана, коли виконання завдання події завершено.

◯ 10 Виконання завдання на вільний хід виконано. Контролер виконує циклічне завдання, оскільки на запит циклічного завдання в ⑦ ще немає відповіді.

◯ 11 Виконання циклічного завдання завершено. Контролер виконує системну обробку через відсутність іншого запиту завдання.

7.1.3 Сторожовий пес для завдання

За кожним завданням можна поставити сторожового пса. Коли час виконання завдання перевищує встановлений час сторожового таймера, контролер перейде в стан помилки, і виконання програми користувача припиниться.

Watchdog time **:** найдовший час, дозволений для виконання завдання

7.1.4Інструкції щодо руху та спілкування для кожного типу завдань

Ось таблиця інструкцій щодо руху для різних типів завдань. «V» означає, що інструкція руху може бути виконана для типу завдання, а «–» означає, що інструкція руху не може бути виконана для типу завдання.

Класифікація	Назва інструкції	Тип завдання			
				Подійне завдання	
		Циклічне завдання	Завдання на вільний хід	Рухове завдання	Рухове завдання
Одноосьові інструкції	MC_Power	-	-	V	-
	MC_Home	-	-	V	-
	MC_MoveVelocity	-	-	V	-
	MC_Halt	-	-	V	-
	MC_Stop	-	-	V	-
	MC_MoveRelative	-	-	V	-
	MC_MoveAdditive	-	-	V	-
	MC_MoveAbsolute	-	-	V	-
	MC_MoveSuperimposed	-	-	V	-
	MC_Halts накладено	-	-	V	-
	MC_SetPosition	-	-	V	-
	MC_SetOverride	-	-	V	-
	MC_Reset	-	-	V	-
	DMC_SetTorque	-	-	V	-
	MC_ReadAxisError	V	V	V	V
	MC_ReadActualPosition	V	V	V	V
	MC_ReadStatus	V	V	V	V
	MC_ReadMotionState	V	V	V	V
	DMC_ReadParameter_Motion	V	V	V	V
	DMC_WriteParameter_Motion	V	V	V	V
	DMC_TouchProbe	-	-	V	-
	DMC_ChangeMechanismGearRatio	-	-	V	-
	DMC_Jog	-	-	V	-
	DMC_MoveVelocity	-	-	V	-
	DMC_MoveVelocityStopByPos	-	-	V	-
	DMC_MoveVelocityStopByLinePos	-	-	V	-
	DMC_ReadPositionLagStatus	V	V	V	V
	DMC_SwitchSoftLimit	V	V	V	V
	DMC_TorqueControl	-	-	V	-
	DMC_TouchProbeCyclically	-	-	V	-
	DMC_WritePositionLagSetting	V	V	V	V

Класифікація	Назва інструкції	Тип завдання			
		Циклічне завдання	Завдання на вільний хід	Подійне завдання	
				Рухове завдання	Рухове завдання
Інструкція по зчепленню	MC_GearIn	-	-	V	-
	MC_GearOut	-	-	V	-
	MC_CombineAxes	-	-	V	-
	MC_CamIn	-	-	V	-
	MC_CamOut	-	-	V	-
	DMC_CamAddTappet	V	V	V	V
	DMC_CamDeleteTappet	V	V	V	V
	DMC_CamReadPoint	V	V	V	V
	DMC_CamWritePoint	V	V	V	V
	DMC_CamSet	-	-	V	-
	DMC_CamReadTappetStatus	V	V	V	V
	DMC_CamReadTappetValue	V	V	V	V
	DMC_CamWriteTappetValue	V	V	V	V
Інструкції із застосування	APF_RotaryCut_Init	-	-	V	-
	APF_RotaryCut_In	-	-	V	-
	APF_RotaryCut_Out	-	-	V	-
Інструкції з кодом G	DMC_CartesianCoordinate	-	-	V	-
	DMC_ReadMFunction	V	V	V	V
	DMC_ResetMFunction	V	V	V	V
	DMC_SetG0Para	V	V	V	V
	DMC_SetG1Para	V	V	V	V
	DMC_SetStartPosition	V	V	V	V
Інструкції групи сокир	DMC_ AddAxisToGroup	-	-	V	-
	DMC_RemoveAxisFromGroup	-	-	V	-
	DMC_UngroupAllAxes	-	-	V	-
	DMC_ GroupEnable	-	-	V	-
	DMC_GroupStop	-	-	V	-
	DMC_GroupInterrupt	-	-	V	-
	DMC_GroupContinue	-	-	V	-
	DMC_MoveDirectAbsolute	-	-	V	-
	DMC_MoveDirectRelative	-	-	V	-
	DMC_MoveLinearAbsolute	-	-	V	-
	DMC_MoveLinearRelative	-	-	V	-
	DMC_MoveCircularAbsolute	-	-	V	-
	DMC_MoveCircularRelative	-	-	V	-

Класифікація	Назва інструкції	Тип завдання			
		Циклічне завдання	Завдання на вільний хід	Подійне завдання	
				Рухове завдання	Рухове завдання
	DMC_GroupSetOverride	-	-	V	-
	DMC_GroupReadActualPosition	V	V	V	V
Інструкція з координації	DMC_ControlAxisByPos	-	-	V	-
	DMC_NC	-	-	V	-
Інструкції щодо спілкування	DMC_ReadParameter_CANopen	V	V	V	V
	DMC_WriteParameter_CANopen	V	V	V	V
	ETH_Link_Config	V	V	V	V
	ETH_Link_Manage	V	V	V	V
	ETH_Link_Status	V	V	V	V
	ETH_Link_Config_Ext	V	V	V	V
	ETH_ SetServerlinkkeeptime	V	V	V	V
	ETH_Socket_Manage	V	V	V	V
	ETH_Socket_Config	V	V	V	V
	ETH_Socket_Open	V	V	V	V
	ETH_Socket_Send	V	V	V	V
	ETH_Socket_Receive	V	V	V	V
	ETH_Socket_Close	V	V	V	V
	ETH_Socket_Status	V	V	V	V
	RS485_Link_Manage	V	V	V	V
	RS485_Link_Config	V	V	V	V
	RS485_Link_Status	V	V	V	V
	RS485_RS	V	V	V	V
	RS485_SetDelayTime	V	V	V	V
	RS232_Link_Manage	V	V	V	V
	RS232_Link_Config	V	V	V	V
	RS232_Link_Status	V	V	V	V
	RS232_RS	V	V	V	V
	RS232_SetDelayTime	V	V	V	V

7.2 Вплив ПЛК RUN або STOP на змінні та пристрої

Коли контролер руху перемикається з RUN на STOP, змінні та пристрої зберігають поточні значення. Коли контролер руху перемикається з STOP на RUN, користувачі можуть вибрати одну опцію, згідно з якою значення змінних і незаблокованих пристроїв очищаються або зберігаються, як показано нижче.

- **Значення змінних і незаблокованих пристроїв очищаються.**

  Коли контролер руху перемикається з STOP на RUN, значення змінних і незаблокованих пристроїв очищаються та відновлюються до початкових значень. Якщо змінні та незаблоковані пристрої не мають початкових значень, значення змінних і незафіксованих областей буде відновлено до значення за замовчуванням 0.

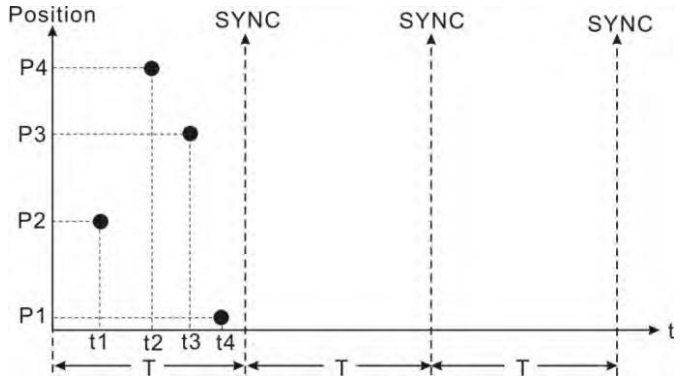
- **Значення змінних і пристроїв зберігаються.**

  Коли контролер руху перемикається з STOP на RUN, змінні та пристрої зберігають поточні значення.

7.3 Зв'язок між програмою руху та шиною руху

Контролер руху забезпечує синхронізацію, що досягається шляхом видачі сигналу SYNC у спосіб трансляції, коли до контролера руху підключено більше одного сервоприводу. Сервоприводи отримують керуючі дані, надіслані контролером руху. Але отримані керуючі дані не будуть ефективними відразу, поки сигнал SYNC не надійде на сервоприводи, щоб реалізувати синхронізацію кількох сервоприводів.

На наступному малюнку контролер руху з’єднаний з 4 сервоприводами, а T – період синхронізації. Чотири сервоприводи отримують керуючі дані в різний час (t1, t2, t3 і t4), але отримані керуючі дані не діють одразу. Коли сервоприводи отримують сигнал SYNC, дані керування починають діяти негайно.



7.4 Налаштування періоду циклу синхронізації

Цикл синхронізації є дуже важливим параметром для керування рухом шини. Якщо період синхронізації не встановлено належним чином, сервопривод може відображати повідомлення про помилку AL303/AL302/AL301 під час зв’язку або сервопривід не може працювати нормально.

Давайте спочатку введемо склад періоду синхронізації.

Програма керування рухом сканується на самому початку періоду синхронізації, а потім контрольні повідомлення, отримані через розрахунок, надсилаються на всі осі. Тому ми можемо розглядати період синхронізації як час для виконання програми керування рухом плюс час для зв’язку між контролером руху та всіма сервоприводами.

Час виконання програми керування рухом – це максимальний час виконання завдань події руху з одиницею вимірювання: мкс (мікросекунда), який можна переглянути подвійним клацанням на **Завдання** в інтерфейсі програмного забезпечення CANopen Builder. 1000 мкс (мікросекунди) це 1 мс (мілісекунда).

У фактичній програмі значення округлюється до цілого числа. Наприклад, максимальний час виконання програми виконання становить 2567 мкс=2,5 мс, у цьому випадку ми можемо розглядати 3 мс як час для виконання програми.

Це приблизно 0,5 мс для зв’язку між контролером руху та сервоприводом.

У програмі ми рекомендуємо округляти значення до цілого числа. Наприклад, у додатку налаштовано 5 сервоприводів. А час зв'язку становить 5*0,5 мс=2,5 мс. У цьому випадку ми можемо розглядати 3 мс як час для спілкування.

Таким чином, ми можемо отримати формулу: час синхронізації (мс) = ціле число, отримане шляхом округлення значення максимального часу виконання програми (мс) + час для зв’язку між контролером руху та всіма сервоприводами (мс) +1 (час зарезервовано для зміни програми) (мс).

Якщо після зміни програми тривалість роботи програми надто збільшиться, заданий час синхронізації більше не підійде. Таким чином, зарезервований час має бути встановлений на 1~2 мс.

Наприклад, максимальний час виконання програми становить 1634 мкс, а в додатку всього 5 сервоприводів. Зарезервований час для зміни програми становить 1 мс.

Період циклу синхронізації = 2 мс (отримано шляхом округлення максимального часу виконання програми, 1634 мкс) + 3 мс (отримано шляхом округлення 5*0,5) +1 мс (зарезервовано для зміни програми) = 6 мс

Примітка:

Наведений вище метод використовується для отримання розрахункового часу, який підходить для більшості програм. Якщо вам потрібен більш точний період циклу синхронізації, фактичний час можна перерахувати, пропустивши зарезервований час після завершення розробки програми.

Пам'ятка

8

Розділ 8 Логічні інструкції

Зміст

8.1 Table of Logic Instructions 8-6

8.2 Explanation of Logic Instructions 8-11

8.2.1 EN та ENO 8-11

8.3 Sequence Input and Output Instructions..... 8-11

8.3.1 R_TRIG 8-11

8.3.2 F_TRIG 8-13

8.3.3 RS 8-15

8.3.4 SR 8-17

8.3.5 SEMA 8-19

8.4 Sequence Control Instructions 8-21

8.4.1 JMP..... 8-21

8.5 Data Movement Instructions 8-22

8.5.1 MOVE 8-22

8.5.2 MoveBit 8-24

8.5.3 TransBit 8-26

8.5.4 MoveDigit 8-28

8.5.5 Exchange..... 8-30

8.5.6 Swap..... 8-32

8.6 Comparison Instructions 8-34

8.6.1 LT..... 8-34

8.6.2 LE..... 8-36

8.6.3 GT 8-38

8.6.4 GE 8-40

8.6.5 EQ..... 8-42

8.6.6 NE 8-44

8.7 Timer Instructions..... 8-46

8.7.1 TON..... 8-46

8.7.2 TOF 8-48

8.7.3 TP 8-50

8.7.4 Sys_ReadTime 8-52

8.7.5 Sys_ReadTotalWorkTime 8-53

8.7.6 Sys_ReadPowerOnTime 8-54

8.7.7 Sys_WdgStatus 8-55
8.8 Counter Instructions 8-57
8.8.1 CTU 8-57
8.8.2 CTD 8-59
8.8.3 CTUD 8-61
8.9 Math Instructions 8-64
8.9.1 ADD 8-64
8.9.2 SUB 8-67
8.9.3 MUL 8-70
8.9.4 DIV 8-73
8.9.5 MOD 8-76
8.9.6 MODREAL 8-78
8.9.7 MODTURNS 8-80
8.9.8 MODABS 8-82
8.9.9 ABS 8-84
8.9.10 DegToRad 8-86
8.9.11 RadToYou 8-88
8.9.12 SIN 8-90
8.9.13 COS 8-92
8.9.14 TAN 8-94
8.9.15 ASIN 8-96
8.9.16 ACOS ..... 8 -99
8.9.17 3A 8-101
8.9.18 LN 8-103
8.9.19 LOG 8-105
8.9.20 SQRT ..... 8 -107
8.9.21 EXP 8-109
8.9.22 EXPT 8-111
8.9.23 RAND 8-113
8.9.24 TRUNC 8-115
8.9.25 FLOOR 8-117
8.9.26 FRACTION 8-119
8.10 Bit String Instructions 8-121
8.10.1 AND 8-121
8.10.2 OR 8-124
8.10.3 NOT 8-127
8.10.4 XOR 8-129
8.10.5 XORN 8-132
8.11 Shift Instructions 8-135
8.11.1 SHL 8-135
8.11.2 SHR 8-137
8.11.3 ROL 8-139

8.11.4 ROR 8-141
8.12 Selection Instructions 8-143
8.12.1 MAX 8-143
8.12.2 MIN 8-145
8.12.3 SEL 8-147
8.12.4 MUX 8-149
8.12.5 LIMIT 8-152
8.12.6 BAND 8-154
8.12.7 ZONE 8-157
8.13 Data Type Conversion Instructions 8-160
8.13.1 BOOL_TO_*** 8-160
8.13.2 Бітові рядки_TO_*** 8-163
8.13.3 Цілі числа_TO_*** 8-170
8.13.4 Дійсні числа_TO_*** 8-179
8.13.5 Час, дата_TO_*** 8-182
8.13.6 Рядки_TO_*** 8-184
8.14 Communication Instructions 8-187
8.14.1 Інструкції зі зв'язку CANopen 8-187
8.14.1.1 DMC_ReadParameter_CANopen 8-187
8.14.1.2 DMC_WriteParameter_CANopen 8-19
8.14.2 Інструкції з Ethernet 8-197
8.14.2.1 ETH_Link_Config 8-197
8.14.2.2 ETH_Link_Manage 8-202
8.14.2.3 ETH_Link_Status 8-204
8.14.2.4 Приклад обміну даними MODBUS TCP 8-207
8.14.2.5 ETH_Link_Config_Ext 8-213
8.14.2.6 ETH_SetServerlinkkeeptime 8-215
8.14.2.7 ETH_Socket_Manage 8-216
8.14.2.8 ETH_Socket_Config 8-218
8.14.2.9 ETH_Socket_Open 8-221
8.14.2.10 ETH_Socket_Send 8-223
8.14.2.11 ETH_Socket_Receive 8-227
8.14.2.12 ETH_Socket_Close 8-231
8.14.2.13 ETH_Socket_Status 8-233
8.14.2.14 Приклад вільного протоколу Ethernet 8-236
8.14.3 Інструкції щодо зв'язку RS485 8-241
8.14.3.1 RS485_Link_Manage 8-241
8.14.3.2 RS485_Link_Config 8-243
8.14.3.3 RS485_Link_Status 8-247
8.14.3.4 Приклад обміну даними RS485 8-250
8.14.3.5 RS485_RS 8-253
8.14.3.6 Приклад безкоштовного протоколу RS485 8-259

8.14.3.7 RS485_SetDelayTime 8-262
8.14.4 Інструкції щодо зв'язку RS232 8-264
8.14.4.1 RS232_Link_Manage 8-264
8.14.4.2 RS232_Link_Config 8-266
8.14.4.3 RS232_Link_Status 8-271
8.14.4.4 Приклад обміну даними RS232 8-274
8.14.4.5 RS232_RS 8-277
8.14.4.6 Приклад безкоштовного протоколу RS232 8-283
8.14.4.7 RS232_SetDelayTime 8-286
8.15 String Processing Instructions 8-288
8.15.1 CONCAT 8-288
8.15.2 DELETE 8-290
8.15.3 INSERT 8-292
8.15.4 LEFT / RIGHT 8-294
8.15.5 MID 8-296
8.15.6 REPLACE 8-298
8.15.7 LEN 8-300
8.15.8 FIND 8-301
8.16 Immediate Refresh Instructions 8-303
8.16.1 FROM 8-303
8.16.2 TO 8-307
8.16.3 ImmediateInput 8-311
8.16.4 Миттєвий вихід 8-313
8.17 PID-related Instructions 8-315
8.17.1 PID 8-315
8.17.2 GPWM 8-326
8.18 Address Instruction 8-328
8.18.1 ADR 8-328
8.19 Network Diagnosis 8-330
8.19.1 Діагностика EtherCAT 8-330
8.19.1.1 EtherCAT_SysDiag 8-330
8.19.2 Діагностика CANopen 8-332
8.19.2.1 CANopen_SysDiag 8-332
8.19.2.2 CANopen_NodeDiag 8-334
8.19.2.3 CANopen_State 8-336
8.20 Read and Write Offset Bit Value 8-338
8.20.1 SetBitOffsetValue 8-338
8.20.2 GetBitOffsetValue 8-340
8.21 FCS Instructions 8-342
8.21.1 CRC16 8-342
8.21.2 LRC 8-344

8.22 Right-Side Extension Module Instructions..... 8-346
8.22.1 Інструкції, пов’язані з AS02PU та AS04PU..... 8-346
8.22.1.1 Умови застосування 8-346
8.22.1.2 Інструкції, пов’язані з AS02PU та AS04PU 8-347

8.1 Таблиця логічних інструкцій

Набір інструкцій	Код інструкції	функція
Інструкції послідовного введення / Вихід	R_TRIG	Тригер наростаючого фронту
	F_TRIG	Тригер із спадаючим краєм
	RS	Скидання – пріоритетна інструкція
	SR	SET–Інструкція пріоритету
	SEMA	Інструкція щодо пріоритету вимоги
Інструкції з контролю послідовності	JMP	Стрибок
Інструкції з переміщення даних	MOVE	рухатися
	MoveBit	Перемістіть один біт
	TransBit	Перемістити біти
	MoveDigit	Переміщення цифр
	Exchange	Обмін даними
	Swap	Поміняти байти
Інструкція порівняння	LT	Менше ніж
	LE	Менше або дорівнює
	GT	Більш чим
	GE	Більше або дорівнює
	EQ	Дорівнює
	NE	Не дорівнює
Інструкції таймера	TON	Таймер затримки ввімкнення
	TOF	Таймер затримки вимкнення
	TP	Таймер імпульсного типу
	Sys_ReadTime	Читайте час годинника реального часу
	Sys_ReadTotalWorkTime	Читайте Загальний робочий час
	Sys_ReadPowerOnTime	Прочитайте час увімкнення
	Sys_WdgStatus	Прочитати стан очікування завдання
Зустрічні інструкції	CTU	Up-Counter
	CTD	Вниз-Counter
	CTUD	Лічильник вгору-вниз
Інструкції з математики	ADD	Доповнення
	SUB	Віднімання
	MUL	Множення

Набір інструкцій	Код інструкції	функція
	DIV	Поділ
	MOD	Ціле число за модулем для отримання залишку
	MODREAL	Ділення за модулем дійсного числа для отримання залишку
	MODTURNS	Ділення за модулем дійсного числа для отримання невід'ємної частини зі знаком
	MODABS	Ділення за модулем дійсного числа, щоб отримати беззнакове значення за модулем
	ABS	Абсолютне значення
	DegToRad	Градуси в Радіани
	RadToDeg	Радіани в градуси
	SIN	синус
	COS	Косинус
	TAN	Дотична
	ASIN	Аркс синус
	ACOS	Арккосинус
	ATAN	Арктангенс
	LN	Натуральний логарифм
	LOG	Логарифм з основою 10
	SQRT	Квадратний корінь
	EXP	Натуральна експоненціальна операція
	EXPT	Піднесення до степеня
	RAND	Випадкове число
	TRUNC	Обрізати
	FLOOR	Поверх дійсного числа
	FRACTION	Реальний дріб
Бітові рядкові інструкції	AND	Логічне AND
	OR	Логічне OR
	NOT	Зворотний біт
	XOR	Логічне виключаюче OR
	XORN	Логічне виняткове HI
Інструкції по зміні	SHL	Зсув бітів вліво
	SHR	Зсув бітів праворуч
	ROL	Повернути біти вліво

Набір інструкцій	Код інструкції	функція
	ROR	Повернути біти вправо
Інструкції з вибору	MAX	Максимум
	MIN	Мінімум
	SEL	Вибір
	MUX	Мультиплексор
	LIMIT	Обмежувач
	BAND	Контроль зони нечутливості
	ZONE	мертвої зони
Інструкції з перетворення типу даних	BOOL_TO_***	Група перетворення Bool
	Бітові рядки_TO_***	Група перетворення бітового рядка
	Цілі числа_TO_***	Група перетворення цілих чисел
	Реальні числа_TO_***	Група перетворення дійсних чисел
	Час, дати_TO_***	Група перетворення часу та даних
	Текстові рядки_TO_***	Група перетворення рядків
Інструкції щодо спілкування	DMC_ReadParameter_CANopen	Прочитайте значення параметра
	DMC_WriteParameter_CANopen	Напишіть значення параметра
	ETH_Link_Config	Налаштувати обмін даними MODBUS TCP
	ETH_Link_Manage	Увімкнути/вимкнути обмін даними MODBUS TCP
	ETH_Link_Status	Переглядайте стан обміну даними MODBUS TCP
	ETH_Link_Config_Ext	Налаштуйте параметри розширення для обміну MODBUS TCP
	ETH_SetServerlinkkeeptime	Встановіть тривалість підключення, оскільки контролер працює як ведений
	ETH_Socket_Manage	Керування сокетом TCP/UDP
	ETH_Socket_Config	Налаштувати параметри обміну даними Socket
	ETH_Socket_Open	Увімкнути Socket
	ETH_Socket_Send	Надіслати дані Socket
	ETH_Socket_Receive	Отримати дані Socket
	ETH_Socket_Close	Вимкнути Socket
	ETH_Socket_Status	Читати статус сокета
	RS485_Link_Manage	Керуйте зв'язком RS485
	RS485_Link_Config	Налаштуйте параметри зв’язку RS485

Набір інструкцій	Код інструкції	функція
	RS485_Link_Status	Спостерігайте за станом зв'язку RS485
	RS485_RS	Налаштуйте параметри безкоштовного протоколу RS485
	RS485_SetDelayTime	Встановіть час затримки відповіді зв’язку RS485
	RS232_Link_Manage	Керуйте зв'язком RS232
	RS232_Link_Config	Налаштуйте параметри зв'язку RS232
	RS232_Link_Status	Спостерігайте за станом зв'язку RS232
	RS232_RS	Налаштувати параметри вільного протоколу RS232
	RS232_SetDelayTime	Встановіть час затримки відповіді RS232
Інструкції з обробки рядка	CONCAT	Конкатенація рядків
	DELETE	Видалити рядок
	INSERT	Вставте рядок
	LEFT / RIGHT	Отримати рядок вліво/вправо
	MID	Отримати рядок
	REPLACE	Замінити рядок
	LEN	Довжина рядка
	FIND	Знайти рядок
Інструкції щодо негайного оновлення	FROM	Прочитайте значення CR
	TO	Записати значення в CR
	ImmediateInput	Негайне оновлення точок введення
	ImmediateOutput	Негайне оновлення вихідних точок
Інструкції щодо PID	PID	робота ПІД
	GPWM	Базове налаштування ширини імпульсу
Адресна інструкція	ADR	Отримати адресу
Діагностика мережі	EtherCAT_SysDiag	Діагностика системи EtherCAT
	CANopen_SysDiag	Діагностика системи CANopen
	CANopen_NodeDiag	CANopen Slave Diagnosis
	CANopen_State	CANopen Master Diagnosis
Значення біта зсуву читання та запису	SetBitOffsetValue	Встановити значення зазначеного біта
	GetBitOffsetValue	Прочитайте значення зазначеного біта
Інструкція FCS	CRC16	Розрахувати значення CRC
	LRC	Розрахувати значення LRC

Набір інструкцій	Код інструкції	функція
Інструкції для правого модуля розширення	PUCONF	Налаштування вихідних параметрів керування модулем ПУ
	PUSTAT	Зчитування вихідного стану модуля PU
	PUPLS	Імпульсний вихід модуля ПУ без прискорення і уповільнення
	PUDRI	Вихід відносної позиції модуля PU
	PUDRA	Вихід абсолютної позиції модуля PU
	PUZRN	Наведення модуля ПУ
	PUCNT	Функція високошвидкісного лічильника модуля PU

8.2 Пояснення логічних інструкцій

8.2.1 EN та ENO

Якщо використовувана інструкція має вхідні параметри EN та ENO, а значення EN дорівнює FALSE (0), функція інструкції не виконуватиметься, а вихідні дані інструкції не оновлюватимуться. Однак, якщо значення EN інструкції TRUE (1), функція інструкції буде виконана, а вихідні дані оновлено.

Вихідний стан ENO узгоджується зі станом EN. Коли EN має значення TRUE, ENO змінюється на TRUE. Коли EN має значення FALSE, ENO змінюється на FALSE.

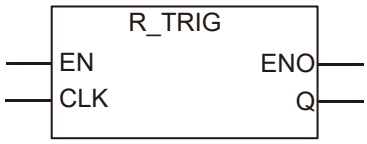
Якщо інструкція є функціональним блоком (FB) і її EN змінюється з TRUE на FALSE після виконання інструкції FB, виконання інструкції FB продовжуватиметься, але вихідні значення інструкції FB не оновлюватимуться.

8.3 Інструкції послідовного введення та Вихід

8.3.1 R_TRIG

FB/FC	Пояснення	Застосовна модель
FB	R_TRIG використовується для тригера наростаючого фронту.	AS516E-B AS532EST-B AS564EST-B

R_TRIG_екземпляр



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CLK	Вхідний сигнал	Вхід	Тригерний сигнал по висхідному фронту	Правда чи брехня
Q	Вихідний сигнал	Вихід	Вихід за період	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	●																			
Q	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Коли CLK для R_TRIG змінюється з FALSE на TRUE, вихід Q є TRUE лише протягом одного періоду. За інших обставин Q є ЛОЖНИМ.

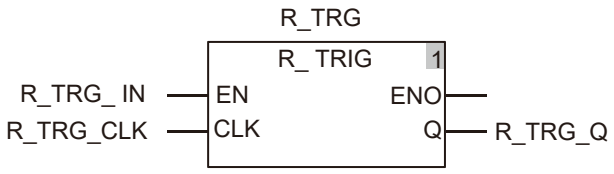
● **Застереження щодо правильного використання**

Q не матиме виходу, доки не буде виявлено наростаючий фронт сигналу на CLK.

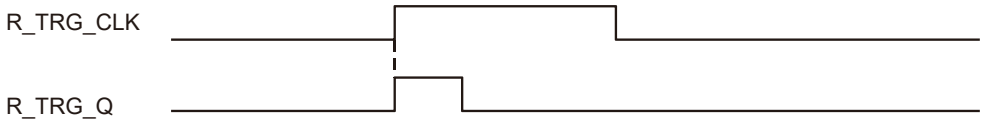
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
R_TRG	R_TRIG	
R_TRG_EN	BOOL	FALSE
R_TRG_CLK	BOOL	FALSE
R_TRG_Q	BOOL	



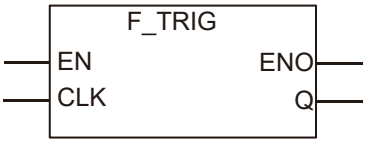
■ **Графік часу:**



8.3.2 F_TRIG

FB/FC	Пояснення	Застосовна модель
FB	F_TRIG використовується для тригера спадаючого фронту.	AS516E-B AS532EST-B AS564EST-B

F_TRIG_екземпляр



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CLK	Вхідний сигнал	Вхід	Тригерний сигнал із спадаючим фронтом	Правда чи брехня
Q	Вихідний сигнал	Вихід	Вихід за період	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLK	●																			
Q	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Коли CLK для F_TRIG змінюється з TRUE на FALSE, вихід Q є TRUE лише протягом одного періоду. За інших обставин Q є ЛОЖНИМ.

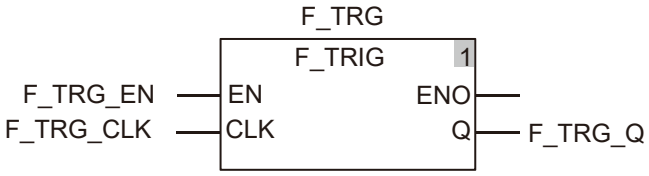
● **Застереження щодо правильного використання**

Q не матиме виходу, доки не буде виявлено спадаючий фронт сигналу на CLK.

Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
F_TRG	F_TRIG	
F_TRG_EN	BOOL	FALSE
F_TRG_CLK	BOOL	FALSE
F_TRG_Q	BOOL	



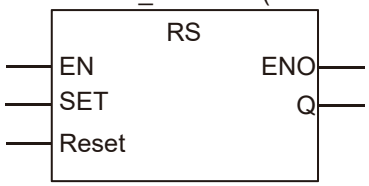
■ **Графік часу:**



8.3.3 RS

FB/FC	Пояснення	Застосовна модель
FB	RS використовується для надання пріоритету входу *Reset* .	AS516E-B AS532EST-B AS564EST-B

RS _instance (екземпляр)



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
SET	Вхідний сигнал	Вхід	сигнал SET	Правда чи брехня
Reset	Вхідний сигнал	Вхід	Скинути сигнал	Правда чи брехня
Q	Вихідний сигнал	Вихід	Вихідний сигнал	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET	●																			
Reset	●																			
Q	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

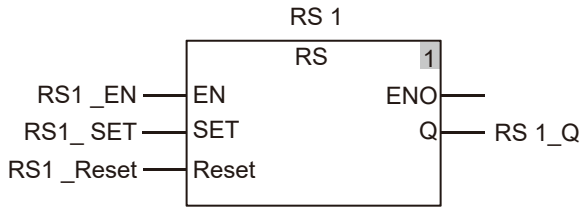
● **Пояснення функції**

Коли обидва входи *SET* і *Reset* RS мають значення TRUE, *Reset* надається пріоритет.

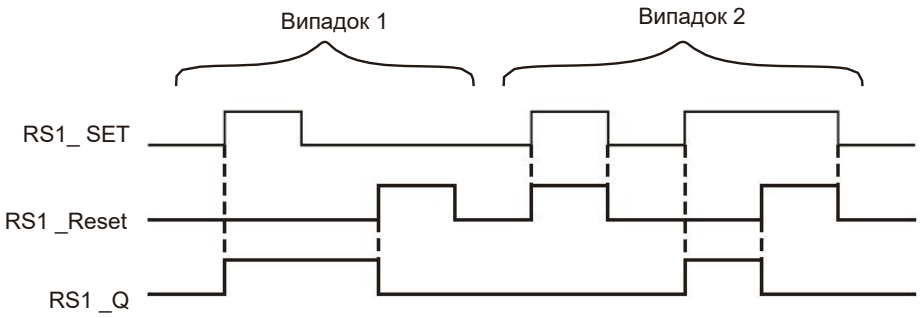
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
RS1	RS	
RS1_EN	BOOL	FALSE
RS1_SET	BOOL	FALSE
RS1_Reset	BOOL	FALSE
RS1_Q	BOOL	



■ Графік часу:

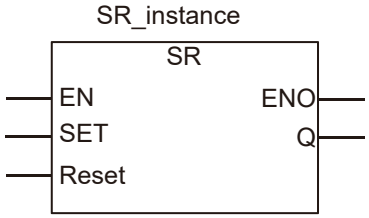


Випадок 1 : коли RS1_SET має значення TRUE, вихід RS1_Q має значення TRUE.
Якщо RS1_Reset має значення TRUE, RS1_Q має значення FALSE.

Випадок 2 : коли RS1_Reset має значення TRUE, RS1_Q завжди має значення FALSE.

8.3.4 SR

FB/FC	Пояснення	Застосовна модель
FB	SR використовується для надання пріоритету входу *Set* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
SET	Вхідний сигнал	Вхід	сигнал SET	Правда чи брехня
Reset	Вхідний сигнал	Вхід	Скинути сигнал	Правда чи брехня
Q	Вихідний сигнал	Вихід	Вихідний сигнал	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
SET	●																			
Reset	●																			
Q	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

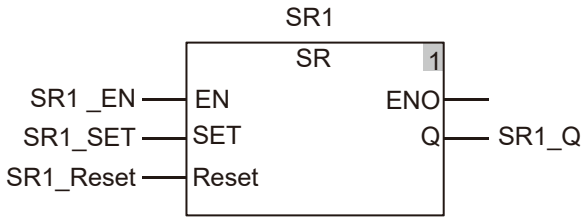
● **Пояснення функції**

Коли обидва входи *SET* і *Reset* RS мають значення TRUE, *SET* надається пріоритет.

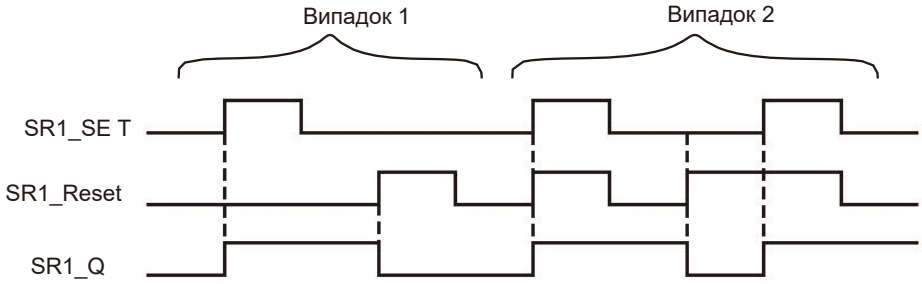
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
SR1	SR	
SR1_EN	BOOL	FALSE
SR1_SET	BOOL	FALSE
SR1_Reset	BOOL	FALSE
SR1_Q	BOOL	



■ Графік часу:

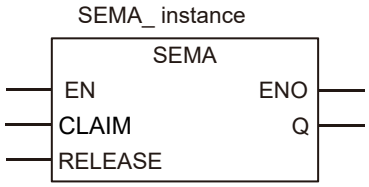


Випадок 1 : коли SR1_SET має значення TRUE, SR1_Q має значення TRUE. Коли SR1_Reset має значення TRUE, SR1_Q має значення FALSE.

Випадок 2 : SR1_SET надається пріоритет, коли SR1_SET і SR1_Reset мають значення TRUE.

8.3.5 SEMA

FB/FC	Пояснення	Застосовна модель
FB	SEMA використовується для надання пріоритету CLAIM. (Вихід буде дійсним у другому періоді.)	AS516E-B AS532EST-B AS564EST-B



- **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CLAIM	Вхідний сигнал	Вхід	Встановити сигнал	Правда чи брехня
RELEASE	Вхідний сигнал	Вхід	Скинути сигнал	Правда чи брехня
Q	Вихідний сигнал	Вихід	Вихідний сигнал	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CLAIM	●																			
RELEASE	●																			
Q	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

- **Пояснення функції**

  Коли *CLAIM* про SEMA є ІСТИННИМ, *Q* є ІСТИННИМ. Коли *RELEASE* має значення TRUE, *Q* має значення FALSE. Коли *ПОЗОВ* і *RELEASE* обидва є TRUE, *Q* є TRUE.

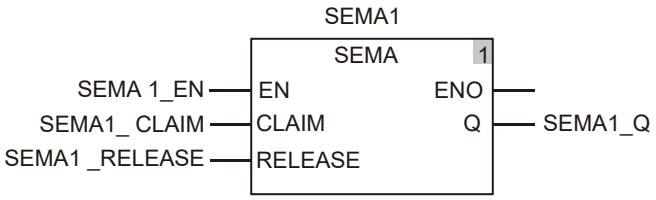
- **Застереження щодо правильного використання**

  Коли *CLAIM* має значення TRUE, Q буде TRUE у другому періоді.

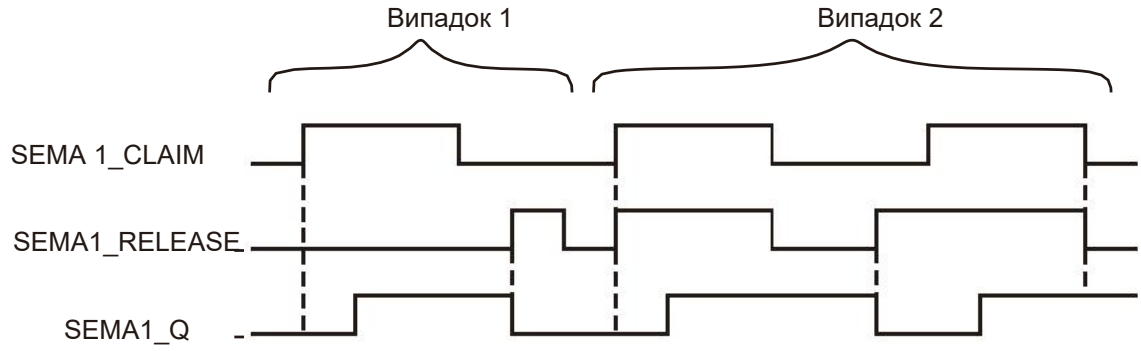
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
SEMA1	SEMA	
SEMA1_EN	BOOL	FALSE
SEMA1_CLAIM	BOOL	FALSE
SEMA1_RELEASE	BOOL	FALSE
SEMA1_Q	BOOL	



■ Графік часу:



Випадок 1 : Коли SEMA1_CLAIM має значення TRUE, SEMA1_Q має значення TRUE у другому періоді. Коли SEMA1_RELEASE має значення TRUE, SEMA1_Q негайно змінюється на FALSE.

Випадок 2 : Коли SEMA1_CLAIM має значення TRUE, SEMA1_Q має значення TRUE у другому періоді незалежно від того, чи SEMA1_RELEASE має значення TRUE чи FALSE.

8.4 Інструкції з контролю послідовності

8.4.1 JMP

FB/FC	Пояснення	Застосовна модель
FC	JMP використовується для переходу в будь-яку позицію, визначену міткою в програмі LD.	AS516E-B AS532EST-B AS564EST-B



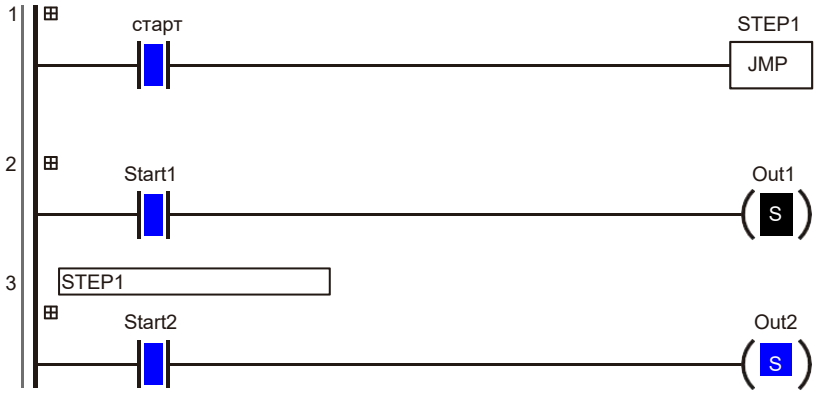
- **Пояснення функції**
  - JMP використовується для переходу в будь-яку позицію, визначену міткою в програмі LD.
- **Застереження щодо правильного використання**
  - Мітка - будь-який рядок.
  - Ключове слово не можна вказати як мітку.
  - JMP можна використовувати для стрибка вгору.
  - Більш ніж одна інструкція JMP може переходити до однієї мітки.
  - Позиції міток, до яких переходять інструкції JMP, мають бути в одному POU у програмі LD. Інакше стрибок не матиме ефекту.

Приклад програмування

- **Таблиця змінних**

Ім'я змінної	Тип даних	Поточне значення
Start	BOOL	TRUE
Start1	BOOL	TRUE
Out1	BOOL	FALSE
Start2	BOOL	TRUE
Out2	BOOL	TRUE

- **програма**



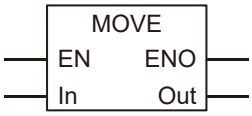
STEP1 береться як мітка переходу інструкції JMP у наведеній вище програмі. Коли **Start** зміниться на TRUE, інструкція JMP перейде до позиції мітки STEP1 і програма мережі 3 буде виконана.

Коли **Start2** змінюється на TRUE, **Out2** також змінюється на TRUE; програма мережі 2 не буде виконана. Коли **Пуск** змінюється на FALSE, усі програми від мережі 1 до мережі 3 будуть виконані.

8.5 Інструкції з переміщення даних

8.5.1 MOVE

FB/FC	Пояснення	Застосовна модель
FC	Move використовується для переміщення даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхідний сигнал	Вхід	Перемістити джерело	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Вихідний сигнал	Вихід	Перемістити пункт призначення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- ■ Інструкція Move переміщує значення джерела переміщення *In* до переміщення пункту призначення *Out* .
- ■ Інструкція підтримує передачу значень елементів масиву.

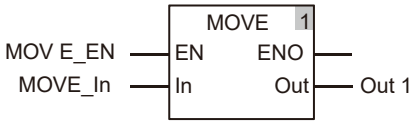
● **Застереження щодо правильного використання**

Тип даних *Out* має бути таким самим, як і *In* . В іншому випадку під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

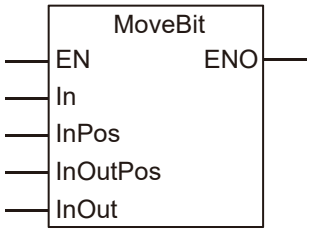
■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
MOVE_EN	BOOL	TRUE
MOVE_In	INT	200
Out1	INT	200


MOVE
1
MOV E_EN
EN
ENO
MOVE_In
In
Out
Out 1

8.5.2 MoveBit

FB/FC	Пояснення	Застосовна модель
FC	MoveBit використовується для надсилання одного біта в рядку.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхідний сигнал	Вхід	Перемістити джерело	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InPos	Вхідний сигнал	Вхід	Перемістити вихідний біт	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOutPos	Вхідний сигнал	Вхід	Перемістити біт призначення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOut	Вхідний сигнал	Вхід	Перемістити пункт призначення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
InOut		●	●	●	●	●	●	●	●											

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

MoveBit переміщує одне бітове значення з бітової позиції *InPos* у джерелі переміщення *In* до бітової позиції *InOutPos*
у місці призначення переміщення *InOut* .

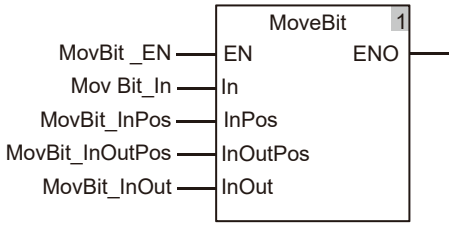
● **Застереження щодо правильного використання**

- Інструкція має не вихід, а вхід.
- Якщо значення *InPos* перевищує діапазон типу даних *In* , переміщення одного біта не виконується.
- Якщо значення *InOutPos* перевищує діапазон типу даних *InOut* , переміщення одного біта не виконується.

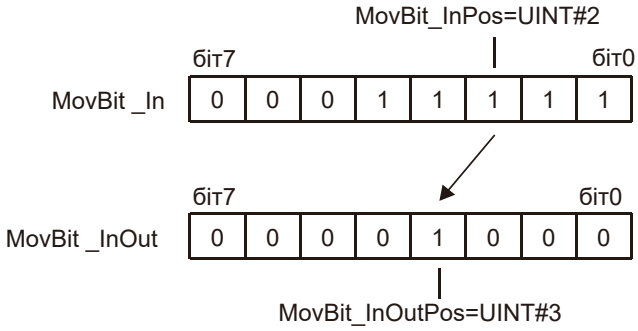
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
MovBit_EN	BOOL	TRUE
MovBit_In	USINT	31
MovBit_Inpos	UINT	2
MovBit_InOutPos	UINT	3
MovBit_Inout	USINT	8



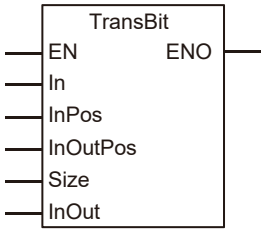
■ **Перемістити фігуру**



8

8.5.3 TransBit

FB/FC	Пояснення	Застосовна модель
FC	TransBit використовується для надсилання одного або кількох бітів у бітовому рядку.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхідний сигнал	Вхід	Перемістити джерело	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InPos	Вхідний сигнал	Вхід	Перемістити вихідний біт	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOutPos	Вхідний сигнал	Вхід	Перемістити біт призначення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Size	Вхідний сигнал	Вхід	Кількість бітів для переміщення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOut	Вхідний сигнал	Вхід	Перемістити пункт призначення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
Size							●													
InOut		●	●	●	●	●	●	●	●											

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

TransBit переміщує дані бітів *розміру* з біта *InPos* у джерелі переміщення *In* до біта *InOutPos* у місці призначення *InOut переміщення* .

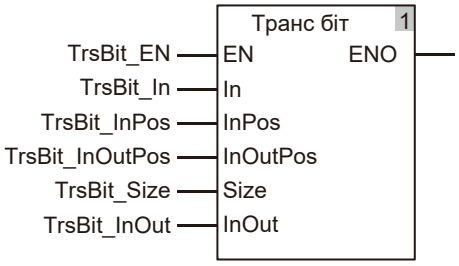
● **Застереження щодо правильного використання**

- Інструкція має не вихід, але вхід.
- Переміщення неможливо виконати, якщо значення *розміру* дорівнює 0.
- Якщо значення *InPos* перевищує діапазон типу даних *In* , переміщення не виконується.
- Якщо значення *InOutPos* перевищує діапазон типу даних *InOut* , переміщення не виконується.
- Якщо значення *розміру* перевищує діапазон, переміщення не виконується.

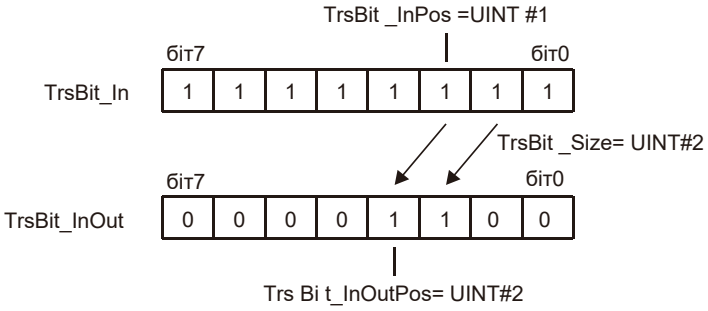
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
TrsBit_EN	BOOL	TRUE
TrsBit_In	USINT	63
TrsBit_InPos	UINT	1
TrsBit_InOutPos	UINT	2
TrsBit_Size	UINT	2
TrsBit_Inout	USINT	12



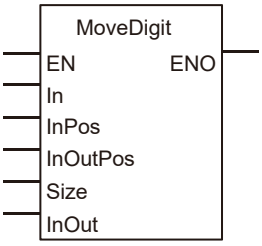
■ **Перемістити фігуру**



8

8.5.4 MoveDigit

FB/FC	Пояснення	Застосовна модель
FC	MoveDigit використовується для переміщення цифр.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхідний сигнал	Вхід	Перемістити джерело	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InPos	Вхідний сигнал	Вхід	Позиція цифри в *In* для переміщення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOutPos	Вхідний сигнал	Вхід	Позиція цифри в *Out* для отримання цифри	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Size	Вхідний сигнал	Вхід	Кількість цифр для переміщення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
InOut	Вхідний сигнал	Вхід	Перемістити пункт призначення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
InPos							●													
InOutPos							●													
Size							●													
InOut		●	●	●	●	●	●	●	●											

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

MoveDigit переміщує цифри *розміру* з *InPos* джерела переміщення *In* до *InOutPos* призначення переміщення *InOut* .

● **Застереження щодо правильного використання**

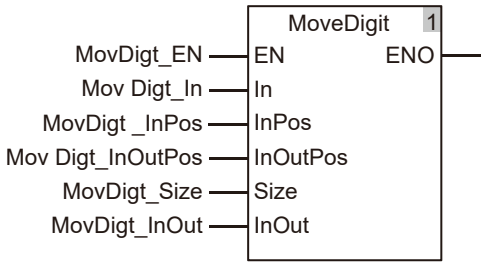
■ Інструкція не має вихідного, але вхідного параметра.

- Переміщення неможливо виконати, якщо значення *розміру* дорівнює 0.
- Якщо значення *InPos* перевищує діапазон типу даних *In* , переміщення не буде виконано.
- Якщо значення *InOutPos* перевищує діапазон типу даних *InOut* , переміщення не виконується.
- Якщо значення *розміру* перевищує діапазон, переміщення не виконується.

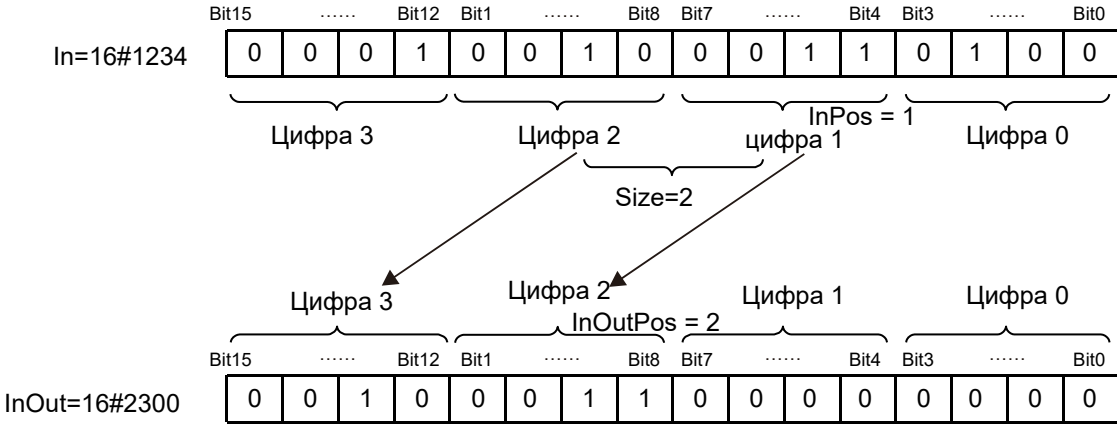
Приклад програмування

Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
MovDigt_EN	BOOL	TRUE
MovDigt_In	UDINT	16#1234
MovDigt_InPos	UINT	1
MovDigt_InOutPos	UINT	2
MovDigt_Size	UINT	2
MovDigt_Inout	UDINT	16#2300

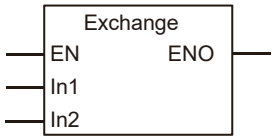


Перемістити фігуру



8.5.5 Обмін

FB/FC	Пояснення	Застосовна модель
FC	Exchange використовується для обміну даними.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Вхідний сигнал	Вхід	Дані для обміну	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Вхідний сигнал	Вхід	Дані для обміну	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
In2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція обміну міняє значення *In1* та *In2* .

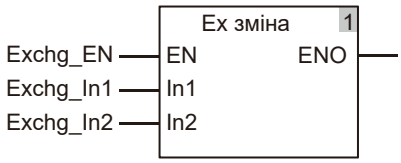
● **Застереження щодо правильного використання**

- Типи даних *In1* і *In2* мають бути однаковими.
- Інструкція не має вихідних даних, але має два вхідних параметри.

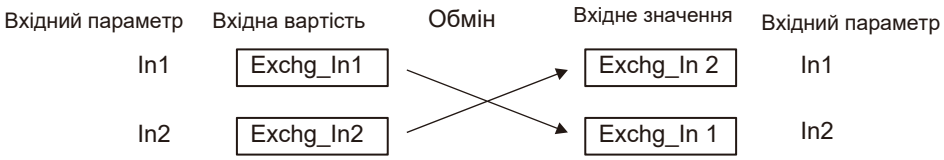
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
Exchg_EN	BOOL	TRUE
Exchg_In1	INT	30
Exchg_In2	INT	10



■ Фігура обміну

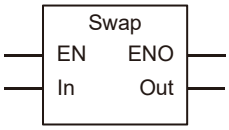


Значення In 1 та In2 міняються місцями.

Під час виконання інструкції обміну значення Exchg_In1 і Exchg_In2 завжди обмінюються.

8.5.6 Обмін

FB/FC	Пояснення	Застосовна модель
FC	Swap використовується для заміни старшого та молодшого байтів 16-бітного значення.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхідний сигнал	Вхід	Дані для обміну	0~65535 для типу даних Word
Out	Вихідний сигнал	Вихід	Результат	0~65535 для типу даних Word

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In			●				●													
Out			●				●													

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

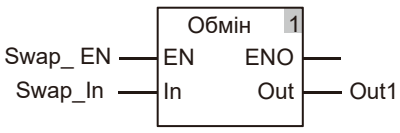
● **Пояснення функції**

Інструкція Swap міняє старший і молодший байт значення *In* і результат виводиться на *Out*

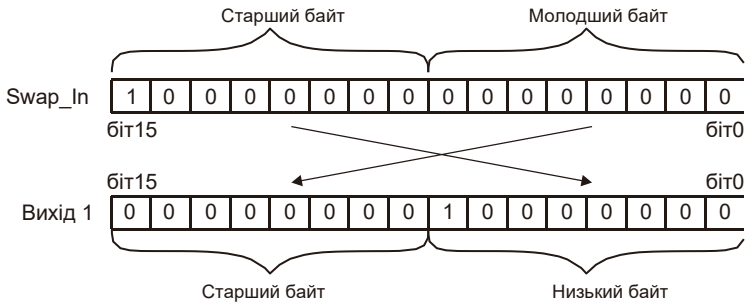
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
Swap_EN	BOOL	TRUE
Swap_In	UINT	32768
Out1	UINT	128



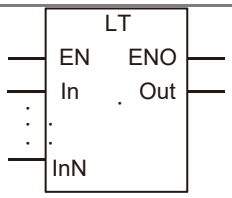
■ **Поміняйте рисунок**



8.6 Інструкція порівняння

8.6.1 LT

FB/FC	Пояснення	Застосовна модель
FC	LT використовується для меншого порівняння двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Кількість порівняльних даних можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2~8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ LT використовується для меншого порівняння двох або більше змінних або констант.

*Я*кщо *In1* < *In2* < ... < *InN, Out* має значення TRUE *(*вірний*)*. В іншому випадку *Out* є FALSE (хибний).

■ вхідні параметри *In1~InN* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Коли тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, усі вхідні параметри In1~InN повинні мати тип даних. Наприклад, якщо тип даних In1 є

TIME, тип даних In2~InN має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типи даних LT_In1, LT_In2 і LT_In3 — INT, UINT і DINT відповідно, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення LT_In1, LT_In2 і LT_In3 становлять -10, 50 і 100 відповідно, а LT_EN змінюється на TRUE, як показано у змінній 1.
  Out1 змінюється на FALSE, коли значення LT_In1, LT_In2 і LT_In3 становлять 20, 10 і 100 відповідно, а LT_EN змінюється на TRUE, як показано у змінній 2.

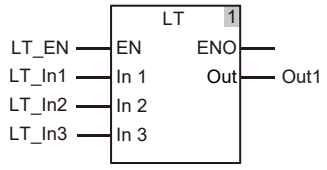
  ➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
LT_EN	BOOL	TRUE
LT_In1	INT	-10
LT_In2	UINT	50
LT_In3	DINT	100
Out1	BOOL	TRUE

  ➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
LT_EN	BOOL	TRUE
LT_In1	INT	20
LT_In2	UINT	10
LT_In3	DINT	100
Out1	BOOL	FALSE

  ➢ **Програма**

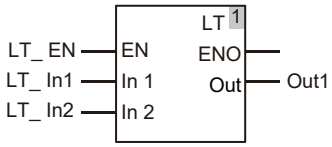


- **Обидва типи даних LT_In1 і LT_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення LT_In1 і LT_In2 становлять T#1ms і T#50ms відповідно, а LT_EN — TRUE.

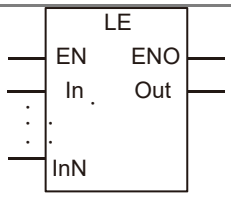
  ➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
LT_EN	BOOL	TRUE
LT_In1	TIME	T#1ms
LT_In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.6.2 LE

FB/FC	Пояснення	Застосовна модель
FC	LE використовується для меншого або рівного порівняння двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Кількість порівняльних даних можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- LE використовується для порівняння двох або більше змінних або констант, що менше або рівно. якщо *In1* ≤ *In2* ≤ ... ≤ *InN, Out* – ІСТИНА. В іншому випадку *Out* є FALSE.
- вхідні параметри *In1~InN* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Коли тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, усі вхідні параметри In1~InN повинні мати тип даних. Наприклад, якщо типом даних In1 є TIME, типом даних In2~InN має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типи даних LE_In1, LE_In2 і LE_In3 — INT, UINT і DINT відповідно, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення LE_In1, LE_In2 і LE_In3 становлять -10, 50 і 50 відповідно, а LE_EN змінюється на TRUE, як показано у змінній 1.

  Out1 змінюється на FALSE, коли значення LE_In1, LE_In2 і LE_In3 становлять 20, 10 і 100 відповідно, а LE_EN змінюється на TRUE, як показано у змінній 2.

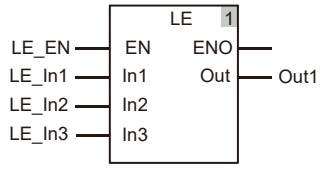
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
LE_EN	BOOL	TRUE
LE_In1	INT	-10
LE_In2	UINT	50
LE_In3	DINT	50
Out1	BOOL	TRUE

➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
LE_EN	BOOL	TRUE
LE_In1	INT	20
LE_In2	UINT	10
LE_In3	DINT	100
Out1	BOOL	FALSE

➢ **Програма**

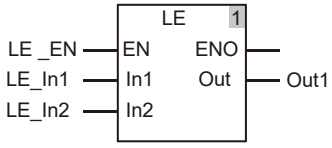


- **Обидва типи даних LE_In1 і LE_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення LE_In1 і LE_In2 становлять T#1ms і T#50ms відповідно, а LE_EN — TRUE.

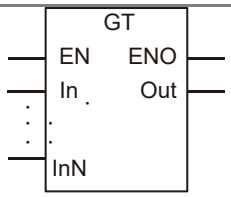
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
LE_EN	BOOL	TRUE
LE_In1	TIME	T#1ms
LE_In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.6.3 GT

FB/FC	Пояснення	Застосовна модель
FC	GT використовується для порівняння двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Кількість порівняльних даних можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- LE використовується для більшого порівняння двох або більше змінних або констант. якщо *In1 > In2 > ... > InN, Out* має значення TRUE . В іншому випадку *Out* є FALSE.
- вхідні параметри *In1~InN* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Коли тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, усі вхідні параметри In1~InN повинні мати тип даних. Наприклад, якщо типом даних In1 є TIME, типом даних In2~InN має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типами даних GT_In1, GT_In2 і GT_In3 є INT, UINT і DINT відповідно, а типом даних Out1 є BOOL.**

  Out1 змінюється на TRUE, коли значення GT_In1, GT_In2 і GT_In3 становлять 100, 50 і 10 відповідно, а GT_EN змінюється на TRUE, як показано у змінній 1.

  Out1 змінюється на FALSE, коли значення GT_In1, GT_In2 і GT_In3 становлять 20, 10 і 100 відповідно, а GT_EN змінюється на TRUE, як показано у змінній 2.

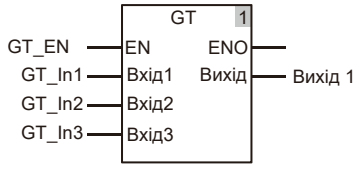
  ➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
GT_EN	BOOL	TRUE
GT _In1	INT	100
GT _In2	UINT	50
GT _In3	DINT	10
Out1	BOOL	TRUE

  ➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
GT_EN	BOOL	TRUE
GT _In1	INT	20
GT _In2	UINT	10
GT _In3	DINT	100
Out1	BOOL	FALSE

  ➢ **Програма**

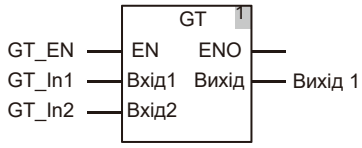


- **Обидва типи даних GT_In1 і GT_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення GT_In1 і GT_In2 становлять T#100ms і T#50ms відповідно, а GT_EN змінюється на TRUE.

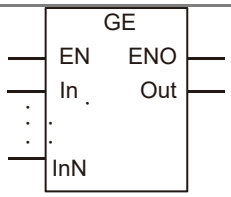
  ➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
GT_EN	BOOL	TRUE
GT _In1	TIME	T#100 мс
GT _In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.6.4 GE

FB/FC	Пояснення	Застосовна модель
FC	GE використовується для порівняння двох або більше змінних або констант на рівні більше або більше.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Кількість порівняльних даних можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- GE використовується для більшого або рівного порівняння двох або більше змінних або констант. якщо *In1* ≥ *In2* ≥ ... ≥ *InN, Out* є TRUE. В іншому випадку *Out* є FALSE.
- вхідні параметри *In1~InN* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Коли тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, усі вхідні параметри In1~InN повинні мати тип даних. Наприклад, якщо типом даних In1 є TIME, типом даних In2~InN має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типи даних GE_In1, GE_In2 і GE_In3 є INT, UINT і DINT відповідно, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення GE_In1, GE_In2 і GE_In3 становлять 100, 50 і 50 відповідно, а GE_EN змінюється на TRUE, як показано у змінній 1.

  Out1 змінюється на FALSE, коли значення GE_In1, GE_In2 і GE_In3 становлять 10, 10 і 100 відповідно, а GE_EN змінюється на TRUE, як показано у змінній 2.

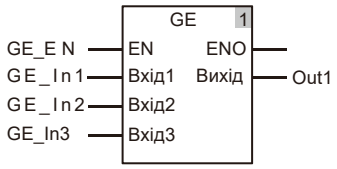
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
GE_EN	BOOL	TRUE
GE_In1	INT	100
GE_In2	UINT	50
GE_In3	DINT	50
Out1	BOOL	TRUE

➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
GE_EN	BOOL	TRUE
GE_In1	INT	10
GE_In2	UINT	10
GE_In3	DINT	100
Out1	BOOL	FALSE

➢ **Програма**

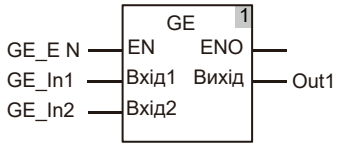


- **Обидва типи даних GE_In1 і GE_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення GE_In1 і GE_In2 становлять T#100ms і T#50ms відповідно, а GE_EN змінюється на TRUE.

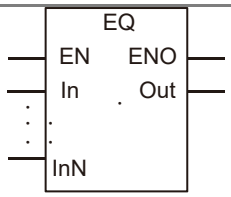
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
GE_EN	BOOL	TRUE
GE_In1	TIME	T#100 мс
GE_In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.6.5 EQ

FB/FC	Пояснення	Застосовна модель
FC	EQ використовується для рівного порівняння двох або більше змінних і констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Кількість порівняльних даних можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

1. EQ використовується для рівного порівняння двох або більше змінних і констант. Якщо *In1* = *In2* = ... = *InN, Out* є TRUE . В іншому випадку *Out* є FALSE.
2. вхідні параметри *In1~InN* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Коли тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, усі вхідні параметри In1~InN повинні мати тип даних. Наприклад, якщо типом даних In1 є TIME, типом даних In2~InN має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типи даних EQ_In1, EQ_In2 і EQ_In3 — INT, UINT і DINT відповідно, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення EQ_In1, EQ_In2 і EQ_In3 становлять 50, 50 і 50 відповідно, а EQ_EN змінюється на TRUE, як показано у змінній 1.

  Out1 змінюється на FALSE, коли значення EQ_In1, EQ_In2 і EQ_In3 становлять 10, 50 і 100 відповідно, а EQ_EN змінюється на TRUE, як показано у змінній 2.

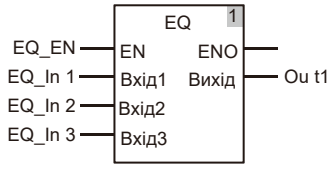
  ➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
EQ_EN	BOOL	TRUE
EQ _In1	INT	50
EQ _In2	UINT	50
EQ _In3	DINT	50
Out1	BOOL	TRUE

  ➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
EQ_EN	BOOL	TRUE
EQ _In1	INT	10
EQ _In2	UINT	50
EQ _In3	DINT	100
Out1	BOOL	FALSE

  ➢ **Програма**

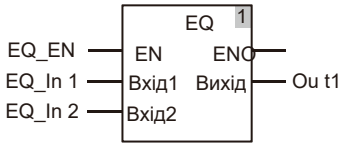


- **Обидва типи даних EQ_In1 і EQ_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення EQ_In1 і EQ_In2 становлять T#50ms і T#50ms відповідно, а EQ_EN змінюється на TRUE.

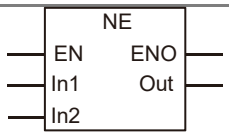
  ➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
EQ_EN	BOOL	TRUE
EQ _In1	TIME	T#50 мс
EQ _In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.6.6 NE

FB/FC	Пояснення	Застосовна модель
FC	NE використовується для нерівного порівняння двох змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Дані порівняння	Вхід	Значення для порівняння	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Дані порівняння	Вхід	Значення для порівняння	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Результат порівняння	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 і In2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

8

● **Пояснення функції**

- NE використовується для нерівного порівняння двох змінних і констант. *Out* є TRUE, якщо *In1*≠*In2.* В іншому випадку *Out* є FALSE.
- вхідні параметри *In1* і *In2* можуть бути змінними різних типів даних, якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD і STRING. Якщо тип даних однієї вхідної змінної є одним із BOOL, TIME, DATE, TOD і STRING, обидва вхідні параметри *In1* та *In2* повинні мати тип даних. Наприклад, якщо тип даних *In1* – TIME, тип даних *In2* має бути TIME. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

- Тип даних вихідних змінних має бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- **Типи даних NE_In1 і NE_In2 — INT і DINT відповідно, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення NE_In1 і NE_In2 становлять 100 і 50 відповідно, а NE _EN змінюється на TRUE, як показано у змінній 1.

  Out1 змінюється на FALSE, коли значення NE_In1 і NE _In2 становлять 100 і 100 відповідно, а NE_EN змінюється на TRUE, як показано у змінній 2.

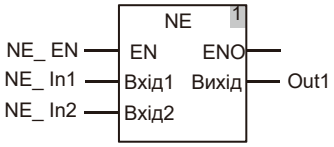
  - **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
NE_EN	BOOL	TRUE
NE _In1	INT	100
NE _In2	UINT	50
Out1	BOOL	TRUE

  - **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
NE_EN	BOOL	TRUE
NE _In1	INT	100
NE _In2	UINT	100
Out1	BOOL	FALSE

  - **Програма**

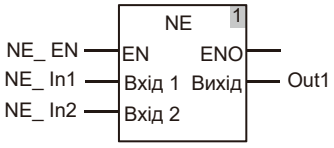


- **Обидва типи даних NE_In1 і NE_In2 — TIME, а тип даних Out1 — BOOL.**

  Out1 змінюється на TRUE, коли значення NE_In1 і NE_In2 становлять T#10ms і T#50ms відповідно, а NE_EN змінюється на TRUE.

  - **Таблиця змінних і програма**

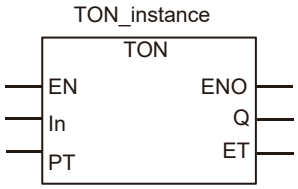
Ім'я змінної	Тип даних	Поточне значення
NE_EN	BOOL	TRUE
NE _In1	TIME	T#10 мс
NE _In2	TIME	T#50 мс
Out1	BOOL	TRUE



8.7 Інструкції таймера

8.7.1 TON

FB/FC	Пояснення	Застосовна модель
FB	TON використовується для затримки увімкнення.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхід таймера	Вхід	Керує таймером для запуску або скидання	Правда чи брехня
PT	Встановити час	Вхід	Час від моменту запуску таймера до зміни Q на TRUE.	
Q	Вихід таймера	Вихід	досягнуто встановлений час *PT* .	Правда чи брехня
ET	Витрачений час	Вихід	Час, що минув від моменту запуску таймера до поточного часу.	

◆ T#0ms ~ 213503d23h34m33s709,551ms

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
ET																●				

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція TON визначається як функція таймера для затримки увімкнення.

■ Коли *In* дорівнює TRUE, таймер починає вимірювати час, і значення *ET* відповідно збільшується. Коли *ET* дорівнює PT, Q має значення ІСТИНА. Коли для *In* встановлено значення FALSE, вимірювання часу припиняється, а *Q* і *ET* скидаються.

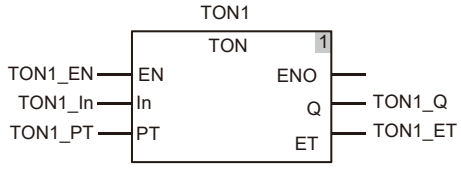
● **Застереження щодо правильного використання**

Коли вихідне значення *ET* досягає встановленого значення *PT* , таймер припиняє вимірювання часу. *ET* скидається до 0 (0 мс), коли *In* змінюється з TRUE на FALSE.

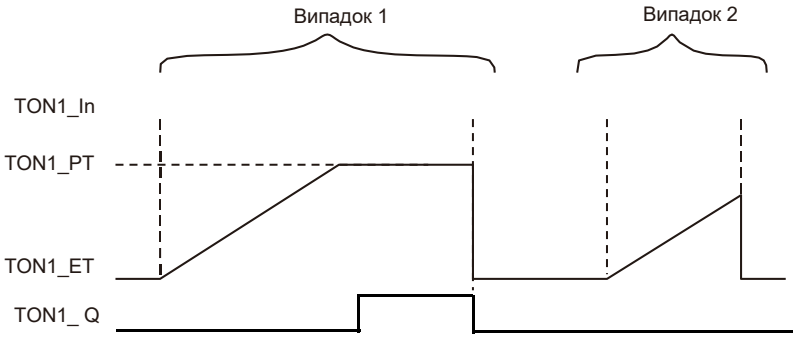
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
TON1	TON	
TON1_EN	BOOL	FALSE
TON1_In	BOOL	FALSE
TON1_PT	TIME	
TON1_Q	BOOL	
TON1_ET	TIME	



■ **Графік часу:**



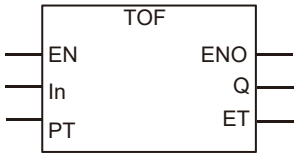
Випадок 1 : TON1_PT – встановлений час. Коли TON1_In дорівнює TRUE, таймер починає вимірювати час. Коли значення TON1_ET дорівнює значенню налаштування TON1_PT, TON1_Q має значення TRUE. Коли таймер припиняє вимірювати час, TON1_In скидається на FALSE, а TON1_ET і TON1_Q скидаються.

Випадок 2 : Коли поточний виміряний час таймера TON1_ET менший за встановлений час TON1_PT і TON1_In скидається на FALSE, TON1_ET скидається, а стан TON1_Q не змінюється.

8.7.2 TOF

FB/FC	Пояснення	Застосовна модель
FB	TOF використовується для затримки вимкнення.	AS516E-B AS532EST-B AS564EST-B

TOF_примірник



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхід таймера	Вхід	Керує таймером для запуску або скидання	Правда чи брехня
PT	Встановити час	Вхід	Встановіть час від моменту запуску таймера до зміни Q на TRUE	
Q	Вихід таймера	Вихід	досягнуто встановлений час *PT* .	Правда чи брехня
ET	Витрачений час	Вихід	Час, що минув від моменту запуску таймера до поточного часу.	

◆ T#0ms ~ 213503d23h34m33s709,551ms

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
I																●				

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- Інструкція TOF визначається як функція таймера для затримки ВИМКНЕННЯ.
- Коли *In* дорівнює ІСТИНІ, *Q* дорівнює ІСТИНІ. Коли *In* змінюється з TRUE на FALSE, таймер починає вимірювати час, і значення *ET* відповідно збільшується. На даний момент Q залишається TRUE. Коли *ET* дорівнює PT, Q є FALSE і таймер припиняє вимірювати час. Коли *In* встановлено на TRUE, ET скидається, а *Q* знову змінюється на TRUE.

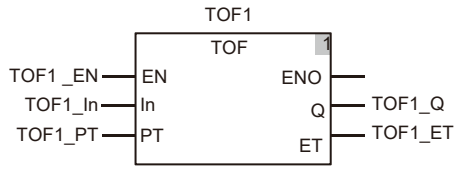
● **Застереження щодо правильного використання**

Коли вихідне значення *ET* досягає встановленого значення *PT* , таймер припиняє вимірювання часу. *ET* скидається до 0 (0 мс), коли *In* змінюється з FALSE на TRUE.

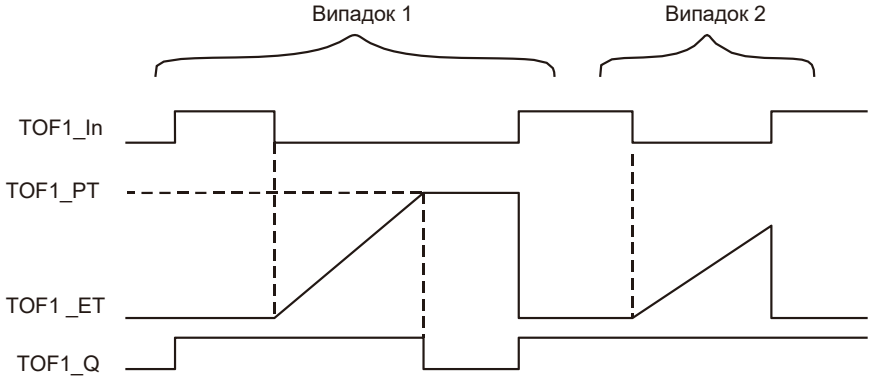
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
TOF1	TOF	
TOF1_EN	BOOL	FALSE
TOF1_In	BOOL	FALSE
TOF1_PT	TIME	
TOF1_Q	BOOL	
TOF1_ET	TIME	



■ Графік часу:



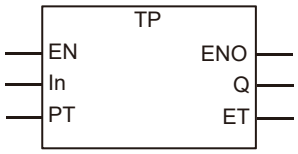
Випадок 1 : TOF1_PT – встановлений час для затримки вимкнення. Коли TOF1_In має значення TRUE, TOF1_Q має значення TRUE. Коли TOF1_In має значення FALSE, таймер починає вимірювати час. Коли значення TOF1_ET дорівнює встановленому значенню TOF1_PT, TOF1_Q має значення FALSE і таймер припиняє відлік часу.

Випадок 3 : Коли TOF1_In змінюється з TRUE на FALSE, таймер починає вимірювати час. Коли поточний час (TOF1_ET) менший за встановлений час (TOF1_PT) і TOF1_In встановлено на TRUE, TOF1_ET скидається, а стан TOF1_Q не змінюється.

8.7.3 TP

FB/FC	Пояснення	Застосовна модель
FB	TP використовується для затримки вимкнення після того, як вхід **In** є TRUE.	AS516E-B AS532EST-B AS564EST-B

TP_примірник



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Вхід таймера	Вхід	Керує таймером для запуску або скидання	Правда чи брехня
PT	Встановити час	Вхід	Встановіть час від моменту запуску таймера до зміни Q на TRUE	◆
Q	Вихід таймера	Вихід	досягнуто встановлений час *PT* .	Правда чи брехня
ET	Витрачений час	Вихід	Час, що минув від моменту запуску таймера до поточного часу.	◆

◆ T#0ms ~ 213503d23h34m33s709,551ms

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
PT																●				
Q	●																			
I																●				

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Коли *In* дорівнює TRUE, *Q* дорівнює TRUE, і таймер починає вимірювати час, а значення *ET* відповідно збільшується. На даний момент Q залишається TRUE. Коли *ET* дорівнює PT, Q є FALSE і таймер припиняє вимірювати час. Коли *In* змінюється з TRUE на FALSE, ET скидається.

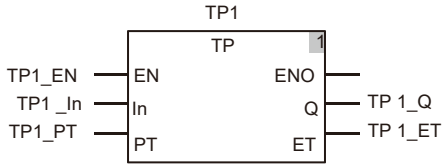
● **Застереження щодо правильного використання**

Коли вихідне значення *ET* досягає встановленого значення *PT* , таймер припиняє вимірювання часу. *ET* скидається до 0 (0 мс), коли *In* змінюється з TRUE на FALSE.

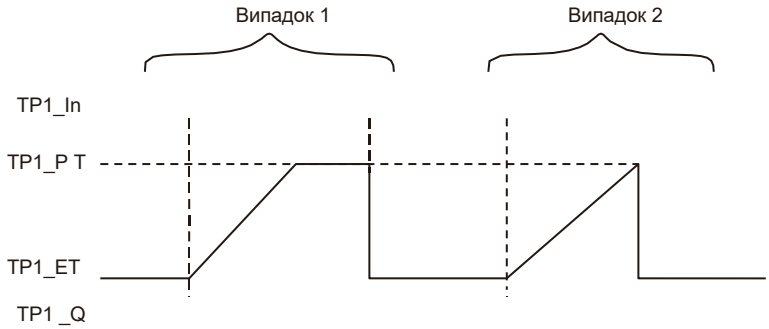
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
TP1	TP	
TP1_EN	BOOL	FALSE
TP1_In	BOOL	FALSE
TP1_PT	TIME	
TP1_Q	BOOL	
TP1_ET	TIME	



■ Графік часу:

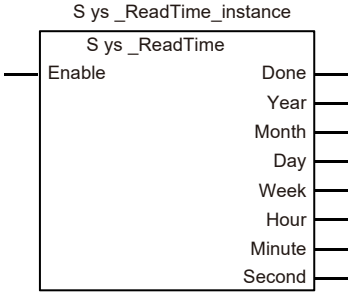


Випадок 1 : TP1_PT встановлює час затримки вимкнення. Коли TP1_In дорівнює TRUE, таймер починає вимірювати час, а TP1_Q дорівнює TRUE. Коли значення TP1_ET дорівнює значенню налаштування TP1_PT, TP1_Q має значення FALSE. Коли TP1_In має значення FALSE, TP1_ET скидається.

Випадок 2 : TP1_PT встановлює час затримки вимкнення. Коли TP1_In має значення TRUE і таймер починає вимірювати час, TP1_Q має значення TRUE. Коли TP1_In дорівнює FALSE, а значення TP1_ET менше значення налаштування TP1_PT, TP1_ET зберігає синхронізацію, а TP1_Q зберігає стан TRUE. Коли значення TP1_ET дорівнює значенню налаштування TP1_PT, TP1_ET і TP1_Q скидаються.

8.7.4 Sys_ReadTime

FB/FC	Пояснення	Застосовна модель
FB	Sys_ReadTime читає час годинника реального часу контролера.	AS516E-B AS532EST-B AS564EST-B



- **Вхідний**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE.

- **Вихідні**

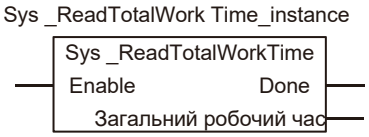
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Year	рік	UINT	1970~2106
Month	місяць	UINT	1~12
Day	День	UINT	1 ~ 31
Week	тиждень	UINT	1~7
Hour	годину	UINT	0~23
Minute	хвилина	UINT	0~59
Second	секундаге	UINT	0~59

- **Пояснення функції**

Sys_ReadTime читає час годинника реального часу контролера. Якщо *Enable* має значення TRUE, інформація про годинник реального часу в контролері, наприклад рік, місяць, день, тиждень, година, хвилина та секунда, буде зчитуватися у вказаних змінних.

8.7.5 Sys_ReadTotalWorkTime

FB/FC	Пояснення	Застосовна модель
FB	Sys_ReadTotalWorkTime читає загальний час роботи контролера.	AS516E-B AS532EST-B AS564EST-B



- **Вхідний**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE.

- **Вихідні**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
TotalWorkTime	Показує загальний час роботи контролера.	TIME	

- T#0ms ~ 213503d23h34m33s709,551ms

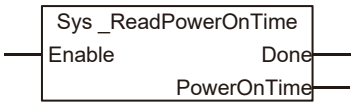
- **Пояснення функції**

  Sys_ReadTotalWorkTime читає загальний час роботи контролера. Якщо *Enable* має значення TRUE, загальний час роботи контролера зчитується у змінній, визначеній *TotalWorkTime* . Наприклад, вчора контролер працював 3 години, а сьогодні – 2 години. Таким чином, загальний час роботи контролера становить 5 годин.

8.7.6 Sys_ReadPowerOnTime

FB/FC	Пояснення	Застосовна модель
FB	Sys_ReadPowerOnTime читає час увімкнення живлення контролера.	AS516E-B AS532EST-B AS564EST-B

Sys _ReadP owerOnTime_instance



- **Вхідний**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE.

- **Вихідні**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
PowerOnTime	Показує загальний час увімкнення контролера.	TIME	

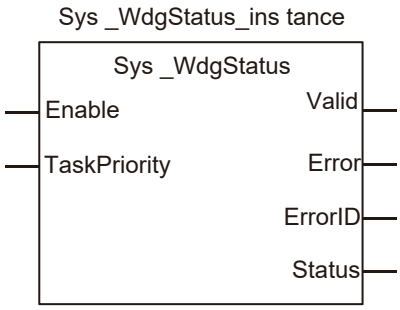
◆ T#0ms ~ 213503d23h34m33s709,551ms

- **Пояснення функції**

Sys_ReadPowerOnTime читає загальний час увімкнення живлення контролера. Інструкція перезапуститься, щоб виміряти час увімкнення живлення, якщо після вимкнення живлення контролера буде відновлено. Якщо *Enable* має значення TRUE, загальний час увімкнення контролера буде зчитано у змінній, указаній *PowerOnTime* .

8.7.7 Sys_WdgStatus

FB/FC	Пояснення	Застосовна модель
FB	Інструкція Sys_WdgStatus використовується для зчитування того, чи перевищує час виконання зазначеного завдання дозволений час налаштування (час сторожового таймера).	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	Коли *Enable* змінюється на TRUE
TaskPriority	Встановити пріоритет завдання, тобто вказати завдання через його пріоритет.	UINT	1~24 (0)	Коли *Enable* змінюється на TRUE

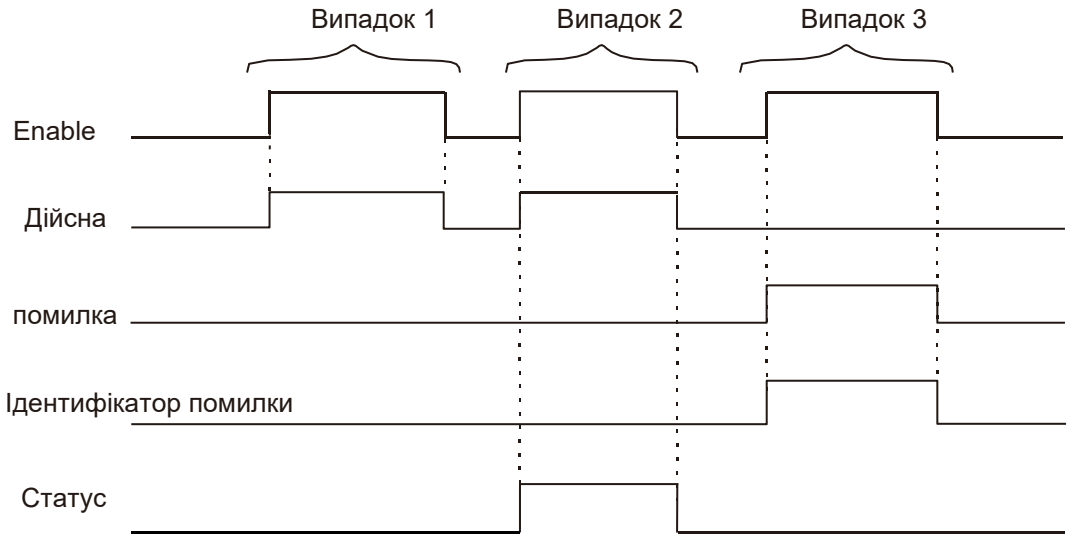
- **Вихідні**

Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	ІСТИНА, коли вихід інструкції дійсний.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	
Статус	TRUE, коли час виконання зазначеного завдання перевищує дозволений час налаштування (час сторожового таймера).	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
Помилка	◆ Коли вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE
Статус	◆ Коли час виконання зазначеного завдання перевищує дозволений час налаштування (час сторожового таймера).	◆ Коли час виконання зазначеного завдання не перевищує дозволений час налаштування (час сторожового таймера).

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : *дійсні* зміни з FALSE на TRUE, коли *Enable* змінюється з FALSE на TRUE.

Випадок 2 : *Дійсні* зміни з FALSE на TRUE, коли *Enable* змінюється з FALSE на TRUE. Коли час виконання завдання перевищує час сторожового таймера, *Cmamyc* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE, а *Status* змінюється на FALSE.

Випадок 3 : коли виникає помилка, коли *Enable* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

- **функція**

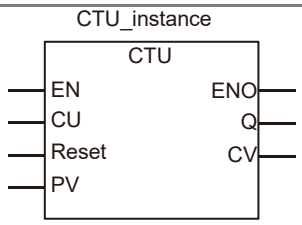
Інструкція Sys_WdgStatus використовується для зчитування того, чи перевищує час виконання зазначеного завдання дозволений час налаштування (час сторожового таймера). Коли використовується ця інструкція, не розміщуйте її в завданні, визначеному *TaskPriority. Й*ого слід розмістити в іншому завданні для виконання.

Вихід інструкції Status змінюється з FALSE на TRUE, виконання завдання, визначеного TaskPriority *, припиняється . Cmamyc* інструкції може бути використаний для виконання відповідної операції в іншому завданні користувачами.

8.8 Зустрічні інструкції

8.8.1 CTU

FB/FC	Пояснення	Застосовна модель
FB	CTU використовується як лічильник підвищення.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CU	Вхідний сигнал лічильника	Вхід	Керуйте лічильником, щоб почати підрахунок	Правда чи брехня
Reset	Скинути сигнал	Вхід	Скинути поточне значення лічильника	Правда чи брехня
PV	Попереднє значення	Вхід	Значення установки лічильника	0 ~ 4294967295
Q	Вихідний сигнал	Вихід	Q є ІСТИННИМ, коли CV дорівнює PV.	Правда чи брехня
CV	Лічильник значення	Вихід	Лічильник теперішньої вартості	0 ~ 4294967295

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
Reset	●																			
PV								●												
Q	●																			
CV								●												

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

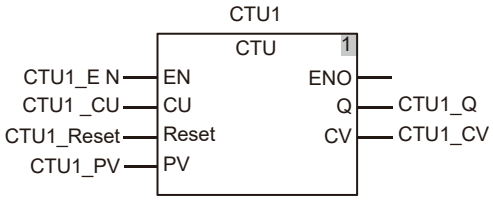
1. CTU функціонує як лічильник підвищення.
2. Коли *CU* змінюється з FALSE на TRUE, лічильник виконує підрахунок один раз, і значення *CV* збільшується на 1. Коли *CV* дорівнює *PV* , *Q* має значення TRUE. Коли *Reset* встановлено на TRUE, *CV* очищається до 0, а *Q* скидається на FALSE.

- **Застереження щодо правильного використання**
  - Поки *Reset* має значення TRUE, лічильник не рахуватиметься.
  - Коли CV дорівнює PV, лічильник припиняє підрахунок.

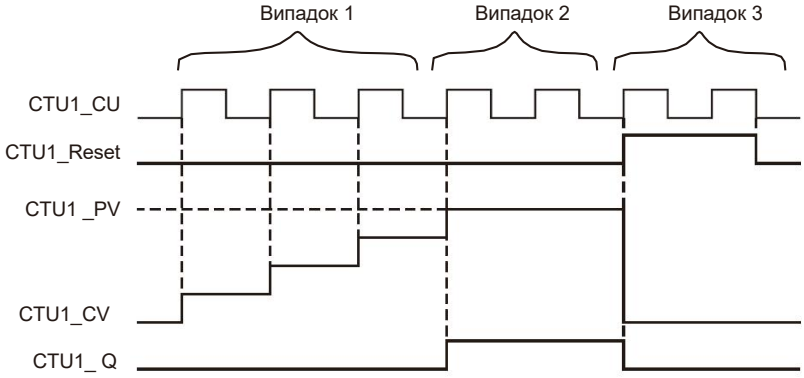
Приклад програмування

- **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
CTU1	CTU	
CTU1_EN	BOOL	FALSE
CTU1_CU	BOOL	FALSE
CTU1_Reset	BOOL	FALSE
CTU1_PV	UDINT	4
CTU1_Q	BOOL	
CTU1_CV	UDINT	



- **Графік часу:**



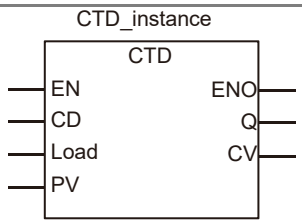
Випадок 1 : Якщо CTU підраховується нормально, значення CTU1_CV збільшується на 1 кожного разу, коли CTU1_CU запускається один раз.

Випадок 2 : Коли CTU1_CV дорівнює CTU1_PV, CTU1_Q має значення TRUE і CTU припиняє підрахунок.

Випадок 3 : Коли CTU1_Reset має значення TRUE, CTU1_CV скидається до 0, CTU1_Q має значення FALSE. І лічильник не буде рахуватися, коли спрацьовує CTU1_CU.

8.8.2 CTD

FB/FC	Пояснення	Застосовна модель
FB	CTD використовується як лічильник зниження.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CD	Вхідний сигнал нижнього лічильника	Вхід	Керуйте лічильником, щоб почати відлік	Правда чи брехня
Load	Сигнал навантаження	Вхід	Для запису значення лічильника	Правда чи брехня
PV	Попереднє значення	Вхід	Значення установки лічильника	0 ~ 4294967295
Q	Вихідний сигнал	Вихід	Q дорівнює ІСТИНІ, коли лічильник відлічує до 0.	Правда чи брехня
CV	Лічильник значення	Вихід	Лічильник теперішньої вартості	0 ~ 4294967295

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
Load	●																			
PV								●												
Q	●																			
CV								●												

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- CTU функціонує як лічильник зниження.
- Коли *Load* скидається на FALSE після встановлення значення TRUE, значення *PV* записується в *CV* . Коли *CD* змінюється з FALSE на TRUE, лічильник зменшує значення лічильника один раз, а значення *CV* зменшується на 1. Коли значення *CV* досягає 0, *Q* є TRUE.

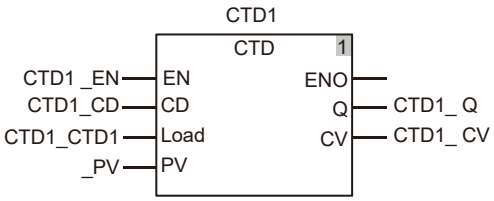
● **Застереження щодо правильного використання**

Поки Load має значення TRUE, лічильник не буде відлічувати.

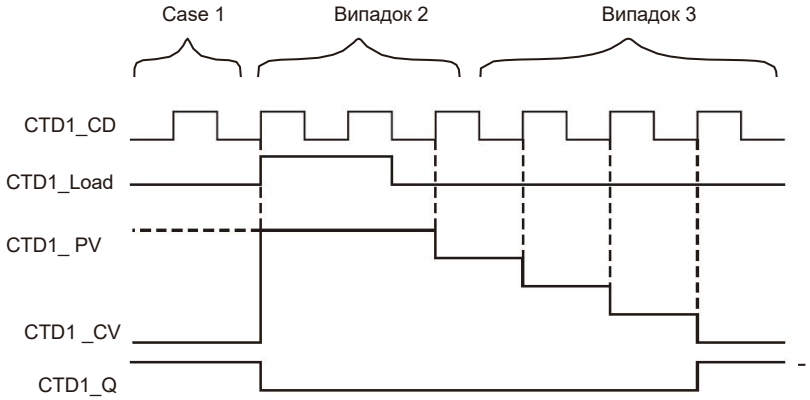
Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
CTD1	CTD	
CTD1_EN	BOOL	FALSE
CTD1_CD	BOOL	FALSE
CTD1_Load	BOOL	FALSE
CTD1_PV	UDINT	4
CTD1_Q	BOOL	
CTD1_CV	UDINT	



■ Графік часу:



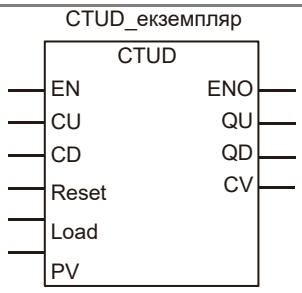
Випадок 1 : Запуск CTD1_CD, коли значення CTD1_CV дорівнює 0, не впливає на вихід.

Випадок 2 : Коли CTD1_Load має значення TRUE і CTD1_CV дорівнює заданому значенню CTD1_PV, CTD1_Q змінюється з TRUE на FALSE. На даний момент CTD1_CV не веде зворотний відлік, коли запускається CTD1_CD.

Випадок 3 : Якщо CTD веде зворотній відлік нормально, а CTD1_Load має значення FALSE, значення CTD1_CV зменшується на 1 кожного разу, коли CTD1_CD запускається один раз. CTD1_Q має значення TRUE, коли значення CTD1_CV зменшено до 0.

8.8.3 CTUD

FB/FC	Пояснення	Застосовна модель
FB	CTUD використовується як лічильник вгору-вниз.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CU	Вхідний сигнал лічильника	Вхід	Керуйте лічильником, щоб підрахувати	Правда чи брехня
CD	Вхідний сигнал нижнього лічильника	Вхід	Контролюйте лічильник для відліку	Правда чи брехня
Reset	Скинути сигнал	Вхід	Скинути поточне значення лічильника	Правда чи брехня
Load	Сигнал навантаження	Вхід	Для запису значення лічильника	Правда чи брехня
PV	Попереднє значення	Вхід	Значення установки лічильника	0 ~ 4294967295
QU	Вихідний сигнал	Вихід	Q є ІСТИННИМ, коли CV дорівнює PV.	Правда чи брехня
QD	Вихідний сигнал	Вихід	Q дорівнює ІСТИНІ, коли лічильник відлічує до 0.	Правда чи брехня
CV	Лічильник значення	Вихід	Лічильник теперішньої вартості	0 ~ 4294967295

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
CU	●																			
CD	●																			
Reset	●																			
Load	●																			
PV								●												
QU	●																			
Q.D.	●																			
CV								●												

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

CTUD використовується як зворотний лічильник для підрахунку вгору та зворотний лічильник для зворотного відліку.

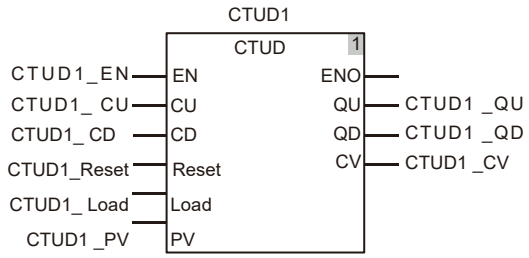
● **Застереження щодо правильного використання**

- ■ Лічильник не буде відраховувати, доки значення параметра « *Навантаження»* має значення «ІСТИНА».
- ■ Лічильник не буде підраховувати, доки *Reset* має значення TRUE.

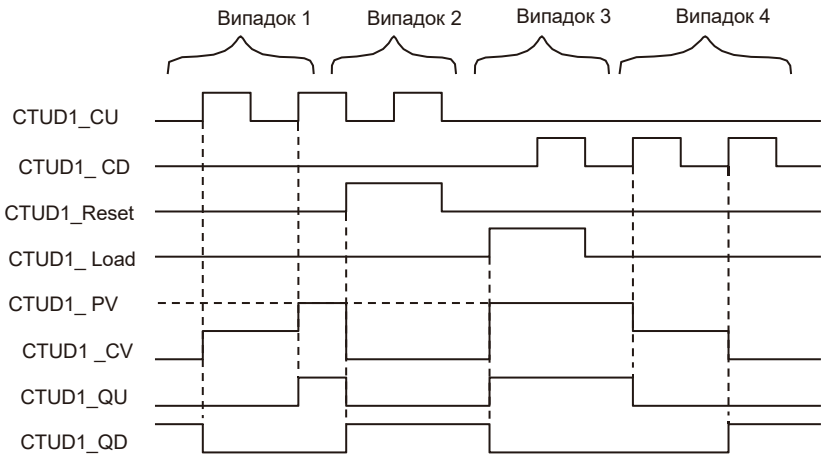
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
CTUD1	CTUD	
CTUD1_EN	BOOL	FALSE
CTUD1_CU	BOOL	FALSE
CTUD1_CD	BOOL	FALSE
CTUD1_Reset	BOOL	FALSE
CTUD1_Load	BOOL	FALSE
CTUD1_PV	UDINT	4
CTUD1_QU	BOOL	
CTUD1_QD	BOOL	
CTUD1_CV	UDINT	



■ **Графік часу:**



Випадок 1 : якщо CTUD підраховується нормально, значення CTUD1_CV збільшується на 1 кожного разу, коли CTUD1_CU запускається один раз.

Випадок 2 : коли CTUD1_Reset має значення TRUE, CTUD1_CV скидається на 0, CTUD1_QU змінюється на FALSE, а CTUD1_QD змінюється на TRUE.

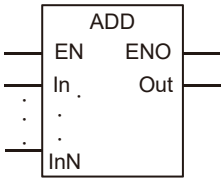
Випадок 3 : коли CTUD1_Load дорівнює TRUE, а CTUD1_CV дорівнює CTUD1_PV, CTUD1_QU змінюється на TRUE, а CTUD1_QD змінюється на FALSE. На даний момент, якщо CTUD1_CD спрацьовує, інструкція не може здійснювати зворотний відлік.

Випадок 4 : якщо інструкція веде зворотній відлік нормально, CTUD1_QU має значення FALSE, коли CTUD1_CD має значення TRUE. Значення CTUD1_CV зменшується на 1 кожного разу, коли CTUD1_CD запускається один раз. CTUD1_QD має значення TRUE, коли значення CTUD1_CV зменшено до 0.

8.9 Інструкції з математики

8.9.1 ADD

FB/FC	Пояснення	Застосовна модель
FC	ADD використовується для операції додавання двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Augend	Вхід	Augend	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2 до InN	Додавання	Вхід	Максимальна кількість доданків становить 7, що означає, що N може становити 2~8, і кількість можна збільшити або зменшити за допомогою програмного забезпечення для програмування під час створення програми.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	сума	Вихід	Результат додавання In1 до InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	
In2 до InN		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція використовується для додавання двох або більше змінних або констант. Результат виводиться в

Out , тобто *Out= In1 + In2 +…+ InN* .

■ Вхідні параметри *In1~InN* у цій інструкції можуть бути змінними різних типів серед бітів, цілих і дійсних чисел. Якщо *In1~InN* є змінними різних типів, операція додавання буде виконана на основі типу даних, який може містити всі допустимі діапазони значень *In1~InN* . Наприклад, тип даних *Out* — DINT, якщо тип даних *In1* — INT, а *In2* — DINT.



■ Вхідні та вихідні змінні можуть бути різних типів даних серед бітів, цілих чи дійсних чисел. Якщо типи даних вхідних і вихідних змінних різні, тип даних вихідної змінної повинен містити дійсні діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка. Наприклад, якщо типи даних *In1* і *In2* є INT і DINT відповідно, тип даних *Out* є DINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних змінної, до якої підключено *Out, є INT.* Під час компіляції програмного забезпечення не буде помилок, якщо тип даних змінної, до якої підключено *Out, — LINT.*

■ Для типу даних про час і дату підтримуються лише такі комбінації.

1. In1 – це TIME, In2 – це TIME, а Out – це TIME;
2. In1 це TOD (TIME_OF_DAY), In2 це TIME і Out це TOD;
3. In1 – DT (DAY_AND_TIME), In2 – TIME, а Out – DT.

● **Застереження щодо правильного використання**

■ Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

■ Сума *In1~InN* може бути поза допустимим діапазоном типу даних *Out.*

■ Різниця між *In1* і *In2* може бути поза допустимим діапазоном типу даних *Out.*

Наприклад, типи даних «ADD_In1» і «ADD_In2» мають значення INT із відповідними значеннями 32767 і 1. Якщо тип даних вихідної змінної INT, значення вихідної змінної буде -32768, як показано нижче. таблиця, змінна 1. Якщо тип даних вихідної змінної встановлено на DINT, значення вихідної змінної буде 32768, як показано в наступній таблиці, змінна 2.

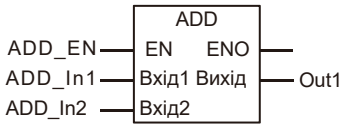
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
ADD_EN	BOOL	TRUE
ADD_In1	INT	32767
ADD_In2	INT	1
Out1	INT	-32768

➢ Змінна 2

Ім'я змінної	Тип даних	Поточне значення
ADD_EN	BOOL	TRUE
ADD_In1	INT	32767
ADD_In2	INT	1
Out1	DINT	32768

➢ Програма



● **Приклад програмування 1**

- типи даних змінних *ADD_In1, ADD_In2* і *Out1* є INT. Значення *ADD_In1* і *ADD_In2* дорівнюють 10 і 50 відповідно. Значення *Out1* становить 60, коли *ADD_EN* змінюється на TRUE, як показано у змінній 1.
- Усі типи даних змінних *ADD_In1, ADD_In2* і *Out1* є TIME. Значення *ADD_In1* і *ADD_In2* становлять TIME #1s і TIME #2s відповідно. Значення *Out1* становить TIME #3s, коли *ADD_EN* змінюється на TRUE, як показано у змінній 2.
- Типами даних змінних *ADD_In1, ADD_In2* і *Out1* є DT, TIME і DT відповідно. Значення *ADD_In1* і *ADD_In2* — DT#2016-9-1-8:00:00 і TIME#1H53M34S відповідно. Значення *Out1* становить DT#2016-09-01-09:53:34, коли *ADD_EN* змінюється на TRUE, як показано у змінній 3.

➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
ADD_EN	BOOL	TRUE
ADD_In1	INT	10
ADD_In2	INT	50
Out1	INT	60

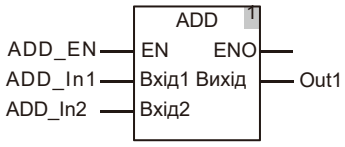
➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
ADD_EN	BOOL	TRUE
ADD_In1	TIME	TIME №1c
ADD_In2	TIME	TIME #2c
Out1	TIME	TIME #3c

➢ **Змінна 3**

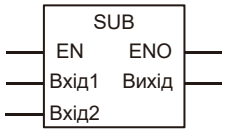
Ім'я змінної	Тип даних	Поточне значення
ADD_EN	BOOL	TRUE
ADD_In1	DT	DT#2016-9-1-8:00:00
ADD_In2	TIME	TIME#1H53M34S
Out1	DT	DT#2016-09-01-09:53:34

➢ **Програма**



8.9.2 SUB

FB/FC	Пояснення	Застосовна модель
FC	SUB використовується для операції віднімання двох змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Мінуенд	Вхід	Мінуенд	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
In2	Від'ємник	Вхід	Від'ємник	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
Out	Різниця	Вихід	Результат віднімання In1 та In2	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
In2		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	

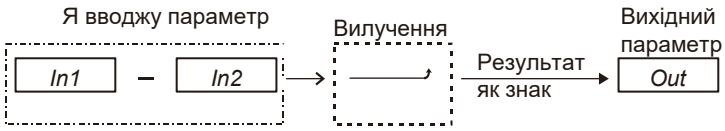
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

8

● **Пояснення функції**

- Ця інструкція використовується для віднімання двох або більше змінних або констант. Результат виводиться на *Out,* тобто *Out= In1 - In2* .
- Вхідні параметри *In1* та *In2* у цій інструкції можуть бути змінними різних типів даних серед бітів, цілих і дійсних чисел. Коли *In1* та *In2* є змінними різних типів, операція віднімання буде виконана на основі типу даних, який може містити допустимі діапазони значень *In1* та *In2* . Наприклад, тип даних *Out* — DINT, якщо тип даних *In1* — INT, а *In2* — DINT.



- Вхідні та вихідні змінні можуть бути різних типів даних серед бітів, цілих чи дійсних чисел. Якщо типи даних вхідних і вихідних змінних різні, тип даних вихідної змінної повинен містити дійсні діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка. Наприклад, якщо типи даних *In1* і *In2* є INT і DINT відповідно, тип даних *Out* є DINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних змінної, до якої підключено *Out, є INT.* Під час компіляції програмного забезпечення не буде помилок, якщо тип даних змінної, до якої підключено *Out, — LINT.*
- Для типу даних часу та дати підтримуються лише такі комбінації.

1. In1 – це TIME, In2 – це TIME, а Out – це TIME;
2. In1 це TOD, In2 це TIME і Out це TOD;
3. In1 це TOD, In2 це TOD і Out це TIME;
4. Вхід1 – DATE, Вхід2 – DATE, вихід – TIME;
5. In1 це DT, In2 це DT і Out це TIME;
6. In1 це DT, In2 це TIME і Out це DT.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Різниця між *In1* і *In2* може бути поза допустимим діапазоном типу даних *Out.*
  Наприклад, типи даних «SUB _In1» і «SUB _In2» мають значення INT із відповідними значеннями, -32768 і 1. Якщо тип даних вихідної змінної INT, значення вихідної змінної буде 32767, як показано в наступну таблицю, змінна 1. Якщо тип даних вихідної змінної встановлено на DINT, значення вихідної змінної буде -32769, як показано в наступній таблиці, змінна 2.

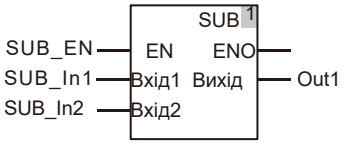
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
SUB_EN	BOOL	TRUE
SUB _In1	INT	-32768
SUB _In2	INT	1
Out1	INT	32767

➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
SUB_EN	BOOL	TRUE
SUB _In1	INT	-32768
SUB _In2	INT	1
Out1	DINT	-32769

➢ **Програма**



● **Приклад програмування**

- Усі типи даних змінних *SUB_In1, SUB _In2* і *Out1* мають значення INT, а значення *SUB _In1* і *SUB _In2* дорівнюють 100 і 40 відповідно. Значення *Out1* становить 60, коли *SUB_EN* змінюється на TRUE, як показано у змінній 1.

- Усі типи даних змінних *SUB_In1, SUB _In2* і *Out1* — це TIME, а значення *SUB _In1* і *SUB _In2* — TIME#4s і TIME#1s відповідно. Значення *Out1* становить TIME#3s, коли *SUB_EN* змінюється на TRUE, як показано у змінній 2.
- Типами даних змінних *SUB_In1, SUB _In2* і *Out1* є DATE, DATE і TIME, а значеннями *SUB _In1* і *SUB _In2* є DATE#2016-10-1 і DATE#2016-9-1 відповідно. Значення *Out1* становить TIME#30D, коли *SUB_EN* змінюється на TRUE, як показано у змінній 3.

➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
SUB_EN	BOOL	TRUE
SUB _In1	INT	100
SUB _In2	INT	40
Out1	INT	60

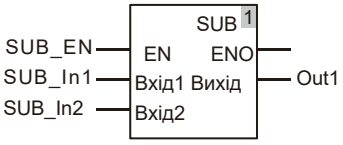
➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
SUB_EN	BOOL	TRUE
SUB_In1	TIME	TIME №4 с
SUB_In2	TIME	TIME №1 с
Out1	TIME	TIME №3 с

➢ **Змінна 3**

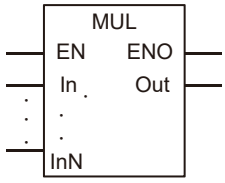
Ім'я змінної	Тип даних	Поточне значення
SUB_EN	BOOL	TRUE
SUB_In1	DATE	DATE №2016-10-1
SUB_In2	DATE	DATE №2016-9-1
Out1	TIME	TIME#30D

➢ **Програма**



8.9.3 MUL

FB/FC	Пояснення	Застосовна модель
FC	MUL використовується для множення двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Множене	Вхід	Множене	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2 до InN	Множник	Вхід	Максимальна кількість множників дорівнює 7, що означає, що N може становити 2–8, і це число можна збільшити або зменшити за допомогою програмного забезпечення для програмування під час створення програми.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Продукт	Вихід	Результат множення In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

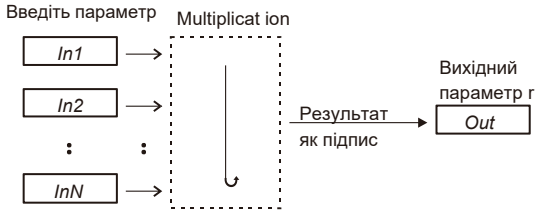
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Ця інструкція використовується для множення двох або більше змінних або констант. Результат виводиться на *Out,* тобто *Out*= In1 * In2 * … * InN.

- вхідні параметри *In1 ~ InN* можуть бути змінними різних типів даних. Якщо *In1 ~ InN* є змінними різних типів даних, множення буде виконано на основі типу даних, який може містити дійсні діапазони значень *In1 ~ InN* . Наприклад, тип даних *Out* — DINT, якщо тип даних *In1* — INT, а *In2* — DINT.



- У цій інструкції вхідні та вихідні змінні можуть мати різні типи даних. Якщо типи даних вхідних і вихідних змінних різні, діапазон типу даних вихідної змінної повинен включати допустимі діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка. Наприклад, якщо типи даних *In1* і *In2* є INT і DINT відповідно, тип даних *Out* є DINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних змінної, до якої підключено *Out, є INT.* Під час компіляції програмного забезпечення не виникне жодної помилки, якщо тип даних змінної, до якої підключено *Out , — LINT.*

● Застереження щодо правильного використання

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Результат множення *In1 ~ In2* може бути поза допустимим діапазоном типу даних *Out.* Наприклад, типи даних «MUL _In1» і «MUL _In2» мають значення INT із відповідними значеннями 20000 і 2. Якщо тип даних вихідної змінної INT, значення вихідної змінної буде -25536, як показано в наступну таблицю, змінна 1. Якщо тип даних вихідної змінної встановлено на DINT, значення вихідної змінної буде 40000, як показано в наступній таблиці, змінна 2.

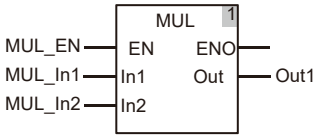
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
MUL_EN	BOOL	TRUE
MUL _In1	INT	20000
MUL _In2	INT	2
Out1	INT	-25536

➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
MUL_EN	BOOL	TRUE
MUL _In1	INT	20000
MUL _In2	INT	2
Out1	DINT	40000

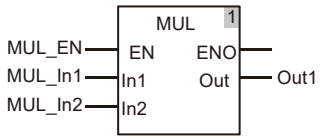
➢ **Програма**



● Приклад програмування

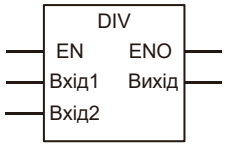
- типи даних змінних *MUL _In1, MUL _In2* і *Out1* є INT. Значення *MUL _In1* і *MUL _In2* дорівнюють 10 і 50 відповідно. Значення *Out1* становить 500, коли *MUL _EN* змінюється на TRUE. **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
MUL_EN	BOOL	TRUE
MUL _In1	INT	10
MUL _In2	INT	50
Out1	INT	500



8.9.4 DIV

FB/FC	Пояснення	Застосовна модель
FC	DIV використовується для операції ділення двох змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Дивіденд	Вхід	Дивіденд	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Дільник	Вхід	Дільник	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. 0 виключається.
Out	Коефіцієнт	Вихід	Результат ділення In1 і In2	Залежить від типу даних змінної, до якої підключено вихідний параметр.

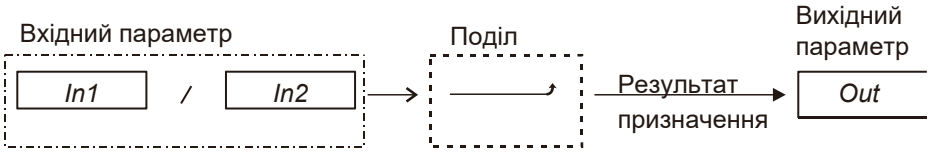
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
In2		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- Інструкція використовується для поділу двох змінних або констант. Результат виводиться на *Out,*
  тобто *Out*= In1 / In2.
- вхідні параметри *In1* та *In2* можуть бути змінними різних типів даних. Коли *In1* та *In2* є змінними різних типів даних, поділ буде виконано на основі типу даних, який може містити дійсні діапазони *In1* та *In2* . Наприклад, тип даних *Out* — DINT, якщо тип даних *In1* — INT, а *In2* — DINT.



- У цій інструкції вхідні та вихідні змінні можуть мати різні типи даних. Якщо типи даних вхідних і вихідних змінних різні, діапазон типу даних вихідної змінної повинен включати допустимі діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка. Наприклад, якщо типи даних *In1* і *In2* є INT і DINT відповідно, тип даних *Out* є DINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних *Out* – INT. Під час компіляції програмного забезпечення не буде помилок, якщо тип даних *Out* – LINT.

Застереження щодо правильного використання

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Вхідне значення *In2* не може бути 0. Іншими словами, дільник в операції ділення не може бути 0. Значення *Out* буде 0, якщо значення *In2* дорівнює 0.
- Результат ділення *In1* та *In2* може бути поза допустимим діапазоном типу даних *Out.*
  Наприклад, типи даних «DIV _In1» і «DIV _In2» мають значення INT разом із відповідними значеннями, -32768 і -1. Якщо тип даних вихідної змінної – INT, значення вихідної змінної буде -32768, як показано в наступній таблиці, змінна 1. Якщо тип даних вихідної змінної встановлено на DINT, значення вихідної змінної буде 32768 як показано в наступній таблиці, змінна 2.

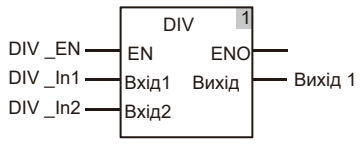
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
DIV_EN	BOOL	TRUE
DIV_In1	INT	-32768
DIV_In2	INT	-1
Out1	INT	-32768

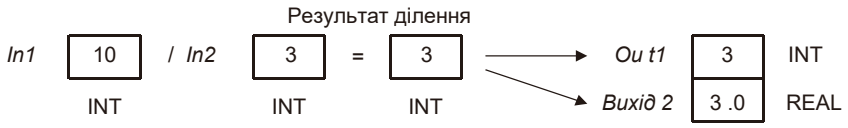
➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
DIV_EN	BOOL	TRUE
DIV_In1	INT	-32768
DIV_In2	INT	-1
Out1	DINT	32768

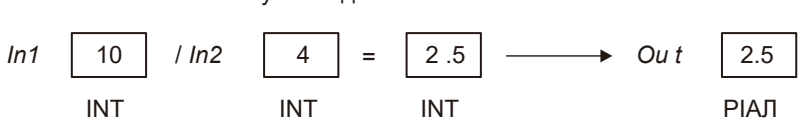
➢ **Програма**



- Результатом завжди є ціле число для ділення двох цілих чисел. Навіть якщо є залишок від ділення двох цілих чисел, залишок вирізається.
  Наприклад, типи даних *In1* та *In2* є INT із відповідними значеннями 10 і 3. А тип даних *Out* є INT і Real, тому його значення дорівнює 3 і 3,0 відповідно, як показано на наступному малюнку.



Тип даних *Out* — це дійсне число для ділення цілого і дійсного чисел або ділення двох дійсних чисел. Значення *Out* показано нижче, включаючи його дробову частину, якщо є залишок для цього типу ділення.

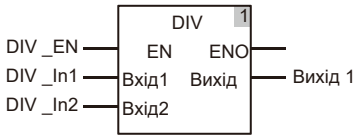


- **Приклад програмування**

  - типи даних змінних *DIV _In1, DIV _In2* і *Out1* є INT. Значення *DIV _In1* і *DIV _In2* дорівнює 100 і 20 відповідно. Значення *Out1* дорівнює 5, коли *DIV _EN* змінюється на TRUE.

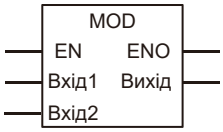
Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
DIV_EN	BOOL	TRUE
DIV_In1	INT	100
DIV_In2	INT	20
Out1	INT	5



8.9.5 MOD

FB/FC	Пояснення	Застосовна модель
FC	MOD знаходить залишок від ділення двох цілих змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Дивіденд	Вхід	Дивіденд	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Дільник	Вхід	Дільник	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. 0 виключається.
Out	Залишок	Вихід	Залишок, отриманий діленням In1 на In2	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1		●	●	●	●	●	●	●	●	●	●	●	●							
In2		●	●	●	●	●	●	●	●	●	●	●	●							
Out		●	●	●	●	●	●	●	●	●	●	●	●							

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

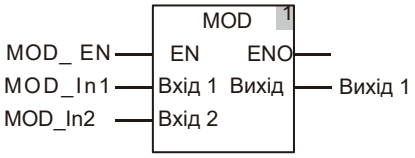
● **Пояснення функції**

- Ця інструкція використовується для отримання залишку від ділення двох цілих змінних або констант. Результат виводиться на *Out,* тобто *Out*= In1 – (In1/ In2)*In2.
- У цій інструкції вхідна змінна та вхідна змінна або вхідна змінна та вихідна змінна можуть мати різні типи даних. Якщо типи даних вхідних і вихідних змінних різні, тип даних вихідної змінної повинен містити дійсні діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка. Наприклад, якщо типи даних *In1* і *In2* є INT і DINT відповідно, тип даних *Out* є DINT. Буде

помилка під час компіляції програмного забезпечення, якщо тип даних *Out* є INT. Під час компіляції програмного забезпечення не буде помилок, якщо тип даних *Out* – LINT.

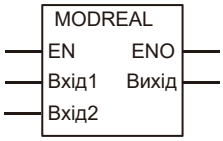
- **Застереження щодо правильного використання**
  - Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
  - Вхідне значення *In2* не може бути 0. Іншими словами, дільник в операції ділення не може бути 0. Значення *Out* буде 0, якщо значення *In2* дорівнює 0.
- **Приклад програмування**
  - типи даних змінних *MOD _In1, MOD _In2* і *Out1* є INT. Значення *MOD _In1* і *MOD _In2* дорівнюють 10 і 4 відповідно. Значення *Out1* дорівнює 2, коли *MOD _EN* змінюється на TRUE.
    Змінна і програма

Ім'я змінної	Тип даних	Поточне значення
MOD_EN	BOOL	TRUE
MOD _In1	INT	10
MOD _In2	INT	4
Out1	INT	2



8.9.6 MODREAL

FB/FC	Пояснення	Застосовна модель
FC	MODREAL знаходить залишок від ділення двох змінних або констант із плаваючою комою.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Дивіденд	Вхід	Дивіденд	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Дільник	Вхід	Дільник	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. 0 виключається.
Out	Залишок	Вихід	Залишок, отриманий діленням In1 на In2	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out														●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- Ця інструкція використовується для знаходження залишку від ділення двох змінних або констант із плаваючою комою, а результат виводиться в *Out* .
- У цій інструкції вхідна змінна та вхідна змінна або вхідна змінна та вихідна змінна можуть мати різні типи даних.

● **Застереження щодо правильного використання**

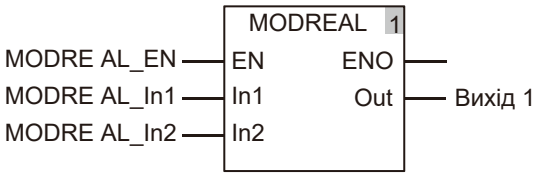
- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

- Вхідне значення *In2* не може бути 0. Іншими словами, дільник в операції ділення не може бути 0. Значення *Out* буде 0, якщо значення *In2* дорівнює 0.

● **Приклад програмування**

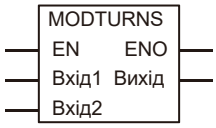
- Типи даних змінних *MODREAL _In1, MODREAL _In2* і *Out1* є REAL, REAL і LREAL відповідно. Значення *MODREAL _In1* і *MOD _In2* становлять 10,5 і 2,5 відповідно. Значення *Out1* становить 0,5, коли *MODREAL _EN* змінюється на TRUE.
  Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
MODREAL_EN	BOOL	TRUE
MODREAL _In1	REAL	10.5
MODREAL _In2	REAL	2.5
Out1	LREAL	0,5



8.9.7 MODTURNS

FB/FC	Пояснення	Застосовна модель
FC	MODTURN знаходить інтегральну частину зі знаком для ділення за модулем двох змінних або констант із плаваючою комою.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Вхідне значення	Вхід	Вхідне значення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Модульний діапазон	Вхід	Модульний діапазон	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. 0 виключається.
Out	Кількість поворотів за модулем	Вихід	Кількість поворотів за модулем	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out												●								

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- MODTURN використовується для виконання ділення за модулем двох змінних чи констант із плаваючою комою та отримання інтегрального компонента зі знаком. Результат виводиться в *Out* . Кількість поворотів осі за модулем можна розрахувати відповідно до її встановленого абсолютного положення.
- У цій інструкції вхідна змінна та вхідна змінна або вхідна змінна та вихідна змінна можуть мати різні типи даних.

● **Застереження щодо правильного використання**

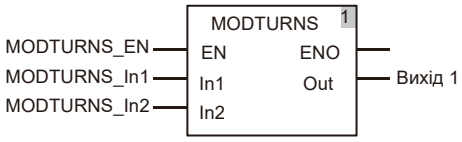
- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Вхідне значення *In2* не може бути 0. Іншими словами, дільник в операції ділення не може бути 0. Значення *Out* буде 0, якщо значення *In2* дорівнює 0.

● **Приклад програмування**

- Типи даних змінних *MODTURNS _In1, MODTURNS _In2 є* REAL, а *Out1* – DINT. Значення *MODTURNS _In1* і *MODTURNS _In2* становлять 800,23 і 360,0 відповідно. Значення *Out1* дорівнює 2, коли *MODTURNS _EN* змінюється на TRUE.

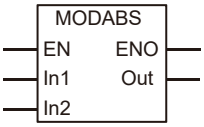
Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
MODTURNS_EN	BOOL	TRUE
MODTURNS _In1	REAL	800,23
MODTURNS _In2	REAL	360,0
Out1	DINT	2



8.9.8 MODABS

FB/FC	Пояснення	Застосовна модель
FC	MODABS знаходить беззнакове модульне значення для модульного ділення двох змінних чи констант із плаваючою комою.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Вхідне значення	Вхід	Вхідне значення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Модульний діапазон	Вхід	Модульний діапазон	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. 0 виключається.
Out	Значення по модулю	Вихід	Значення по модулю	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1														●	●					
In2														●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- MODABS використовується для виконання модульного ділення двох змінних чи констант із плаваючою комою та отримання беззнакового значення за модулем. Результат виводиться в *Out* . Позицію за модулем можна обчислити згідно з абсолютним положенням осі.
- У цій інструкції вхідна змінна та вхідна змінна або вхідна змінна та вихідна змінна можуть мати різні типи даних.

- **Застереження щодо правильного використання**
  - Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
  - Вхідне значення *In2* не може бути 0. Іншими словами, дільник в операції ділення не може бути 0. Значення *Out* буде 0, якщо значення *In2* дорівнює 0.
- **Приклад програмування**
  - типи даних змінних *MODABS _In1* і *MODABS _In2* є REAL, а тип даних *Out1* – LREAL. Значення *MODABS _In1* і *MODABS _In2* становлять 400,23 і 360,0 відповідно. Значення *Out1* становить 40,2300109863281, коли *MODABS _EN* змінюється на TRUE. Значення *MODABS _In1* і *MODABS _In2* становлять -400,23 і 360,0 відповідно. Значення *Out1* становить 319,769989013672, коли *MODABS _EN* змінюється на TRUE.

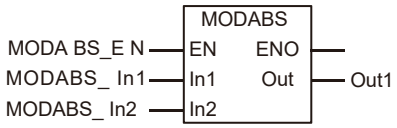
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
MODABS_EN	BOOL	TRUE
MODABS _In1	REAL	400,23
MODABS _In2	REAL	360,0
Out1	LREAL	40,2300109863281

➢ Змінна 2

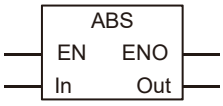
Ім'я змінної	Тип даних	Поточне значення
MODABS_EN	BOOL	TRUE
MODABS _In1	REAL	-400,23
MODABS _In2	REAL	360,0
Out1	LREAL	319,769989013672

➢ Програма



8.9.9 ABS

FB/FC	Пояснення	Застосовна модель
FC	ABS знаходить абсолютне значення цілого чи дійсного числа.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Абсолютне значення	Вихід	Абсолютне значення *In*	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ ABS знаходить абсолютне значення вхідного параметра *In* .
Результат виводиться в *Out* . Тобто Out = | У |.

■ У цій інструкції вхідна та вихідна змінні можуть мати різні типи даних. Якщо типи даних вхідних і вихідних змінних різні, діапазон типу даних вихідної змінної повинен включати допустимі діапазони типів даних усіх вхідних змінних. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

■ Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

● **Приклад програмування**

■ типи даних змінних *ABS _In* та *Out1* є INT, а значення *ABS _In* дорівнює -10. Значення *Out1* дорівнює 10, коли *ABS _EN* змінюється на TRUE. Значення *Out1* дорівнює 20, а *ABS_In* дорівнює 20.

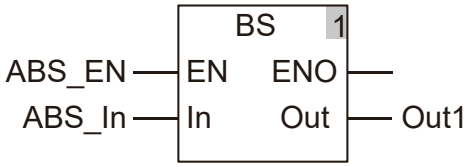
- **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
ABS_EN	BOOL	TRUE
ABS _In	INT	-10
Out1	INT	10

- **Змінна 2**

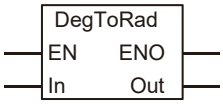
Ім'я змінної	Тип даних	Поточне значення
ABS_EN	BOOL	TRUE
ABS _In	INT	20
Out1	INT	20

- **Програма**



8.9.10 DegToRad

FB/FC	Пояснення	Застосовна модель
FC	DegToRad використовується для перетворення градусів у радіани.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Ступені	Вхід	Градуси для конвертації	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Радіани	Вихід	Радіани, конвертовані в градуси	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- DegToRad використовується для перетворення вхідного параметра *In* у радіан, а результат виводиться у *Out* . Тобто Out =(In/180)* π.
- Одиницями вимірювання *In* і *Out* є градус (°) і радіан відповідно.
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідні змінні дозволено опускати.

● **Приклад програмування**

- Типи даних *DegToRad _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* становить 0,174532925199433, якщо значення *DegToRad _In* дорівнює 10, коли *DegToRad _EN* змінюється на TRUE. Значення *Out1* становить -0,174532925199433, оскільки *DegToRad _In* дорівнює -10.

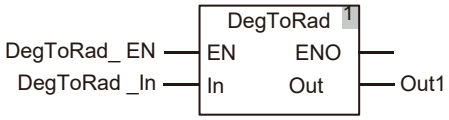
- **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
DegToRad_EN	BOOL	TRUE
DegToRad _In	INT	10
Out1	LREAL	0,174532925199433

- **Змінна 2**

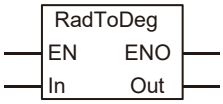
Ім'я змінної	Тип даних	Поточне значення
DegToRad_EN	BOOL	TRUE
DegToRad _In	INT	-10
Out1	LREAL	-0,174532925199433

- **Програма**



8.9.11 RadToDeg

FB/FC	Пояснення	Застосовна модель
FC	DegToRad використовується для перетворення радіанів у градуси.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Радіани	Вхід	Радіани для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Ступені	Вихід	Градуси, перетворені з радіан	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- RadToDeg використовується для перетворення вхідного параметра *In* у градуси, а результат виводиться у *Out* . Тобто Out =(In/π)* 180.
- Одиницями *In* та *Out* є радіан і градус (°) відповідно.
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

● **Приклад програмування**

■ Типами даних змінних *RadToDeg _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* становить 572. 957795130824, якщо значення *RadToDeg _In* дорівнює 10, коли *RadToDeg _EN* змінюється на TRUE. Значення *Out1* дорівнює -572. 957795130824, оскільки *RadToDeg _In* дорівнює -10.

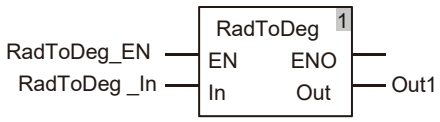
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
RadToDeg _EN	BOOL	TRUE
RadToDeg _In	INT	10
Out1	LREAL	572. 957795130824

➢ **Змінна 2**

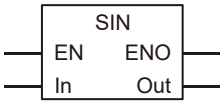
Ім'я змінної	Тип даних	Поточне значення
RadToDeg_EN	BOOL	TRUE
RadToDeg_In	INT	-10
Out1	LREAL	-572. 957795130824

➢ **Програма**



8.9.12 SIN

FB/FC	Пояснення	Застосовна модель
FC	SIN використовується для знаходження синуса числа, а результат виводиться в *Out* . Одиницею *In* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Радіани для обробки	Вхід	Радіани для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	-1,00000000000000 ~ 1,00000000000000

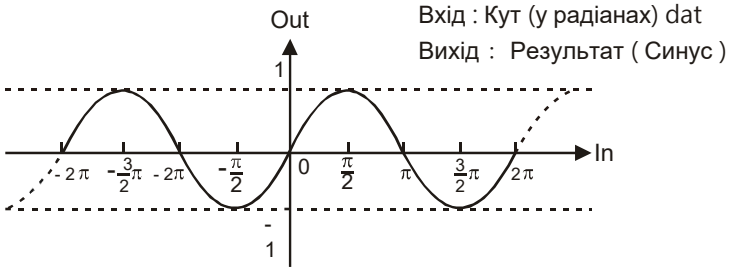
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BxOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ SIN використовується для обчислення синуса вхідного параметра *In* , а результат виводиться в *Out* .



■ Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

- **Застереження щодо правильного використання**

  - Налаштування вхідної змінної не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яке налаштування вхідної змінної пропущено. Але налаштування вихідної змінної дозволено опускати.

- **Приклад програмування**

  - Типами даних змінних *SIN _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* становить -0,54402111088937, якщо значення *SIN _In* дорівнює 10, коли *SIN _EN* змінюється на TRUE. Значення *Out1* становить 0,54402111088937, оскільки *SIN _In* дорівнює -10.

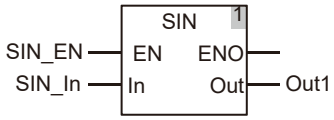
    ➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
SIN_EN	BOOL	TRUE
SIN_In	INT	10
Out1	LREAL	-0,54402111088937

    ➢ Змінна 2

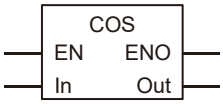
Ім'я змінної	Тип даних	Поточне значення
SIN_EN	BOOL	TRUE
SIN_In	INT	-10
Out1	LREAL	0,54402111088937

    ➢ Програма



8.9.13 COS

FB/FC	Пояснення	Застосовна модель
FC	COS використовується для отримання косинуса числа, а результат виводиться в *Out* . Одиницею *In* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Радіани для обробки	Вхід	Радіани для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	-1,00000000000000 ~ 1,00000000000000

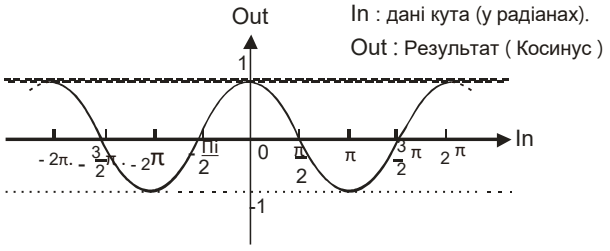
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ COS використовується для обчислення косинуса вхідного параметра *In* , а результат виводиться в *Out* .



■ Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

- **Застереження щодо правильного використання**

  - Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

- **Приклад програмування**

  - Типами даних змінних *COS _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* становить -0,839071529076452, якщо значення *COS _In* дорівнює 10, коли *COS _EN* змінюється на TRUE. Значення *Out1* становить -0,839071529076452, оскільки *COS _In* дорівнює -10.

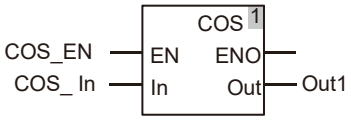
    ➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
COS_EN	BOOL	TRUE
COS_In	INT	10
Out1	LREAL	-0,839071529076452

    ➢ Змінна 2

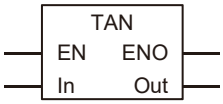
Ім'я змінної	Тип даних	Поточне значення
COS_EN	BOOL	TRUE
COS_In	INT	-10
Out1	LREAL	-0,839071529076452

    ➢ Програма



8.9.14 TAN

FB/FC	Пояснення	Застосовна модель
FC	TAN використовується для отримання тангенса числа, а результат виводиться в *Out* . Одиницею *In* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Радіани для обробки	Вхід	Радіани для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	Залежить від типу даних змінної, до якої підключено вихідний параметр.

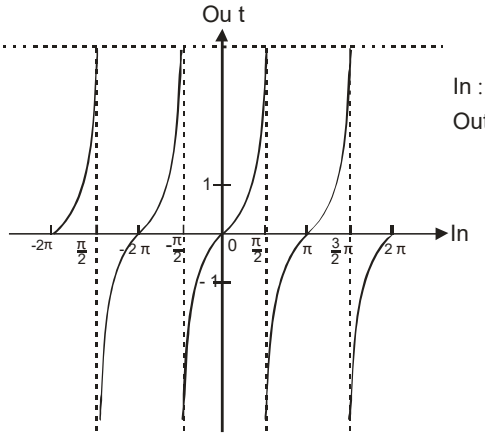
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ TAN використовується для обчислення тангенса вхідного параметра *In* , а результат виводиться в *Out* .



In : Дані кута (у радіанах).
Out : результат (тангенс)

■ Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

■ Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

● **Приклад програмування**

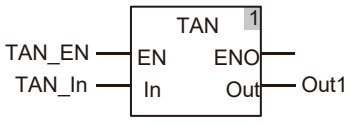
■ Типами даних змінних *TAN _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* становить 0,648360827459087, якщо значення *TAN _In* дорівнює 10, коли *TAN _EN* змінюється на TRUE. Значення *Out1* дорівнює -0,648360827459087, а *TAN _In* дорівнює -10.

Ім'я змінної	Тип даних	Поточне значення
TAN_EN	BOOL	TRUE
TAN_In	INT	10
Out1	LREAL	0,648360827459087

➢ Змінна 2

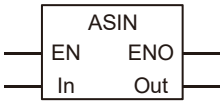
Ім'я змінної	Тип даних	Поточне значення
TAN_EN	BOOL	TRUE
TAN_In	INT	-10
Out1	LREAL	-0,648360827459087

➢ Програма



8.9.15 ASIN

FB/FC	Пояснення	Застосовна модель
FC	ASIN використовується для отримання арксинуса числа, а результат виводиться в *Out* . Одиницею *Out* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	-π/2 ~ π/2

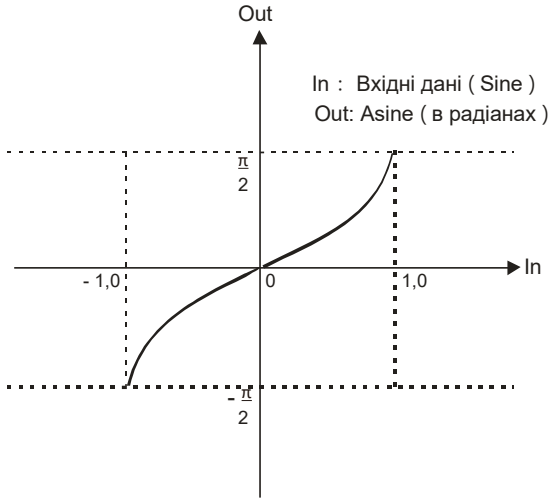
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ ASIN використовується для обчислення арксинуса вхідного параметра *In* , а результат виводиться в *Out* .

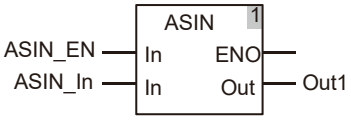


- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Значення *Out* змінюється від -π/2 до π/2, коли значення *In* змінюється від -1,0 до 1,0. Інструкція не перейде в стан помилки, якщо значення *In* виходить за межі -1,0 ~1,0, а значення *Out* є нечисловим, як показано в наступній таблиці та програмі.

Ім'я змінної	Тип даних	Поточне значення
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	2.0
Out1	LREAL	1. #QNAN



● **Приклад програмування**

- Типи даних змінних *ASIN_In* і *Out1* є REAL і LREAL відповідно. Значення *Out1* становить 1,5707963267949, якщо значення *ASIN_In* дорівнює 1,0, коли *ASIN_EN* змінюється на TRUE. Значення *Out1* дорівнює -1,5707963267949, а *ASIN_In* дорівнює -1,0.

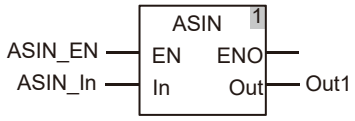
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	1.0
Out1	LREAL	1,5707963267949

➢ Змінна 2

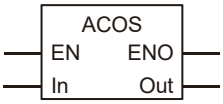
Ім'я змінної	Тип даних	Поточне значення
ASIN_EN	BOOL	TRUE
ASIN_In	REAL	-1,0
Out1	LREAL	-1,5707963267949

- Програма



8.9.16 ACOS

FB/FC	Пояснення	Застосовна модель
FC	ACOS використовується для отримання арккосинуса числа, а результат виводиться в *Out* Одиницею *Out* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	0 ~ π

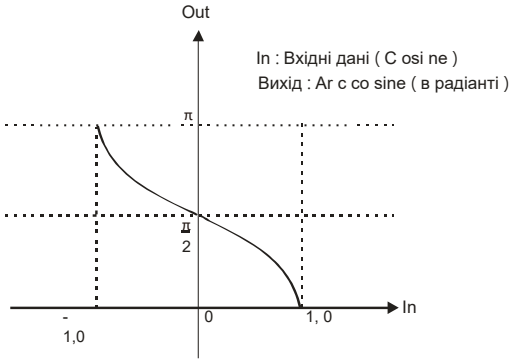
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ ACOS використовується для обчислення арккосинуса вхідного параметра *In* , а результат виводиться в *Out* .



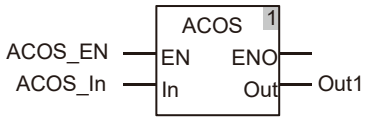
■ Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- ■ Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- ■ Значення *Out* змінюється від 0 до π, коли значення *In* змінюється від -1,0 до 1,0. Інструкція не перейде в стан помилки, якщо значення *In* виходить за межі -1,0 ~1,0, а значення *Out* є нечисловим.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	2.0
Out1	LREAL	1.#QNAN



● **Приклад програмування**

- ■ Типи даних змінних *ACOS_In* і *Out1* є REAL і LREAL відповідно. Значення *Out1* дорівнює 0, якщо значення *ACOS_In* дорівнює 1,0, коли *ACOS_EN* змінюється на TRUE. Значення *Out1* становить 3,14159265358979, оскільки *ACOS_In* дорівнює -1,0.

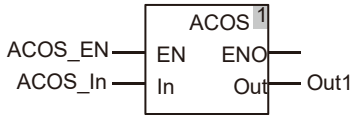
➢ змінна

Ім'я змінної	Тип даних	Поточне значення
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	1.0
Out1	LREAL	0

➢ змінна

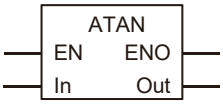
Ім'я змінної	Тип даних	Поточне значення
ACOS_EN	BOOL	TRUE
ACOS_In	REAL	-1,0
Out1	LREAL	3,14159265358979

➢ Програма



8.9.17 ATAN

FB/FC	Пояснення	Застосовна модель
FC	ATAN використовується для знаходження арктангенса числа, а результат виводиться в *Out.* Одиницею *Out* є радіан.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції	-π/2 ~ π/2

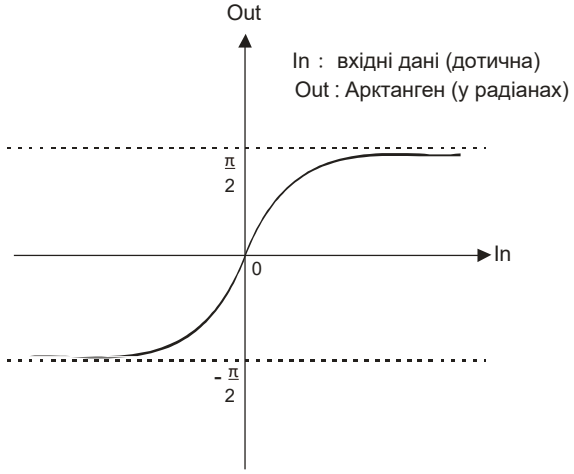
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ ATAN використовується для обчислення арктангенса вхідного параметра *In* , а результат виводиться в *Out* .



- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Вихідне значення *Out* дорівнює -π/2, якщо вхідне значення *In* дорівнює $-\infty$.
  Вихідне значення *Out* дорівнює π/2, якщо вхідне значення *In* дорівнює $+\infty$.

● **Приклад програмування**

- Типи даних змінних *ATAN_In* і *Out1* є REAL і LREAL відповідно. Значення *Out1* становить 0,785398163397448, якщо значення *ATAN_In* дорівнює 1,0, коли *ATAN_EN* змінюється на TRUE. Значення *Out1* становить -0,785398163397448, а *ATAN_In* - -1,0.

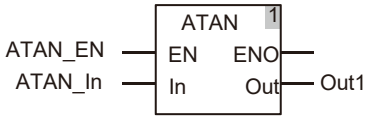
  ➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
ATAN_EN	BOOL	TRUE
ATAN_In	REAL	1.0
Out1	LREAL	0,785398163397448

  ➢ Змінна 2

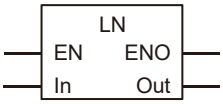
Ім'я змінної	Тип даних	Поточне значення
ATAN_EN	BOOL	TRUE
ATAN_In	REAL	-1,0
Out1	LREAL	-0,785398163397448

  ➢ Програма



8.9.18 LN

FB/FC	Пояснення	Застосовна модель
FC	LN використовується для знаходження натурального логарифма числа, а результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Логарифм	Вихід	Натуральний логарифм *In*	Залежить від типу даних змінної, до якої підключено вихідний параметр.

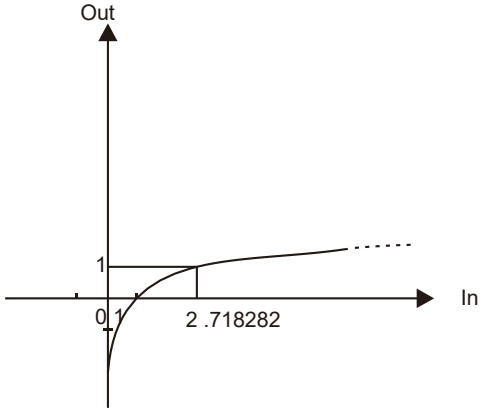
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ LN використовується для обчислення натурального логарифма вхідного параметра *In,* тобто логарифма з e （ e=2,718282 ） як основу , а результат виводиться на *вихід* .

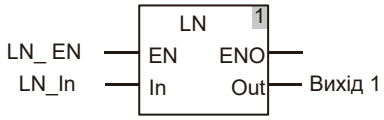


- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Вихідне значення *Out* є нечисловим, якщо вхідне значення *In* є недодатним числом, як показано в наступній таблиці.
  Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
LN_EN	BOOL	TRUE
LN_In	REAL	-2,0
Out1	LREAL	1.#QNAN



● **Приклад програмування**

- Типами даних змінних *LN_In* і *Out1* є INT і LREAL відповідно. Значення *Out1* є 0,0, якщо значення *LN_In* дорівнює 1, коли *LN_EN* змінюється на TRUE. Значення *Out1* дорівнює 1,00000005734143, а *LN_In* — 2,718282.

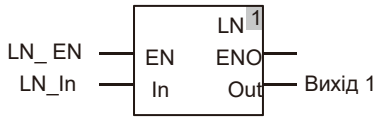
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
LN_EN	BOOL	TRUE
LN_In	INT	1
Out1	LREAL	0,0

➢ Змінна 2

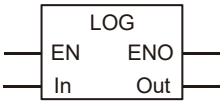
Ім'я змінної	Тип даних	Поточне значення
LN_EN	BOOL	TRUE
LN_In	REAL	2,718282
Out1	LREAL	1,00000005734143

➢ Програма



8.9.19 LOG

FB/FC	Пояснення	Застосовна модель
FC	LOG використовується для знаходження логарифма числа за основою 10, а результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Логарифм	Вихід	Логарифм з основою 10	Залежить від типу даних змінної, до якої підключено вихідний параметр.

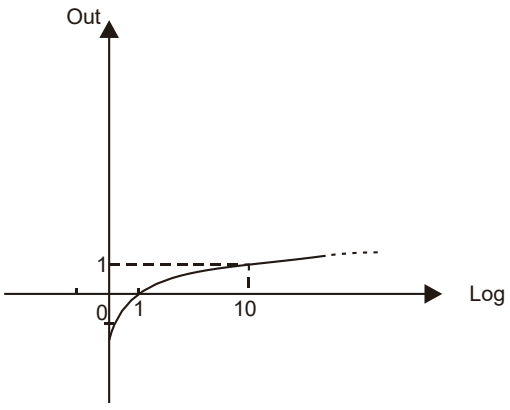
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ LOG використовується для обчислення логарифма за основою 10 вхідного параметра *In* , а результат виводиться в*Out*



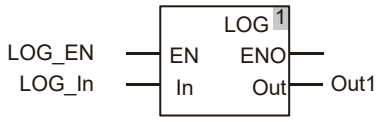
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежений LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Вихідне значення *Out* є нечисловим значенням, якщо вхідне значення *In* є недодатним числом, як показано в наступній таблиці.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
LOG_EN	BOOL	TRUE
LOG_In	REAL	-2,0
Out1	LREAL	1.#QNAN



● **Приклад програмування**

- Типами даних змінних *LOG_In* і *Out1* є INT і LREAL відповідно. Значення *Out1* дорівнює 0,0, якщо значення *LOG_In* дорівнює 1, коли LOG_EN змінюється на TRUE. Значення *Out1* дорівнює 1,0, а *LOG_In* дорівнює 10.

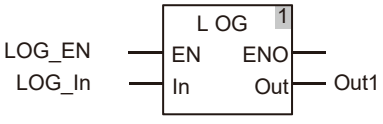
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
LOG_EN	BOOL	TRUE
LOG_In	INT	1
Out1	LREAL	0,0

➢ Змінна 2

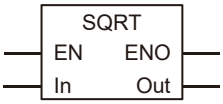
Ім'я змінної	Тип даних	Поточне значення
LOG_EN	BOOL	TRUE
LOG_In	INT	10
Out1	LREAL	1.0

➢ Програма



8.9.20 SQRT

FB/FC	Пояснення	Застосовна модель
FC	SQRT використовується для обчислення квадратного кореня з числа, а результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Номер для обробки	Вхід	Номер для обробки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр. І це невід'ємне число.
Out	Квадратний корінь	Вихід	Квадратний корінь	Залежить від типу даних змінної, до якої підключено вихідний параметр. І це невід'ємне число.

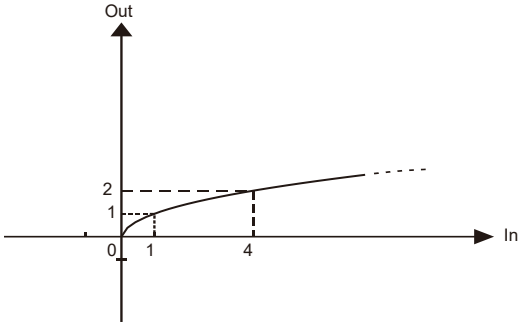
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ SQRT використовується для обчислення квадратного кореня з *In* , а результат виводиться в *Out* .



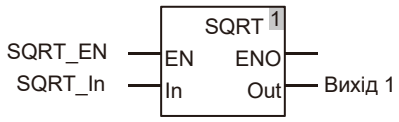
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Вихідне значення *Out* є нечисловим значенням, якщо вхідне значення *In* є від'ємним числом.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
SQRT_EN	BOOL	TRUE
SQRT_In	REAL	-2,0
Out1	LREAL	1.#QNAN



● **Приклад програмування**

- Типами даних змінних *SQRT _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* дорівнює 4,0, якщо значення *SQRT _In* дорівнює 16, коли *SQRT _EN* змінюється на TRUE. Значення *Out1* є 10,0, оскільки *SQRT_In* дорівнює 100.

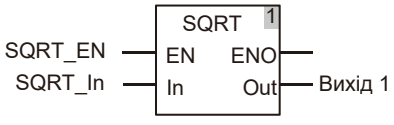
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
SQRT_EN	BOOL	TRUE
SQRT_In	INT	16
Out1	LREAL	4.0

➢ Змінна 2

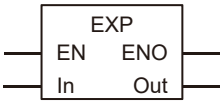
Ім'я змінної	Тип даних	Поточне значення
SQRT_EN	BOOL	TRUE
SQRT_In	INT	100
Out1	LREAL	10,0

➢ Програма



8.9.21 EXP

FB/FC	Пояснення	Застосовна модель
FC	EXP використовується для виконання операції з e як базове число та *In* як експонента. Результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Експонента	Вхід	Експонента	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції з основою e і показником *In*	Залежить від типу даних змінної, до якої підключено вихідний параметр. І це невід'ємне число.

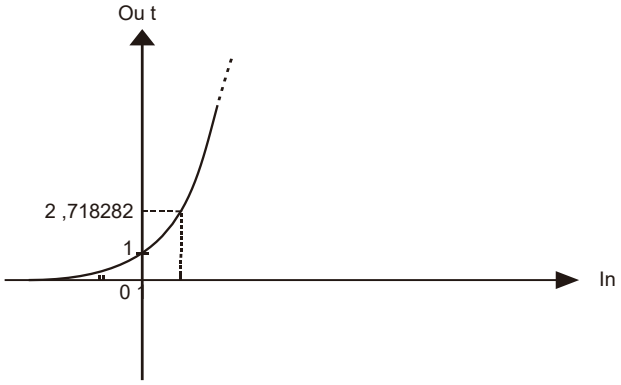
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ EXP використовується для виконання операції з e （ e=2,718282 ） як базове число та *In* як експонента. Результат виводиться в *Out* .



- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.
- Коли значення *In* дорівнює 0, + ∞ , - ∞ і нечислове значення, відповідні вихідні значення *Out* перераховано в наступній таблиці.

In	Out
0	1.0
+ ∞	+ ∞
- ∞	0,0
нечисловий	нечисловий

● **Приклад програмування**

- Типи даних *EXP _In* і *Out1* є INT і LREAL відповідно. Значення *Out1* дорівнює 1,0, якщо значення *EXP _In* дорівнює 0, коли *EXP _EN* змінюється на TRUE. А значення *Out1* дорівнює 2,71828182845905, оскільки *EXP _In* дорівнює 1.

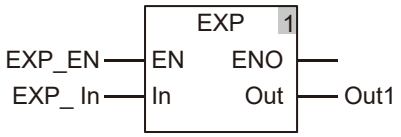
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
EXP_EN	BOOL	TRUE
EXP_In	INT	0
Out1	LREAL	1.0

➢ Змінна 2

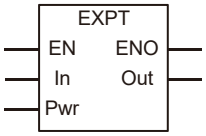
Ім'я змінної	Тип даних	Поточне значення
EXP_EN	BOOL	TRUE
EXP_In	INT	1
Out1	LREAL	2,71828182845905

➢ Програма



8.9.22 EXPT

FB/FC	Пояснення	Застосовна модель
FC	EXPT використовується для виконання операції піднесення до степеня з *In* як базовим числом і *Pwr* як показником степеня. Результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Базове число	Вхід	Базове число	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Pwr	Експонента	Вхід	Експонента	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат розрахунку	Вихід	Результат операції з *In* як базовим числом і *Pwr як* показником степеня	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Pwr		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- EXPT використовується для виконання операції піднесення до степеня з *In* як базовим числом і *Pwr* як показником степеня. І результат виводиться в *Out* .
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежений LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

● **Приклад програмування**

■ типи даних змінних *EXPT _In* та *EXPT_Pwr* мають значення INT із відповідними значеннями 10 і 2 . Тип даних *Out1* — LREAL. Тоді значення *Out1* становить 100,0, коли *EXPT _EN* змінюється на TRUE. Значення *Out1* становить 100,0, оскільки значення *EXPT _In* і *EXPT_Pwr* дорівнюють -10 і 2 відповідно.

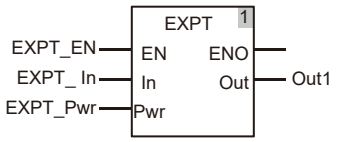
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
EXPT_EN	BOOL	TRUE
EXPT_In	INT	10
EXPT_Pwr	INT	2
Out1	LREAL	100,0

➢ Змінна 2

Ім'я змінної	Тип даних	Поточне значення
EXPT_EN	BOOL	TRUE
EXPT_In	INT	-10
EXPT_Pwr	INT	2
Out1	LREAL	100,0

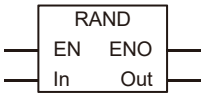
➢ Програма



8

8.9.23 RAND

FB/FC	Пояснення	Застосовна модель
FC	RAND використовується для генерації випадкового числа.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Зарезервований	Вхід	Зарезервований	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Випадкове число	Вихід	Випадкове число	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1								●												
Out												●								

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- RAND використовується для генерації випадкового числа, а результат виводиться в *Out* , у межах діапазону 0 ~ 32767.
- Введене значення не впливає на випадкове число, яке потрібно згенерувати. Але значення має бути введене для *In* .
- Щоб отримати випадкове число в певному діапазоні, користувачам потрібно просто виконати обчислення MOD над згенерованим значенням і отримати залишок. Наприклад, випадкове число від 0 до 10 можна згенерувати, написавши програму RAND(0) MOD10.

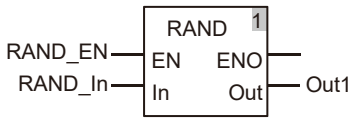
● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

● Приклад програмування

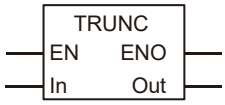
■ Випадкове число генерується записом RAND(0), як показано нижче.
Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
RAND_EN	BOOL	TRUE
RAND_In	INT	0
Out1	DINT	256



8.9.24 TRUNC

FB/FC	Пояснення	Застосовна модель
FC	TRUNC використовується для отримання цілої частини дійсного числа.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Реальне число для перетворення	Вхід	Дійсне число, ціла частина якого отримана	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Ціла частина дійсного числа	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out													●							

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- TRUNC використовується для отримання невід'ємної частини дійсного числа, а результат виводиться в *Out* .
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра – лише LINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LINT.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

- **Приклад програмування**

  - Тип даних *TRUNC _In* — REAL зі значенням -5,6 . Тип даних *Out1* — LINT. Тоді значення *Out1* становить -5, коли *TRUNC _EN* змінюється на TRUE. А значення *Out1* дорівнює 10 як значення *TRUNC _In* 10.8.

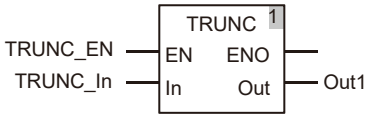
    ➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
TRUNC_EN	BOOL	TRUE
TRUNC _In	REAL	-5,6
Out1	LINT	-5

    ➢ Змінна 2

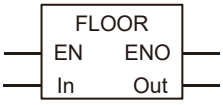
Ім'я змінної	Тип даних	Поточне значення
TRUNC_EN	BOOL	TRUE
TRUNC _In	REAL	10.8
Out1	LINT	10

    ➢ Програма



8.9.25 FLOOR

FB/FC	Пояснення	Застосовна модель
FC	FLOOR використовується для отримання цілої частини дійсного числа. Вихідним значенням є інтегральна частина дійсного числа, віднята на 1, якщо введене дійсне число є від'ємним числом.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Реальне число для перетворення	Вхід	Дійсне число, ціла частина якого отримана	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Ціла частина дійсного числа	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out													●							

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ FLOOR використовується для отримання цілої частини дійсного числа, а результат виводиться в *Out* . Вихідне значення є невід'ємною частиною дійсного числа, якщо вхідне дійсне число є додатним числом. Наприклад, вихідне значення дорівнює 3, якщо вхідне значення дорівнює 3,5. Вихідне значення є невід'ємною частиною дійсного числа, віднімається на 1, якщо вхідне дійсне число є від'ємним числом. Наприклад, вихідне значення -4, якщо вхідне значення -3,5.

■ Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежений LINT. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LINT.

● **Застереження щодо правильного використання**

■ Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

Приклад програмування

- Тип даних змінної *FLOOR _In* — REAL зі значенням 5,6 . Тип даних *Out1* – LINT. Тоді значення *Out1* дорівнює 5, коли *FLOOR _EN* змінюється на TRUE. А значення *Out1* становить -11, а значення *FLOOR _In* -10,2.

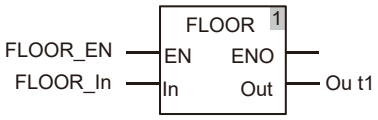
➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
FLOOR_EN	BOOL	TRUE
ПОВЕРХ _In	REAL	5.6
Out1	LINT	5

➢ Змінна 2

Ім'я змінної	Тип даних	Поточне значення
FLOOR_EN	BOOL	TRUE
ПОВЕРХ _In	REAL	-10.2
Out1	LINT	-11

➢ Програма



8.9.26 FRACTION

FB/FC	Пояснення	Застосовна модель
FC	FRACTION використовується для отримання дробової частини дійсного числа.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Реальне число для перетворення	Вхід	Дійсне число, дробова частина якого отримана	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Дробова частина дійсного числа	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out															●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- FRACTION використовується для отримання дробової частини дійсного числа, а результат виводиться в *Out* . Знак результату має збігатися зі знаком вхідного значення.
- Користувачі можуть вибрати різні типи даних для вхідного параметра в цій інструкції. Але тип даних вихідного параметра обмежено LREAL. Під час компіляції програмного забезпечення виникне помилка, якщо тип даних вихідного параметра не LREAL.

● **Застереження щодо правильного використання**

- Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

- **Приклад програмування**

  - Тип даних змінної *FRACTION _In* — REAL зі значенням -5,6 . Тип даних *Out1* — LREAL. Тоді значення *Out1* становить -0,6, коли *FRACTION _EN* змінюється на TRUE. А значення *Out1* дорівнює 0,8 як значення *FRACTION _In* 10,8.

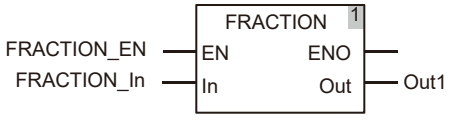
    ➢ Змінна 1

Ім'я змінної	Тип даних	Поточне значення
FRACTION_EN	BOOL	TRUE
FRACTION _In	REAL	-5,6
Out1	LREAL	-0,6

    ➢ Змінна 2

Ім'я змінної	Тип даних	Поточне значення
FRACTION_EN	BOOL	TRUE
FRACTION _In	REAL	10.8
Out1	LREAL	0,8

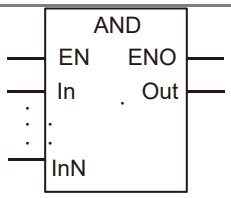
    ➢ Програма



8.10 Інструкції рядка бітів

8.10.1 I

FB/FC	Пояснення	Застосовна модель
FC	I використовується для виконання логічної операції I двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Операнди	Вхід	Кількість операндів можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції I In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

Примітка:

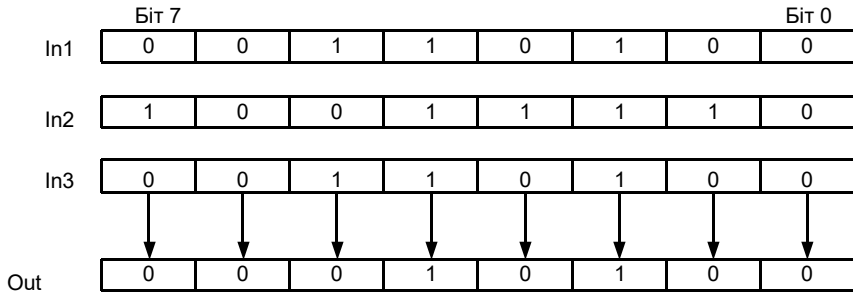
Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ AND використовується для виконання порозрядної логічної операції AND двох або більше змінних або констант, а результат виводиться на *виході* . Тобто *Out* = *In1* & *In2* & ... & *InN*

Правило експлуатації:

Відповідний біт вихідної змінної є TRUE, якщо всі відповідні біти вхідних змінних TRUE, як показано нижче. В іншому випадку відповідний біт вихідної змінної буде FALSE.

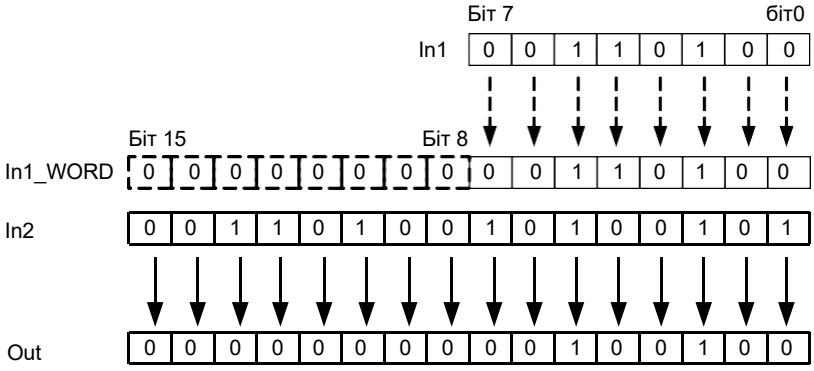


■ *In1 ~ InN* можуть бути змінними різних типів даних, якщо жоден із типів даних вхідних змінних не є BOOL.

Якщо *від In1* до *InN є змінними різних типів даних, візьміть* для операції тип даних, який може включати всі діапазони значень *In1 ~ InN .*

Наприклад, якщо тип даних *In1* — BYTE, а *In2* — WORD, тип даних *Out* — WORD. Під час роботи значення *In1* перетворюється з BYTE на WORD, як показано на наступному малюнку.

виконується логічне І для значень бітів *In1* і *In2 , як показано нижче.*



■ Якщо тип даних вхідної змінної — BOOL, типи даних усіх вхідних і вихідних змінних мають бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

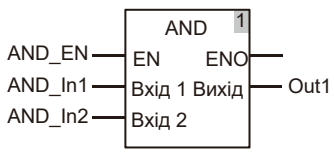
Приклад програмування

■ Усі типи даних AND_In1, AND_In2 і Out1 є BYTE. Значення AND_In1 і

AND_In2 дорівнює 10 і 50 відповідно, а значення Out1 дорівнює 2, коли AND_EN має значення TRUE.

➢ **Таблиця змінних і програма**

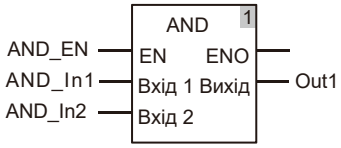
Ім'я змінної	Тип даних	Поточне значення
AND_EN	BOOL	TRUE
AND_In1	BYTE	10
AND_In2	BYTE	50
Out1	КВАРТИРИ	2



■ Типами даних AND_In1, AND_In2 і Out1 є BYTE, WORD і WORD відповідно. Значення AND_In1 і AND_In2 дорівнюють 255 і 256 відповідно, а значення Out1 дорівнює 0, коли AND_EN має значення TRUE.

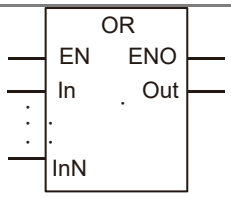
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
AND_EN	BOOL	TRUE
AND_In1	BYTE	255
AND_In2	WORD	256
Out1	WORD	0



8.10.2 OR

FB/FC	Пояснення	Застосовна модель
FC	OR використовується для виконання логічної операції OR двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Операнд	Вхід	Кількість операндів можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції OR In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

Примітка:

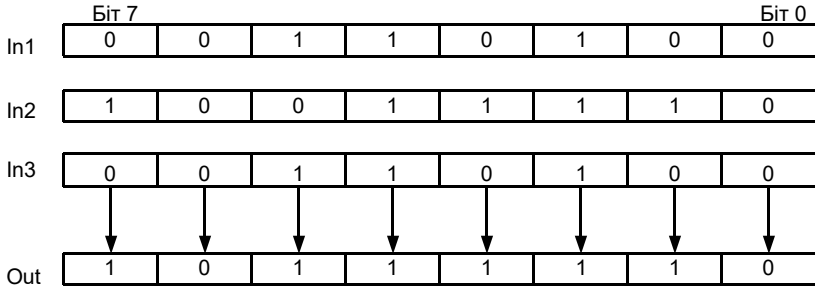
Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ OR використовується для виконання порозрядної логічної операції OR двох або більше змінних або констант, а результат виводиться на *вихід* . Тобто *Out* = *In1* OR *In2* OR…OR *InN.*

Правило експлуатації:

Коли всі відповідні біти всіх вхідних змінних мають значення FALSE, відповідний біт вихідної змінної є FALSE. В іншому випадку відповідний біт вихідної змінної буде TRUE.

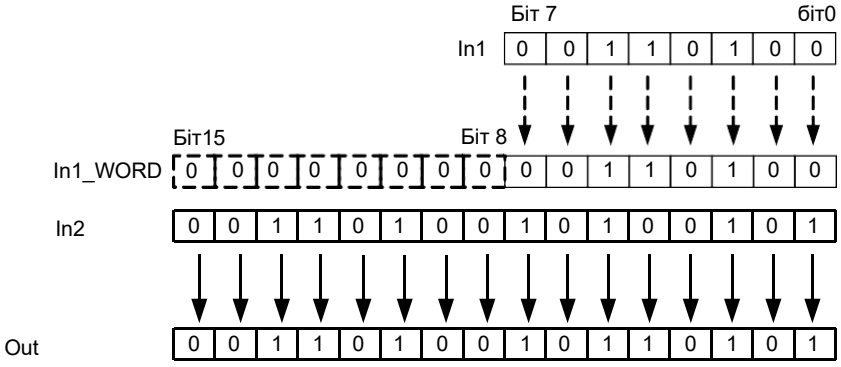


■ *In1 ~ InN* можуть бути змінними різних типів даних, якщо жоден із типів даних вхідних змінних не є BOOL.

Якщо від In1 до *InN є змінними різних типів даних, візьміть* для операції тип даних, який може включати всі діапазони значень *In1 ~ InN .*

Наприклад, якщо тип даних *In1* — BYTE, а *In2* — WORD, тип даних *Out* — WORD. Під час роботи значення *In1* перетворюється з BYTE на WORD, як показано на наступному малюнку.

виконується логічне OR значень бітів *In1* і *In2 , як показано нижче.*



■ Якщо тип даних вхідної змінної — BOOL, типи даних усіх вхідних і вихідних змінних мають бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

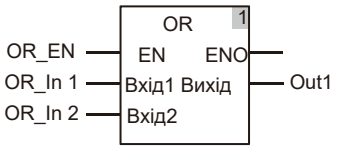
Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

Приклад програмування

■ Усі типи даних OR_In1, OR_In2 і Out1 є BYTE. Значення OR_In1 і OR_In2 дорівнюють 10 і 50 відповідно, а значення Out1 дорівнює 58, коли OR_EN має значення TRUE.

➢ **Таблиця змінних і програма**

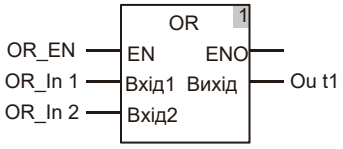
Ім'я змінної	Тип даних	Поточне значення
OR_EN	BOOL	TRUE
OR_In1	BYTE	10
OR_In2	BYTE	50
Out1	BYTE	58



■ Типами даних OR_In1, OR_In2 і Out1 є BYTE, WORD і WORD відповідно. Значення OR_In1 і OR_In2 становлять 255 і 256 відповідно, а значення Out1 дорівнює 511, коли OR_EN має значення TRUE.

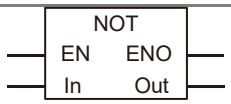
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
OR_EN	BOOL	TRUE
OR_In1	BYTE	255
OR_In2	WORD	256
Out1	WORD	511



8.10.3 NOT

FB/FC	Пояснення	Застосовна модель
FC	NOT використовується для операції NOT, що приймає зворотне значення змінної або константи.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Операнд	Вхід	Вхідний параметр для отримання зворотного	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Не результат операції	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

Примітка:

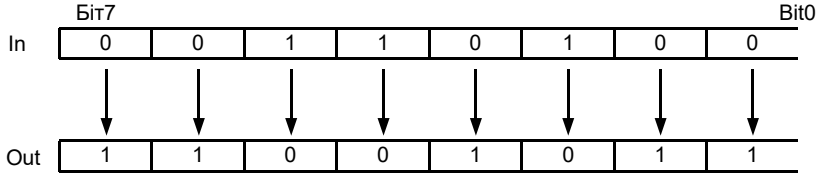
Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ NOT використовується для порозрядної операції NOT, яка приймає зворотне значення змінної або константи, а результат виводиться в *Out* .

Правило експлуатації:

Якщо один біт вхідної змінної дорівнює TRUE, відповідний біт вихідної змінної дорівнює FALSE. Якщо один біт вхідної змінної має значення FALSE, відповідний біт вихідної змінної має значення TRUE.



■ Тип даних *Out* має бути таким самим, як *In* .

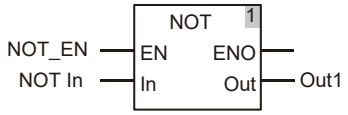
● **Застереження щодо правильного використання**

Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

Приклад програмування

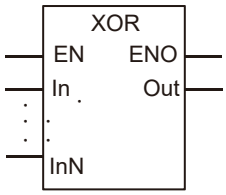
- Обидва типи даних NOT _In та Out1 є BYTE. Значення In1 дорівнює 10, а значення Out1 – 245, коли NOT_EN має значення TRUE.
  - **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
NOT_EN	BOOL	TRUE
NOT In	BYTE	10
Out1	BYTE	245



8.10.4 XOR

FB/FC	Пояснення	Застосовна модель
FC	XOR використовується для операції XOR двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Операнд	Вхід	Кількість операндів можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції XOR In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

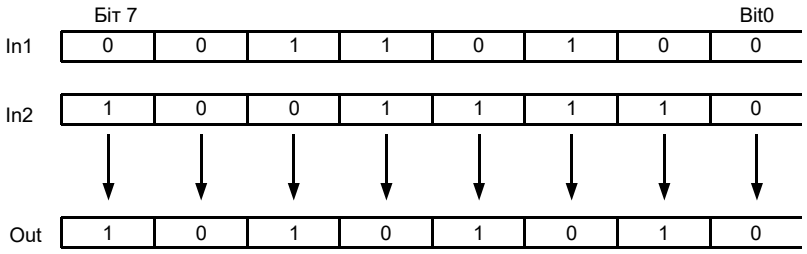
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

Примітка:

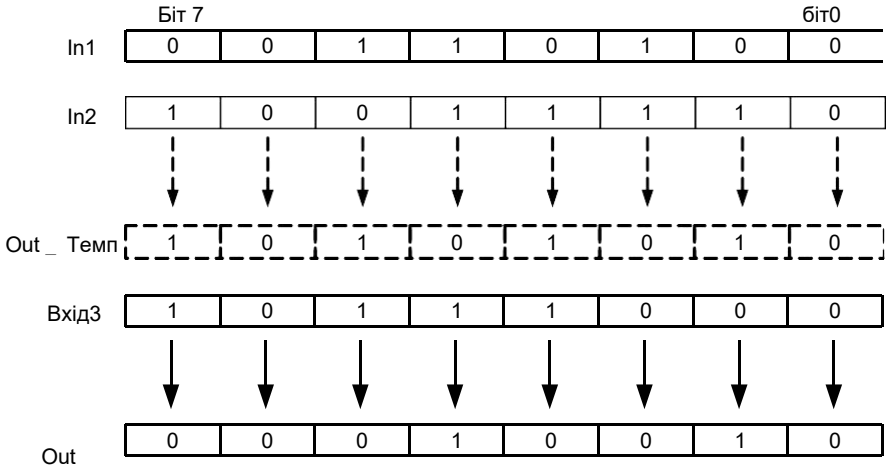
Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

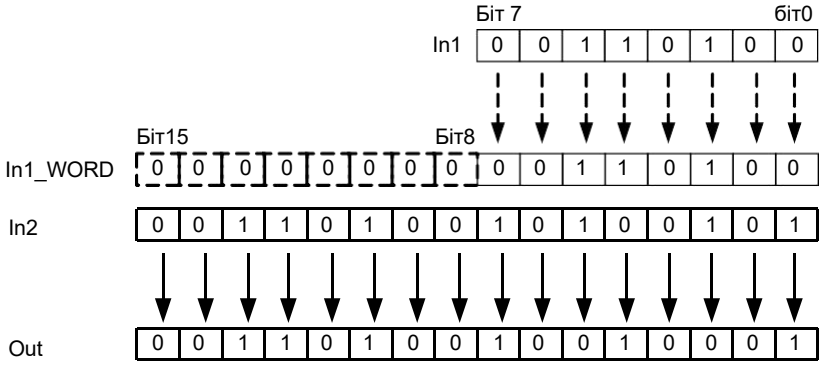
XOR використовується для порозрядної операції XOR двох або більше змінних або констант, а результат виводиться на *Out* . Це *Out* = *In1* XOR *In2* XOR…XOR *InN.*
Операційне правило для XOR In1 і In2 показано на наступному малюнку.



■ Кроки для операції XOR, коли існує більше 2 вхідних параметрів:
Спочатку отримується результат XOR In1 і In2; потім виконується операція XOR попереднього результату та In3 і так далі. Нарешті, обробляється операція XOR попереднього результату XOR і InN. Результатом XOR для In1 і In2 є Out_Temp, а результатом XOR для Out_Temp і In3 є Out, як показано нижче.



■ *In1 ~ InN* можуть бути змінними різних типів даних, якщо жоден із типів даних вхідних змінних не є BOOL.
Якщо *від In1* до *InN* є змінними різних типів даних, візьміть тип даних, який може включати всі діапазони значень *In1 ~ InN* для операції XOR.
Наприклад, якщо тип даних *In1* — BYTE, а *In2* — WORD, тип даних *Out* — WORD. Під час роботи значення *In1* перетворюється з BYTE на WORD, як показано на наступному малюнку. (Біт 8 ~ Біт 15 доповнюється, і всі їхні значення дорівнюють 0.) Потім виконується логічне XOR значень бітів *In1* і *In2 , як показано нижче.*



■ Якщо тип даних вхідної змінної — BOOL, типи даних усіх вхідних і вихідних змінних мають бути BOOL. В іншому випадку під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

Приклад програмування

■ Усі типи даних XOR_In1, XOR_In2 і Out1 є BYTE. Значення XOR_In1 і XOR_In2 дорівнює 10 і 50, а значення Out1 дорівнює 56, коли XOR_EN має значення TRUE, як показано у змінній 1.
Типами даних XOR_In1, XOR_In2 і Out1 є BYTE, WORD і WORD. Значення XOR_In1 і XOR_In2 становлять 255 і 256, а значення Out1 становить 511, коли XOR_EN має значення TRUE, як показано у змінній 2.

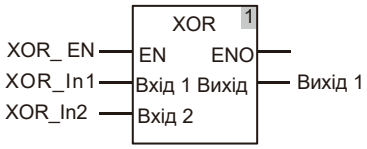
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
XOR_EN	BOOL	TRUE
XOR _In1	BYTE	10
XOR _In2	BYTE	50
Out1	BYTE	56

➢ **Змінна 2**

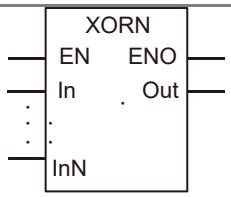
Ім'я змінної	Тип даних	Поточне значення
XOR_EN	BOOL	TRUE
XOR_In1	BYTE	255
XOR_In2	WORD	256
Out1	WORD	511

➢ **Програма**



8.10.5 XORN

FB/FC	Пояснення	Застосовна модель
FC	XORN використовується для операції XORN з двома або більше змінними або константами.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Операнд	Вхід	Кількість операндів можна збільшити або зменшити за допомогою програмного забезпечення. Максимум: 8. Мінімум: 2. Тобто N=2 ~ 8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Результат операції XORN In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●											
Out	●	●	●	●	●	●	●	●	●											

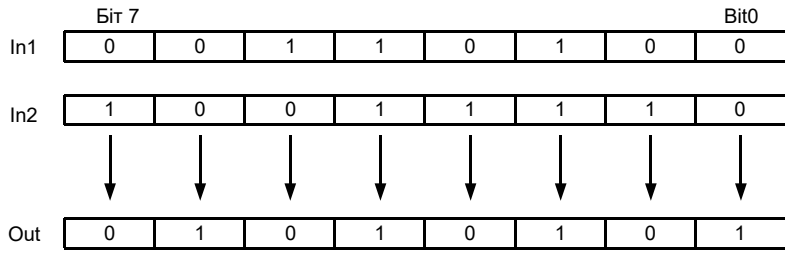
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

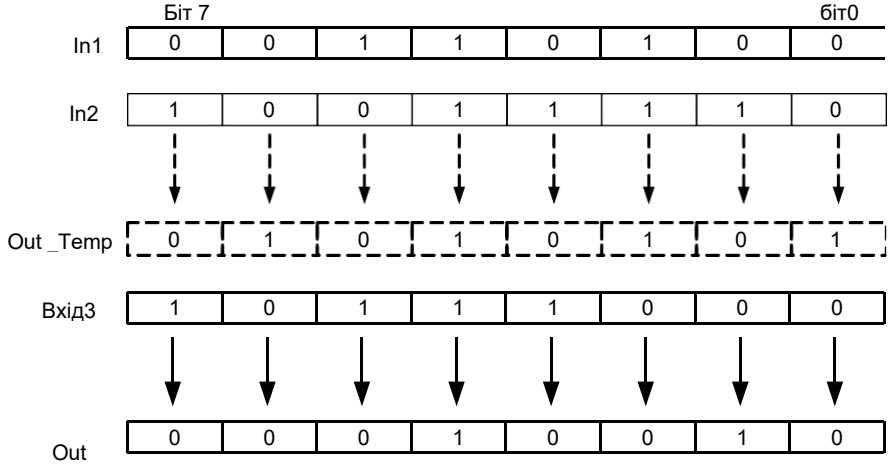
8

● **Пояснення функції**

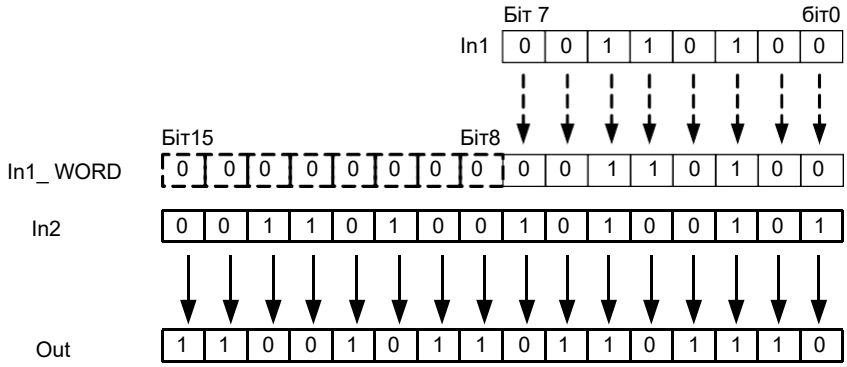
XORN використовується для порозрядного XORN двох або більше змінних або констант, а результат виводиться на *Out* . Тобто *Out = In1* XORN *In2* XORN…XORN *InN.*
Правило роботи для XORN In1 і In2 показано на наступному малюнку.



■ Кроки для роботи XORN призначені для випадків, коли існує більше 2 вхідних параметрів:
Результат XORN In1 і In2 отримується першим; потім виконується XORN попереднього результату та In3 і так далі. Нарешті обробляється XORN попереднього результату XORN і InN.
Результатом XORN для In1 і In2 є Out_Temp, а результатом XORN для Out_Temp і In3 є Out, як показано нижче.



■ *In1 ~ InN* можуть бути змінними різних типів даних, якщо жоден із типів даних вхідних змінних не є BOOL.
Якщо *від In1* до *InN є змінними різних типів даних, візьміть* для операції тип даних, який може включати всі діапазони значень *In1 ~ InN .*
Наприклад, якщо тип даних *In1* — BYTE, а *In2* — WORD, тип даних *Out* — WORD. Під час роботи значення *In1* перетворюється з BYTE на WORD, як показано на наступному малюнку. (Біт 8 ~ Біт 15 доповнюється, і всі їхні значення дорівнюють 0.) Потім виконується логічний XORN бітових значень *In1* і *In2 , як показано нижче.*



■ Якщо тип даних вхідної змінної — BOOL, типи даних усіх вхідних і вихідних змінних мають бути BOOL. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.

Приклад програмування

- Усі типи даних XORN_In1, XORN_In2 і Out1 є BYTE. Значення XORN_In1 і XORN_In2 дорівнює 10 і 50, а значення Out1 дорівнює 199, коли XORN_EN має значення TRUE, як показано у змінній 1. Типами даних XORN_In1, XORN_In2 і Out1 є BYTE, WORD і WORD. Значення XORN_In1 і XORN_In2 становлять 255 і 256, а значення Out1 становить 65535, коли XORN _EN має значення TRUE, як показано у змінній 2.

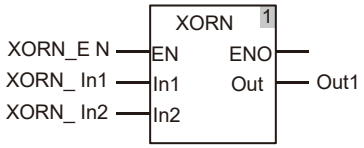
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
XORN_EN	BOOL	TRUE
XORN _In1	BYTE	10
XORN _In2	BYTE	50
Out1	BYTE	199

➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
XORN _EN	BOOL	TRUE
XORN _In1	BYTE	255
XORN _In2	WORD	256
Out1	WORD	65535

➢ **Програма**

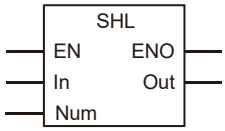


8

8.11 Інструкції по зміні

8.11.1 SHL

FB/FC	Пояснення	Застосовна модель
FC	SHL використовується для зсуву всіх бітів змінної або константи на вказану кількість бітів вліво, а результат виводиться на вихід.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для переміщення	Вхід	Вихідні дані змістити вліво	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Num	Номер для зміни	Вхід	Кількість бітів, на яку всі біти вихідних даних зсуваються вліво	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат	Вихід	Результат зсуву всіх бітів вихідних даних на кількість бітів, визначену Num, вліво	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	Тип даних *Out* має бути таким самим, як *In* .																			

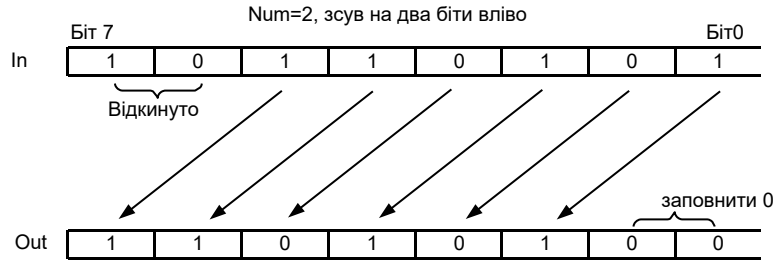
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

SHL використовується для зсуву всіх бітів значення *In* на кількість бітів, визначену *Num* , вліво, а результат виводиться в *Out* .

Коли *Num* =2, усі біти значення *In* зсуваються на два біти вліво, а значення Bit0~Bit1 доповнюються 0, а Bit6~Bit7 відкидаються, як показано на наступному малюнку.

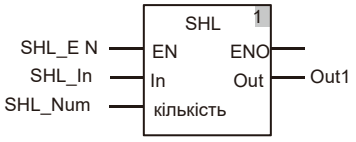


- **Застереження щодо правильного використання**
  - Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
  - Значення *Out* таке саме, як *In* , коли значення *Num* дорівнює 0.

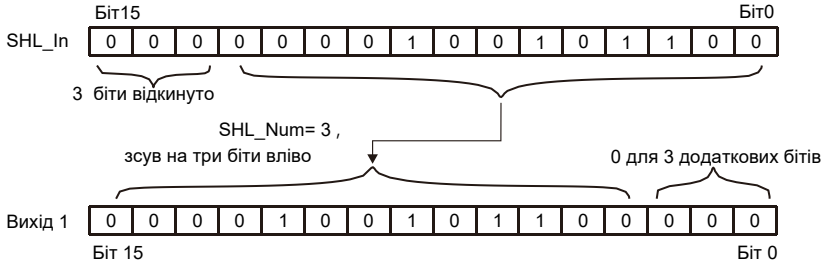
Приклад програмування

- Типи даних SHL_In і SHL_Num — це UINT і USINT відповідно, а їхні значення — 300 і 3 відповідно. Тип даних Out1 — BYTE, а значення Out1 — 2400, коли SHL_EN — TRUE.
  - **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
SHL_EN	BOOL	TRUE
SHL_In	UINT	300
SHL_Num	USINT	3
Out1	UINT	2400

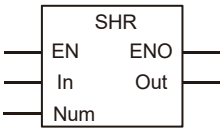


  - **Ілюстрація прикладу**



8.11.2 SHR

FB/FC	Пояснення	Застосовна модель
FC	SHR використовується для зсуву всіх бітів змінної або константи на вказану кількість бітів праворуч, а результат виводиться на *вихід* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для переміщення	Вхід	Вихідні дані змістити вправо	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
кількість	Номер для зміни	Вхід	Кількість бітів, на яку біти вихідних даних зсуваються вправо	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат	Вихід	Результат зсуву всіх бітів вихідних даних на кількість бітів, визначену Num, праворуч	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	Тип даних *Out* має бути таким самим, як *In* .																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

SHR використовується для зсуву всіх бітів значення *In* на кількість бітів, визначену *Num,* праворуч, і результат виводиться на *Out* .

Коли *Num* =2, усі біти значення *In* зсуваються на два біти вправо, а Bit0~Bit1 *In* відкидаються, а значення Bit6~Bit7 доповнюються 0, як показано на наступному малюнку.



● **Застереження щодо правильного використання**

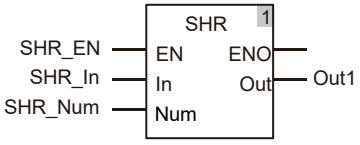
- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Коли значення Num дорівнює 0, значення *Out* є таким самим, як *In* .

Приклад програмування

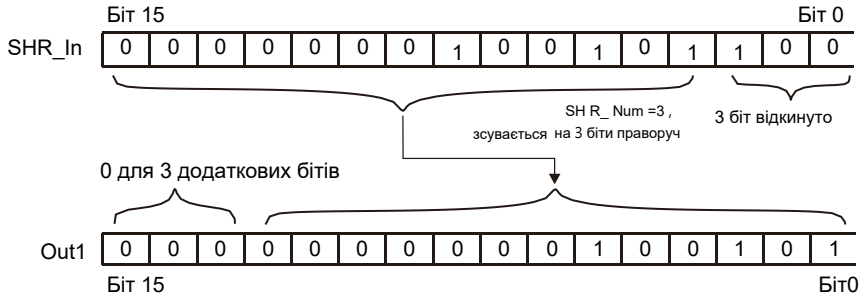
- Типи даних SHR_In і SHR_Num є UINT і USINT відповідно, а їхні значення 300 і 3 відповідно. Тип даних Out1 — UINT, а значення Out1 — 37, коли SHR_EN — TRUE.

➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
SHR_EN	BOOL	TRUE
SHR_In	UINT	300
SHR_Num	USINT	3
Out1	UINT	37

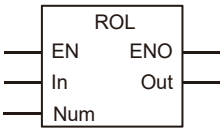


➢ **Ілюстрація прикладу**



8.11.3 ROL

FB/FC	Пояснення	Застосовна модель
FC	ROL використовується для повороту вліво всіх бітів змінної або константи на вказану кількість бітів, а результат виводиться в *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для обертання	Вхід	Вихідні дані для повороту вліво	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Num	Кількість біт	Вхід	Кількість бітів, на яку біти вихідних даних повертаються вліво	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат	Вихід	Результат обертання всіх бітів вихідних даних на кількість бітів, визначену Num, ліворуч	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	Тип даних *Out* має бути таким самим, як *In* .																			

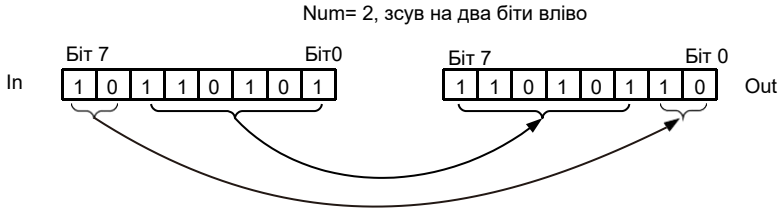
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

ROL використовується для повороту всіх бітів значення *In* на кількість бітів, визначену *Num* , вліво, і результат виводиться в *Out* .

За допомогою ROL біти, зсунуті зліва, один за одним зсуватимуться до нульових бітів справа. Коли *Num* =2, усі біти значення *In* повертаються на два біти вліво. Метод обертання полягає в тому, що Біт0~Біт5 зміщуються на Біт2~Біт7 відповідно, Біт 7 зміщується на Біт1, а Біт 6 — на Біт0.

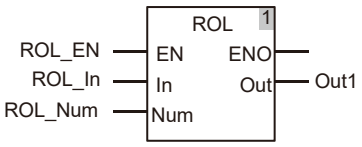


- **Застереження щодо правильного використання**
  - Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
  - Значення *Out* таке саме, як *In* , коли значення *Num* дорівнює 0.
  - Кількість бітів, на яку біти вихідних даних повертаються вліво, дорівнює значенню Num MOD In, якщо значення *Num* більше, ніж кількість бітів значення *In* .
    Наприклад, якщо тип даних *In* – BYTE, значення *out* , коли Num=USINT#1, буде таким самим, коли Num=USINT#9.

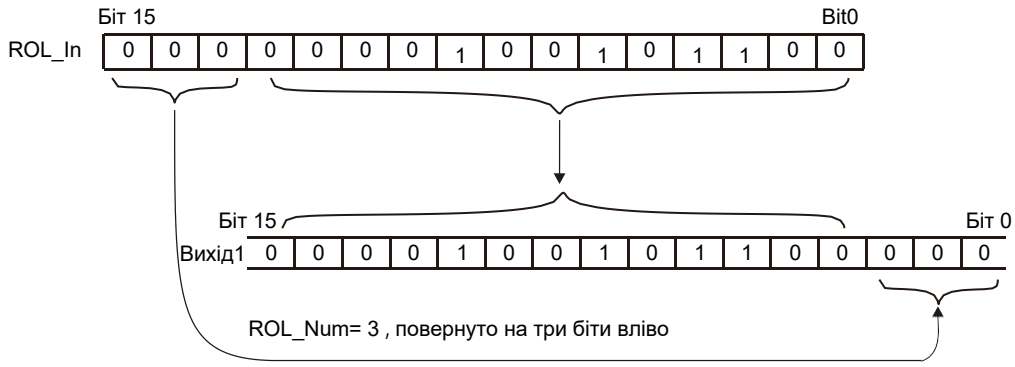
Приклад програмування

- Типи даних ROL_In і ROL_Num є UINT і USINT відповідно, а їхні значення 300 і 3 відповідно. Тип даних Out1 – BYTE, а значення Out1 – 2400, якщо ROL_EN має значення TRUE.
  - **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
ROL_EN	BOOL	TRUE
ROL _In	UINT	300
ROL _Num	USINT	3
Out1	UINT	2400

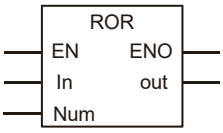


  - **Ілюстрація прикладу**



8.11.4 ROR

FB/FC	Пояснення	Застосовна модель
FC	ROR використовується для повороту всіх бітів змінної або константи на вказану кількість бітів вправо, а результат виводиться на *вихід* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для обертання	Вхід	Вихідні дані для повороту вправо	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Num	Кількість біт	Вхід	Кількість бітів, на яку біти даних повертаються вправо	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат	Вихід	Результат повороту всіх бітів вихідних даних на кількість бітів, визначену Num, праворуч	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●	●	●	●	●											
Num						●														
Out	Тип даних *Out* має бути таким самим, як *In* .																			

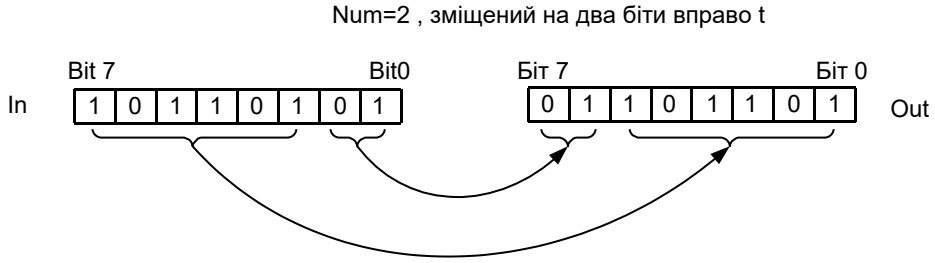
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

ROR використовується для обертання всіх бітів значення *In* на кількість бітів, визначену *Num,* праворуч, і результат виводиться In *Out* .

За допомогою ROR біти, зсунуті справа, один за одним переміститься до нульових бітів ліворуч. Коли *Num* =2, усі біти значення *In* повертаються на два біти праворуч. Метод обертання полягає в тому, що Біт2~Біт7 зсувається на Біт0~Біт5 відповідно, Біт0 зміщується на Біт6, а Біт1 зміщується на Біт7.



● **Застереження щодо правильного використання**

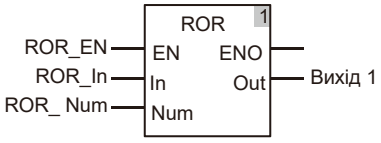
- Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена. Але вихідну змінну можна опускати.
- Значення *Out* таке саме, як *In* , коли значення *Num* дорівнює 0.
- Кількість бітів, на яку біти даних повертаються вправо, дорівнює значенню Num MOD In, коли значення *Num* більше, ніж кількість бітів значення *In* .
  Наприклад, якщо тип даних *In* – BYTE, значення *out* , коли Num=USINT#1, буде таким самим, коли Num=USINT#9.

Приклад програмування

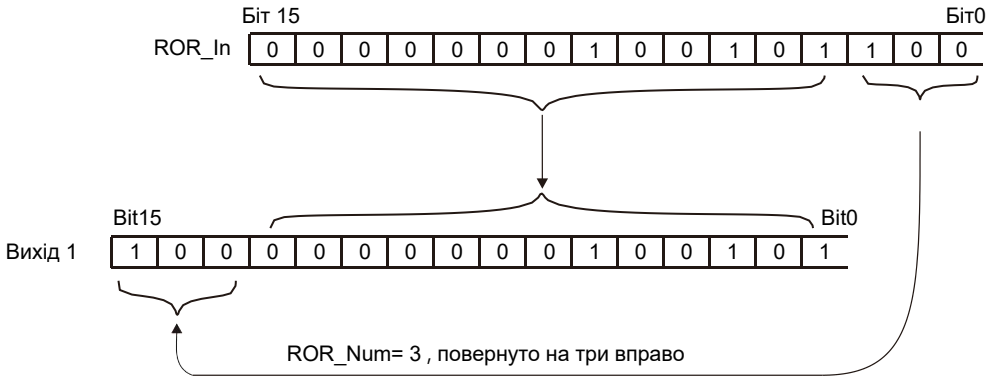
- Типи даних ROR_In і ROR_Num є UINT і USINT відповідно, а їхні значення 300 і 3 відповідно.
  Тип даних Out1 – BYTE, а значення Out1 – 32805, якщо ROR_EN має значення TRUE.

➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
ROR_EN	BOOL	TRUE
ROR_In	UINT	300
ROR_Num	USINT	3
Out1	UINT	32805



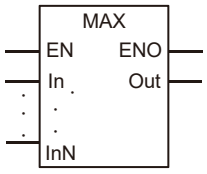
➢ **Ілюстрація прикладу**



8.12 Інструкції з вибору

8.12.1 MAX

FB/FC	Пояснення	Застосовна модель
FC	Макс використовується для знаходження найбільшого значення двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Дані порівняння можна додавати або видаляти під час написання програми. Максимальна кількість порівняльних даних – 8. N=2~8	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Найбільше значення In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

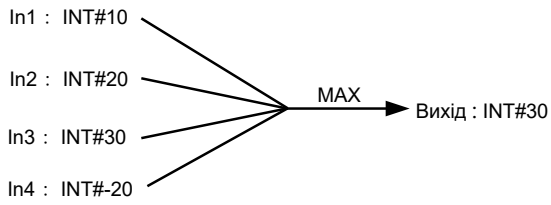
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція Max знаходить найбільше значення двох або більше змінних або констант, і найбільше значення виводиться на *Out* .



- Якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD або STRING, вхідні параметри *In1~InN* можуть бути змінними різних типів даних.

- Коли типи даних вхідних змінних є одним із таких: BOOL, TIME, DATE, TOD і STRING, усі вхідні змінні та вихідна змінна мають бути типу даних. Наприклад, якщо типом даних *In1* є TIME, типом даних *In2~InN* має бути TIME. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

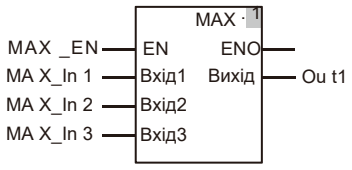
- Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.
- Довжина типу даних вихідної змінної повинна містити довжину всіх вхідних параметрів. Інакше під час компіляції програмного забезпечення виникне помилка

Приклад програмування

- Типи даних MAX_In1, MAX_In2 і MAX_In3 є INT, UINT і DINT відповідно. The Тип даних Out1 – DINT. Якщо значення MAX_In1, MAX_In2 і MAX_In3 дорівнюють -10, 50 і 100 відповідно, значення Out1 дорівнює 100, коли MAX_EN має значення TRUE.

  ➢ **Таблиця змінних і програма**

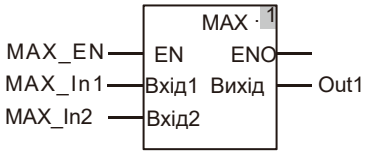
Ім'я змінної	Тип даних	Поточне значення
MAX_EN	BOOL	TRUE
MAX_In1	INT	- 10
MAX_In2	UINT	50
MAX_In3	DINT	100
Out1	DINT	100



- Типами даних MAX_In1 і MAX_In2 є TIME. Тип даних Out1 – TIME.
  Якщо значення MAX_In1 і MAX_In2 дорівнюють T#1ms і T#50ms відповідно, значення Out1 дорівнює T#50ms, коли MAX_EN має значення TRUE.

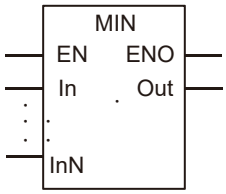
  ➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
MAX_EN	BOOL	TRUE
MAX_In1	TIME	T#1ms
MAX_In2	TIME	T#50 мс
Out1	TIME	T#50 мс



8.12.2 MIN

FB/FC	Пояснення	Застосовна модель
FC	MIN використовується для знаходження найменшого значення двох або більше змінних або констант.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Дані порівняння	Вхід	Дані порівняння можна додавати або видаляти під час написання програми. Максимальна кількість порівняльних даних – 8. N=2~8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат порівняння	Вихід	Найменше значення In1 ~ InN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

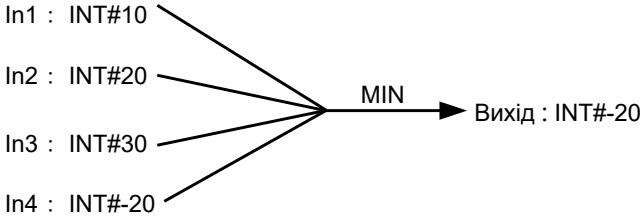
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

Пояснення функції

■ Інструкція MIN знаходить найменше значення двох або більше змінних і констант, і найменше значення виводиться на *Out* .



- Якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD або STRING, вхідні параметри *In1~InN* можуть бути змінними різних типів даних.
- Коли типи даних вхідних змінних є одним із таких: BOOL, TIME, DATE, TOD і STRING, усі вхідні змінні та вихідна змінна мають бути типу даних. Наприклад, якщо типом даних *In1* є TIME, типом даних *In2~InN* має бути TIME. Інакше під час компіляції програмного забезпечення виникне помилка.

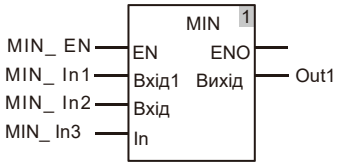
● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.
- Довжина типу даних вихідної змінної повинна містити довжину всіх вхідних параметрів. Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- Типи даних MIN_In1, MIN_In2 і MIN_In3 є INT, UINT і DINT відповідно. Дані тип Out1 – DINT. Якщо значення MIN_In1, MIN_In2 і MIN_In3 дорівнюють -10, 50 і 100 відповідно, значення Out1 дорівнює -10, коли MIN_EN має значення TRUE.
  - **Таблиця змінних і програма**

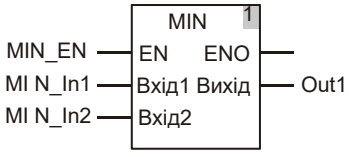
Ім'я змінної	Тип даних	Поточне значення
MIN_EN	BOOL	TRUE
MIN_In1	INT	- 10
MIN_In2	UINT	50
MIN_In3	DINT	100
Out1	DINT	- 10



- Типами даних MIN_In1 і MIN_In2 є TIME. Тип даних Out1 – TIME.

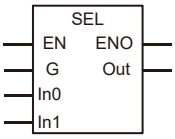
  Якщо значення MIN_In1 і MIN_In2 дорівнюють T#1ms і T#50ms відповідно, значення Out1 дорівнює T#1ms, коли MIN_EN має значення TRUE.
  - **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
MIN_EN	BOOL	TRUE
MIN_In1	TIME	T#1ms
MIN_In2	TIME	T#50 мс
Out1	TIME	T#1ms



8.12.3 SEL

FB/FC	Пояснення	Застосовна модель
FC	SEL використовується для вибору однієї з двох змінних або констант, а вибране значення виводиться на *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
G	Gate (Ворота)	Вхід	In0 вибрано, коли G має значення FALSE; In1 вибирається, коли G має значення TRUE.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In0 і In1	Вибірки	Вхід	Дані для вибору	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат відбору	Вихід	Результат відбору	Залежить від типу даних змінної, до якої підключено вихідний параметр.

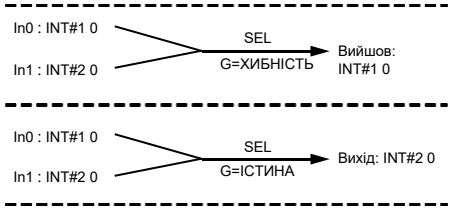
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
G	●																			
In0 і In1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Згідно з умовою вибору G, інструкція SEL вибирає одну з двох змінних або констант, і результат вибору виводиться на *Out* .



- Якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD або STRING, вхідні параметри *In0~In1* можуть з'єднувати змінні різних типів даних.
- Коли типи даних вхідних змінних є одним із таких: BOOL, TIME, DATE, TOD і STRING, усі вхідні змінні та вихідна змінна мають бути типу даних. Наприклад, якщо типом даних змінної, пов'язаної з *In0,* є TIME, типи даних змінних, пов'язаних з *In1* і *Out,* мають бути TIME. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.
- Довжина типу даних вихідної змінної повинна містити довжину змінних, які з'єднують вхідні параметри *In0* та *In1* . Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- Типами даних SEL_G, SEL_In0 і SEL_In1 є BOOL, UINT і DINT, а типом даних Out1 є DINT. Якщо SEL_EN дорівнює TRUE, значення Out1 дорівнює 50, якщо значення SEL_G, SEL_In0 і SEL_In1 є FALSE, 50 і 100 відповідно, як показано в наведеній нижче таблиці змінної 1. Якщо значення SEL_G, SEL_In0 і SEL_In1 дорівнюють TRUE, 50 і 100 відповідно, значення Out1 дорівнює 100, як показано в наступній таблиці змінної 2.

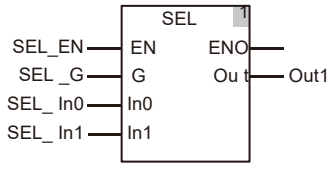
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
SEL_EN	BOOL	TRUE
SEL_G	BOOL	FALSE
SEL_In0	UINT	50
SEL_In1	DINT	100
Out1	DINT	50

➢ **Змінна 2**

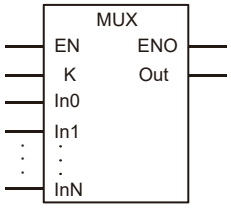
Ім'я змінної	Тип даних	Поточне значення
SEL_EN	BOOL	TRUE
SEL_G	BOOL	TRUE
SEL_In0	UINT	50
SEL_In1	DINT	100
Out1	DINT	100

■ **Програма**



8.12.4 MUX

FB/FC	Пояснення	Застосовна модель
FC	MUX використовується для вибору однієї з двох або більше змінних або констант, а результат виводиться на *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
K	Ворота	Вхід	Ворота	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In0 , In1 до InN	Вибірки	Вхід	Вибір можна додавати або видаляти під час написання програми. Максимальна кількість виборів – 8. N=2~8.	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат відбору	Вихід	Результат відбору	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
K						●														
In0, In1 до InN	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Базуючись на умові вибору *K* , інструкція MUX вибирає один із *In0~InN* , і результат вибору виводиться на *Out* . Значення *Out* відповідає значенню *K* , як показано в наступній таблиці.

Значення до	Значення Out
0	In0
1	In1
2	In2
3	In3
4	In4
5	In5
6	In6
7	In7

- Якщо типи даних вхідних змінних не є BOOL, TIME, DATE, TOD або STRING, вхідні параметри *In0~InN* можуть з'єднувати змінні різних типів даних.
- Коли типи даних вхідних змінних є одним із таких: BOOL, TIME, DATE, TOD і STRING, усі вхідні змінні та вихідна змінна мають бути типу даних. Наприклад, якщо тип даних *In0* - TIME, типи даних змінних, пов'язаних з *In1~InN* і *Out,* повинні бути TIME. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

- Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.
- Довжина типу даних вихідної змінної повинна містити довжину змінних, які з'єднують вхідні параметри *In0 ~ InN* . Інакше під час компіляції програмного забезпечення виникне помилка.

Приклад програмування

- Типами даних MUX_K, MUX_In0 і MUX_In1 є UINT, UINT і DINT, а типом даних Out1 є DINT. Коли MUX_EN дорівнює TRUE, значення Out1 дорівнює 50, якщо значення MUX_K, MUX_In0 і MUX_In1 дорівнюють 0, 50 і 100, як показано в наступній таблиці змінної 1. Якщо значення MUX_K, MUX_In0 і MUX_In1 дорівнюють 1, 50 і 100, значення Out1 дорівнює 100, як показано в наступній таблиці змінної 2.

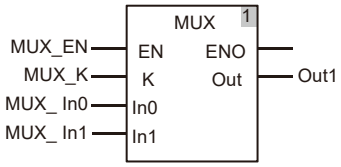
➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
MUX_EN	BOOL	TRUE
MUX_K	USINT	0
MUX_In0	UINT	50
MUX_In1	DINT	100
Out1	DINT	50

➢ **Змінна 2**

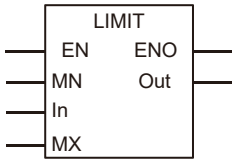
Ім'я змінної	Тип даних	Поточне значення
MUX_EN	BOOL	TRUE
MUX_K	UINT	1
MUX_In0	UINT	50
MUX_In1	DINT	100
Out1	DINT	100

➢ **Програма**


MUX
1
MUX_EN
EN
ENO
MUX_K
K
Out
Out1
MUX_ In0
In0
MUX_ In1
In1

8.12.5 LIMIT

FB/FC	Пояснення	Застосовна модель
FC	LIMIT використовується для обмеження вихідного значення в межах зони між вказаними мінімальним і максимальним значеннями.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
MN	Мінімальна вартість	Вхід	Мінімальна вартість	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In	Дані для обмеження	Вхід	Дані для обмеження	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
MX	Максимальне значення	Вхід	Максимальне значення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат обробки	Вихід	Результат обробки	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
In		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
MX		●	●	●	●	●	●	●	●	●	●	●	●	●	●					
Out		●	●	●	●	●	●	●	●	●	●	●	●	●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція LIMIT обмежує значення в діапазоні між MN і MX, а результат виводиться на *вихід* .

Значення In	Значення Out
In < MN	MN
MN ≤ In ≤ MX	In

Значення In	Значення Out
MX < In	MX

- Інструкція дозволяє за допомогою вхідних параметрів *MN* , *In* і *MX* з'єднати змінні різних типів даних. Коли *MN* , *In* і *MX* є змінними різних типів даних, обчислення виконується з типом даних, який може містити діапазон значень *MN* , *In* і *MX* . Наприклад, якщо тип даних *MN* — INT, а типи даних *In* і *MX* — DINT, тип даних *Out* — DINT.
- Інструкція дозволяє вхідні параметри та вихідні параметри з'єднати змінні різних типів даних. Але довжина типу даних вихідної змінної повинна містити довжину змінних, які з'єднують вхідні параметри *In0 ~ InN* . Інакше під час компіляції програмного забезпечення виникне помилка.

Застереження щодо правильного використання

- Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

- Типи даних LIMIT_MN, LIMIT_In і LIMIT_MX — UINT, UINT і DINT, а тип даних Out1 — DINT. Коли LIMIT_EN дорівнює TRUE, значення Out1 дорівнює 50, якщо значення LIMIT_MN, LIMIT_In і LIMIT_MX дорівнюють 1, 50 і 100, як показано в наведеній нижче змінній таблиці 1. Якщо значення LIMIT_MN, LIMIT_In і LIMIT_MX дорівнюють 2, 200 і 100 , значення Out1 дорівнює 100, як показано в наступну змінну таблиці 2. Якщо значення LIMIT_MN, LIMIT_In і LIMIT_MX дорівнюють 50, 10 і 100, значення Out1 дорівнює 50, як показано в наступній таблиці змінної 3.

➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	1
LIMIT_In	UINT	50
LIMIT_MX	DINT	100
Out1	DINT	50

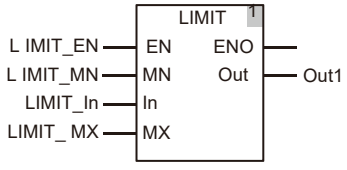
➢ **Змінна 2**

Ім'я змінної	Тип даних	Поточне значення
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	2
LIMIT_In	UINT	200
LIMIT_MX	DINT	100
Out1	DINT	100

➢ **Змінна 3**

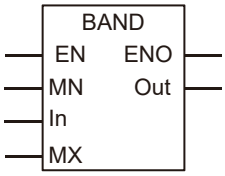
Ім'я змінної	Тип даних	Поточне значення
LIMIT_EN	BOOL	TRUE
LIMIT_MN	UINT	50
LIMIT_In	UINT	10
LIMIT_MX	DINT	100
Out1	DINT	50

➢ **Програма**



8.12.6 BAND

FB/FC	Пояснення	Застосовна модель
FC	BAND виконує контроль зони нечутливості, і результат обробки виводиться на *Out*	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
MN	Мінімальна вартість	Вхід	Мінімальна вартість	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In	Дані для обмеження	Вхід	Дані для обмеження	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
MX	Максимальне значення	Вхід	Максимальне значення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат обробки	Вихід	Результат обробки	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
MN														●	●					
In														●	●					
MX														●	●					
Out														●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція BAND виконує контроль зони нечутливості значення *In* відповідно до максимального значення *MX* і мінімального значення *MN ,* а результат обробки виводиться на ***вихід*** .

Значення In	Значення Out
In < MN	In - MN
MN ≤In ≤MX	0
MX < In	In - MX

■ Інструкція дозволяє за допомогою вхідних параметрів *MN* , *In* і *MX* з'єднати змінні різних типів даних. Коли *MN* , *In* і *MX* є змінними різних типів даних, обчислення виконується з типом даних, який може містити діапазон значень *MN* , *In* і *MX* . Наприклад, якщо тип даних *MN* — REAL, а типи даних *In* і *MX* — LREAL, тип даних *Out* — LREAL.

■ Інструкція дозволяє вхідні параметри та вихідні параметри з'єднати змінні різних типів даних. Але довжина типу даних вихідної змінної повинна містити довжину змінних, які з'єднують вхідні параметри. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

■ Коли значення *MN* більше, ніж значення *MX* , інструкція все ще виконуватиметься нормально, а значення *Out* буде рівним значенню *MX* .

Приклад програмування

■ Типи даних BAND_MN, BAND_In і BAND_MX є REAL, а тип даних Out1 – LREAL. Коли BAND_EN дорівнює TRUE, значення Out1 дорівнює 0, якщо значення BAND_MN, BAND_In і BAND_MX дорівнюють 1, 50 і 100, як показано в наведеній нижче змінній таблиці 1. Якщо значення BAND_MN, BAND_In і BAND_MX дорівнюють 2, 250 і 100 , значення Out1 дорівнює 150 (150=250-100), як показано в наступній таблиці Змінна 2. Якщо значення BAND_MN, BAND_In і BAND_MX 50, 10 і 100, значення Out1 дорівнює -40 (- 40 = 10 – 50), як показано в наступній таблиці Змінна 3.

➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
BAND_EN	BOOL	TRUE
BAND_MN	REAL	1
BAND_In	REAL	50
BAND_MX	REAL	100
Out1	LREAL	0

➢ **Змінна 2**

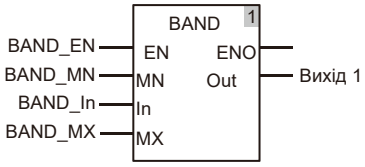
Ім'я змінної	Тип даних	Поточне значення
BAND_EN	BOOL	TRUE
BAND_MN	REAL	2
BAND_In	REAL	250
BAND_MX	REAL	100
Out1	LREAL	150

➢ **Змінна 3**

Ім'я змінної	Тип даних	Поточне значення
BAND_EN	BOOL	TRUE
BAND_MN	REAL	50

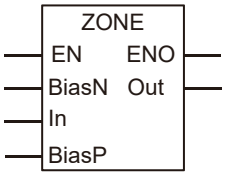
Ім'я змінної	Тип даних	Поточне значення
BAND_In	REAL	10
BAND_MX	REAL	100
Out1	LREAL	-40

- **Програма**



8.12.7 ZONE

FB/FC	Пояснення	Застосовна модель
FC	ZONE використовується для додавання значення зсуву до вхідного значення, а результат обробки виводиться на *Out* .	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
BiasN	Від'ємне значення зсуву	Вхід	Негативне упередження	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
In	Дані для контролю	Вхід	Дані для контролю	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
BiasP	Позитивне значення зсуву	Вхід	Позитивне упередження	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
Out	Результат обробки	Вихід	Результат обробки	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
BiasN														●	●					
In														●	●					
BiasP														●	●					
Out														●	●					

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція ZONE додає встановлене значення зсуву до значення *In* , а результат обробки виводиться в *Out* . Якщо значення *In* є від'ємним значенням, *BiasN* є значенням зсуву. Коли значення *In* є позитивним значенням, *BiasP* є значенням зсуву.

Значення In	Значення Out
B<0	In+BiasN
B=0	0
B>0	In+BiasP

■ Інструкція дозволяє вхідні параметри *BiasN* , *In* і *BiasP* з'єднувати змінні різних типів даних. Якщо *BiasN* , *In* і *BiasP* є змінними різних типів даних, обчислення виконується з типом даних, який може містити діапазон значень *BiasN* , *In* і *BiasP* . Наприклад, якщо тип даних *BiasN* — INT, а типи даних *In* і *BiasP* — DINT, тип даних *Out* — DINT.

■ Інструкція дозволяє вхідні параметри та вихідні параметри з'єднати змінні різних типів даних. Але довжина типу даних вихідної змінної повинна містити довжину змінних, які з'єднують вхідні параметри. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

■ Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідні змінні опущено. Але вихідну змінну можна опускати.

■ Якщо значення *BiasN* більше, ніж *BiasP,* інструкція все одно виконуватиметься нормально.

Приклад програмування

■ Типами даних ZONE_BiasN, ZONE_In і ZONE_BiasP є INT, INT і DINT, а типом даних Out1 є DINT. Коли ZONE_EN дорівнює TRUE, значення Out1 дорівнює 0, якщо значення ZONE_BiasN, ZONE_In і ZONE_BiasP дорівнюють 1, 0 і 100, як показано в таблиці змінної 1. Якщо значення ZONE_BiasN, ZONE_In і ZONE_BiasP дорівнюють 2, 50 і 100 , значення Out1 дорівнює 150 (150 = 50 + 100), як показано в наступній таблиці Змінна 2. Якщо значення ZONE_BiasN, ZONE_In і ZONE_BiasP дорівнюють 50, -10 і 100, значення Out1 дорівнює 40 (40 = - 10 + 50), як показано в наступній таблиці змінної 3.

➢ **Змінна 1**

Ім'я змінної	Тип даних	Поточне значення
ZONE_EN	BOOL	TRUE
ZONE_BiasN	INT	1
ZONE_In	INT	0
ZONE_BiasP	DINT	100
Out1	DINT	0

➢ **Змінна 2**

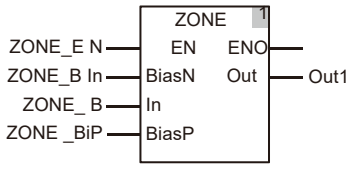
Ім'я змінної	Тип даних	Поточне значення
ZONE_EN	BOOL	TRUE
ZONE_BiasN	INT	2
ZONE_In	INT	50
ZONE_BiasP	DINT	100
Out1	DINT	150

➢ **Змінна 3**

Ім'я змінної	Тип даних	Поточне значення
ZONE_EN	BOOL	TRUE
ZONE_BiasN	INT	50

Ім'я змінної	Тип даних	Поточне значення
ZONE_In	INT	- 10
ZONE_BiasP	DINT	100
Out1	DINT	40

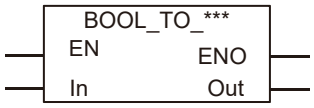
➢ **Програма**



8.13 Інструкції з перетворення типу даних

8.13.1 BOOL_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції BOOL_TO_*** перетворюють булеві дані в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In	●																			
Out		Тип даних *Out* має збігатися з «***» назви інструкції.																		

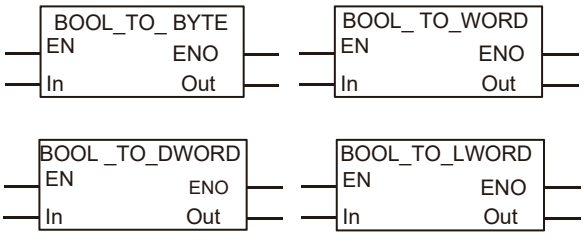
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ **BOOL в бітовий рядок**

➢ Відповідні інструкції:

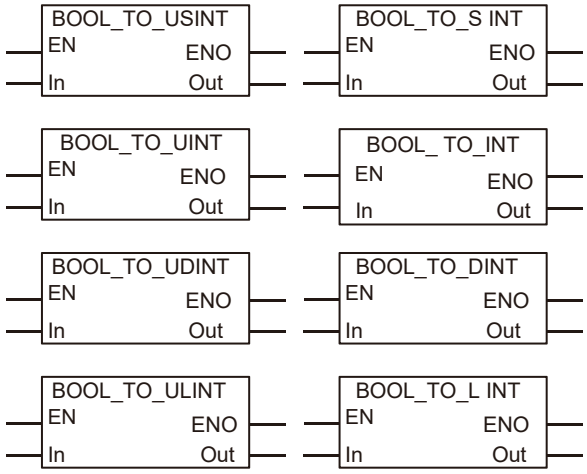


- Правило для перетворення з логічного значення на бітовий рядок показано в наступній таблиці. (Потрібно підтвердити формат значення бітового рядка та

Логічний	Бітовий рядок			
	BYTE	WORD	DWORD	LWORD
FALSE	16#00	16#0000	16#0000_0000	16#0000_0000_0000_0000
TRUE	16#01	16#0001	16#0000_0001	16#0000_0000_0000_0001

■ BOOL в ціле число

- Відповідні інструкції:

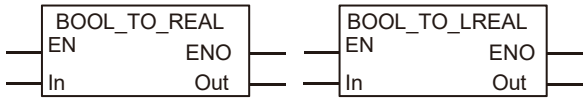


- У наведеній нижче таблиці показано правило, згідно з яким логічні дані

Логічний	Ціле число							
	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT
FALSE	0	0	0	0	0	0	0	0
TRUE	1	1	1	1	1	1	1	1

■ BOOL до дійсного числа

- Відповідні інструкції:

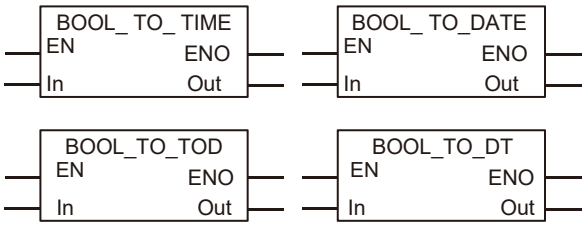


- У наведеній нижче таблиці показано правило, згідно з яким логічні дані перетворюються на дані дійсного числа.

Логічний	Справжня	
	REAL	LREAL
FALSE	0	0
TRUE	1	1

■ **BOOL для часу та дати**

➢ Відповідні інструкції:

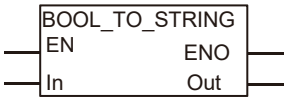


➢ Правило, згідно з яким логічні дані перетворюються на дані часу або дати, показано в наведеній нижче таблиці.

Логічний	**Час і Дата**			
	TIME	DATE	TOD	DT
FALSE	T#0ms	D#1970-1-1	TOD#	DT#
TRUE	T#1ms	D#1970-1-1	TOD#	DT#

■ **BOOL в String**

➢ Відповідні інструкції:



➢ Правило, згідно з яким логічні дані перетворюються на рядкові дані, показано в наведеній нижче таблиці. (Формат рядка потрібно підтвердити.)

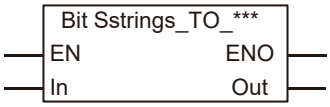
Логічний	**Рядок**
	STRING
FALSE	'FALSE'
TRUE	"TRUE"

● **Застереження щодо правильного використання**

Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Вихідну змінну можна опускати.

8.13.2 Бітові рядки_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції Bit strings_TO_*** перетворюють дані бітового рядка в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In		●	●	●	●															
Out	Тип даних *Out* має збігатися з «***» назви інструкції.																			

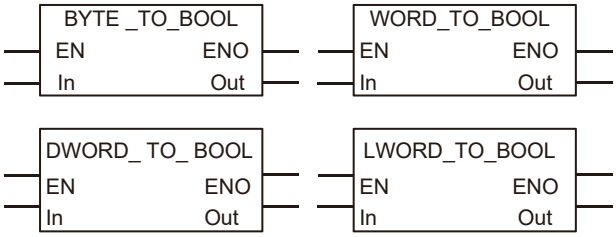
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ **Бітовий рядок для BOOL**

➢ Відповідні інструкції:

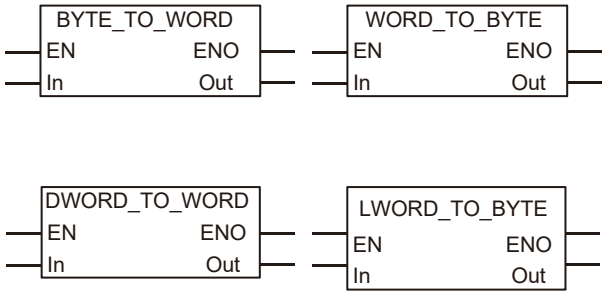


➢ Правило, згідно з яким бітові рядкові дані перетворюються на логічні дані, показано в

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
BYTE	BOOL	16#00	FALSE
		16#01~16#FF	TRUE
WORD	BOOL	16#0000	FALSE
		16#0001~16#FFFF	TRUE
DWORD	BOOL	16#0000_0000	FALSE
		16#0000_0001~16#FFFF_FFFF	TRUE
LWORD	BOOL	16#0000_0000_0000_0000	FALSE
		16#0000_0000_0000_0001~ 16#FFFF_FFFF_FFFF_FFFF	TRUE

■ **Бітовий рядок до бітового рядка**

➢ Дані бітового рядка можна перетворити на дані бітового рядка. Деякі інструкції наведено нижче.



Існує два види перетворення для різних типів бітових рядкових даних. Одним з них є перетворення даних меншої довжини в дані більшої довжини. Інший — це перетворення даних більшої довжини в дані меншої довжини.

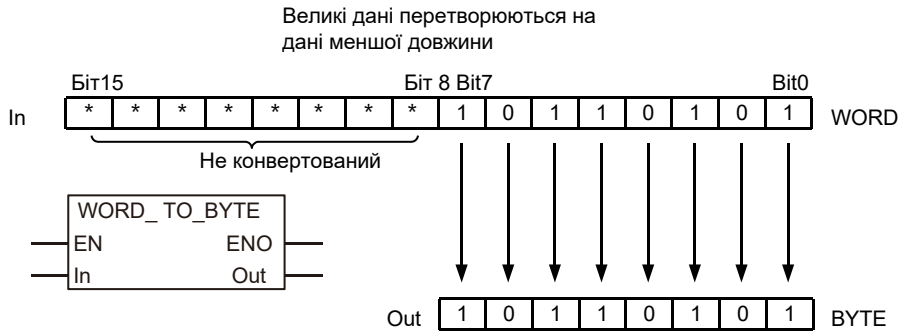
Дані меншої довжини перетворюються на дані більшої довжини шляхом запису значень усіх бітів даних меншої довжини у відповідні біти даних більшої довжини та встановлення значень решти бітів даних більшої довжини на 0.

Перегляньте наведений нижче приклад, коли дані Byte у *In* перетворюються на дані Word у *Out* . Значення Bit0~Bit7 *In* скопійовано та вставлено в Bit0~Bit7 *Out* . А значення Bit8~Bit15 *Out* встановлюються на 0.



Дані більшої довжини перетворюються на дані меншої довжини шляхом перегляду значень усіх бітів даних меншої довжини на значення відповідних бітів даних більшої довжини та значень решти бітів більшої довжини. дані про довжину не перетворюються і не впливають на перетворення.

Перегляньте наведений нижче приклад, коли дані Word *In* перетворюються на дані Byte *Out* . Значення Bit0~Bit7 *In* скопійовано та вставлено в Bit0~Bit7 *Out* . А значення Біт8~Біт15 *In* не перетворюються і не впливають на перетворення.

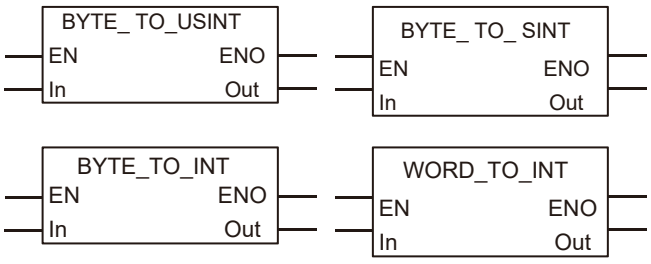


➢ Дані бітового рядка перетворюються на дані бітового рядка, як показано в наведеній нижче таблиці.

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
BYTE	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~ 16#0000_0000_0000_00FF
WORD	BYTE	16#**00~16#**FF	16#00~16#FF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~ 16#0000_0000_0000_FFFF
DWORD	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
LWORD	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF

■ **Бітовий рядок до цілого числа**

➢ Дані бітового рядка можна перетворити на ціле число. Деякі інструкції наведено нижче.



Правило для перетворення даних бітового рядка в цілі дані узгоджується з правилом для перетворення даних бітового рядка в дані бітового рядка.

Дані меншої довжини перетворюються на дані більшої довжини шляхом запису значень усіх бітів даних меншої довжини у відповідні біти даних більшої довжини та встановлення значень решти бітів даних більшої довжини на 0.

Дані більшої довжини перетворюються на дані меншої довжини шляхом перегляду значень усіх бітів даних меншої довжини на значення відповідних бітів даних більшої довжини та значень решти бітів більшої довжини. дані про довжину не перетворюються і не впливають на перетворення.

Якщо довжини двох даних для перетворення рівні, усі значення всіх бітів *In* копіюються та вставляються у відповідні біти *Out* .

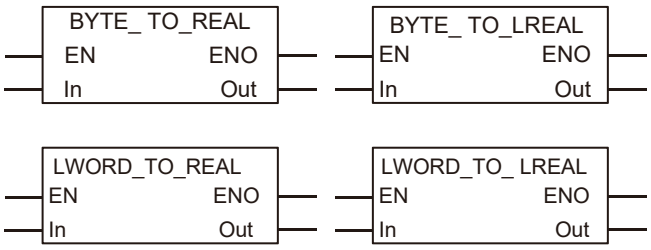
- Дані бітового рядка перетворюються на дані цілого числа, як показано в наведеній нижче таблиці.

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
BYTE	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	-128~-1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
WORD	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	-128~-1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	-32768~-1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
DWORD	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767
		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
LWORD	USINT	16#****_****_****_**00~ 16#****_****_****_**FF	0~255
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647
		16#****_****_8000_0000~ 16#***_*****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0

■ **Бітовий рядок до дійсного числа**

➢ Дані бітового рядка можна перетворити на дані дійсного числа. Деякі інструкції наведено нижче.



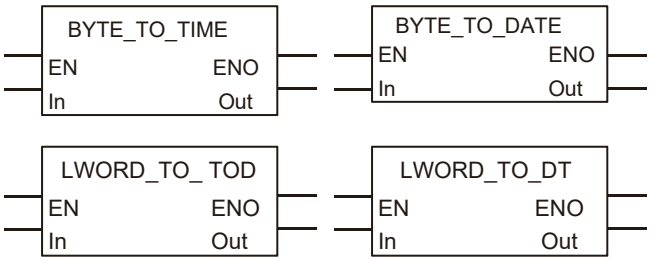
➢ Дані бітового рядка перетворюються на дані дійсного числа, як показано в наступній таблиці.

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
BYTE	REAL	16#00~16#FF	0~2,55e+2
	LREAL	16#00~16#FF	0~2,55e+2
WORD	REAL	16#0000~16#FFFF	0~6,5535e+4
	LREAL	16#0000~16#FFFF	0~6,5535e+4
DWORD	REAL	16#0000_0000~ 16#FFFF_FFFF	0~4,294967e+9
	LREAL	16#0000_0000~ 16#FFFF_FFFF	0~4,294967295e+9

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
LWORD	REAL	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~1,844674e+19
	LREAL	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	0~1,84467440737095e+19

■ **Рядок бітів для часу та дати**

➢ Дані бітового рядка можна перетворити на дані часу або дати. Деякі інструкції наведено нижче.



Правило для перетворення даних бітового рядка в дані часу або дати таке ж, як і для перетворення даних бітового рядка в дані без знаку.

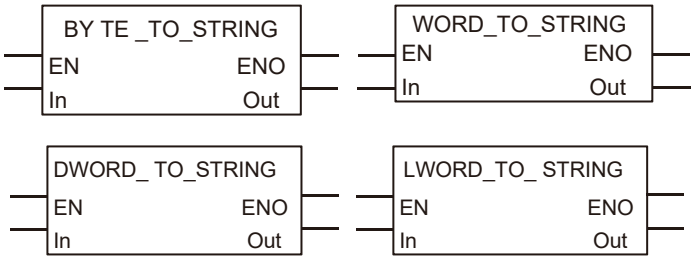
➢ Дані бітового рядка перетворюються на дані часу та дати, як показано в таблиці нижче.

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
BYTE	TIME	16#00~16#FF	T#0ns~T#255ns
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0,255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970- 1-1-0:4:15
WORD	TIME	16#0000~16#FFFF	T#0ns~T#65us535ns
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970- 1-1-18:12:15
DWORD	TIME	16#0000_0000~16#FFFF_FFFF	T#0ns~ T#4s294ms967us295ns
	DATE	16#0000_0000~16#FFFF_FFFF	D#1970-1-1~D#2016-2-7
	TOD	16#0000_0000~16#0526_5BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#0526_5C00~16#0A4C_B7FF	
		…….	
		16#FC57_9C00~16#FFFF_FFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#0000_0000~16#FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2016- 2-7-6:28:15
LWORD	TIME	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	T#213503d23h34m33s709ms551us615ns
	DATE	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	D#1970-1-1~D#2016-2-7
	TOD	16#****_****_0000_0000~ 16#****_****_0A4C_B7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#****_****_0526_5C00~	

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
		16#****_****_0A4C_B7FF	
		……	
		16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15

■ **Бітовий рядок до рядка**

➢ Дані рядка бітів можна перетворити на дані рядка. Деякі інструкції наведено нижче.



➢ Дані рядка бітів перетворюються на дані рядка, як показано в наведеній нижче таблиці.

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
BYTE	STRING	16#00~16#FF	'00'~'FF'
WORD	STRING	16#0000~16#FFFF	"0000" ~ "FFFF"
DWORD	STRING	16#0000_0000~16#FFFF_FFFF	'00000000' ~ 'FFFFFFFF'
LWORD	STRING	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	'0000000000000000'~ 'FFFFFFFFFFFFFFFF'

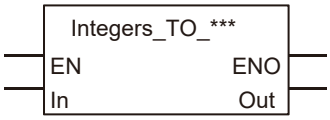
Коли бітові рядкові дані перетворюються на рядкові дані, довжина вихідних рядкових даних має відповідати довжині вхідного параметра. Наприклад, під час використання інструкції BYTE_TO_STRING вихідні рядкові дані повинні містити більше 2 символів. Інакше під час компіляції програмного забезпечення виникне помилка.

● **Застереження щодо правильного використання**

Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

8.13.3 Цілі числа_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції Integers_TO_*** перетворюють цілі числа в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In						●	●	●	●	●	●	●	●							
Out	Тип даних *Out* має збігатися з «***» назви інструкції.																			

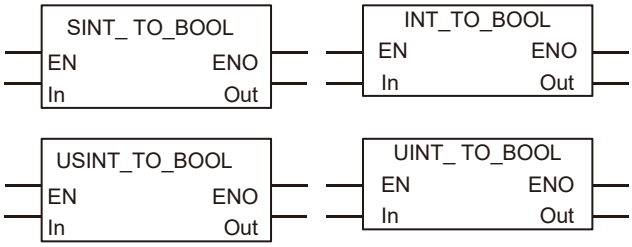
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ **Ціле число до BOOL**

➢ Деякі інструкції наведено нижче.

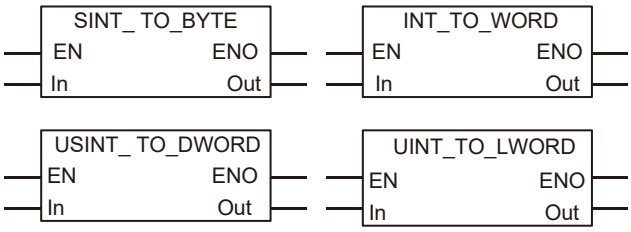


- Цілі дані перетворюються на логічні дані, як показано в таблиці нижче. Якщо значення Integer дорівнює 0, результатом перетворення є FALSE. Якщо не 0, результат TRUE. Докладніше про правило перетворення див. у таблиці нижче.

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
USINT	BOOL	0	FALSE
		1~255	TRUE
UINT	BOOL	0	FALSE
		1~65535	TRUE
UDINT	BOOL	0	FALSE
		1~4294967295	TRUE
ULINT	BOOL	0	FALSE
		1~18446744073709551645	TRUE
SINT	BOOL	0	FALSE
		-128~-1, 1~127	TRUE
INT	BOOL	0	FALSE
		-32768~-1, 1~32767	TRUE
DINT	BOOL	0	FALSE
		-2147483648~-1, 1~2147483647	TRUE
LINT	BOOL	0	FALSE
		-9223372036854775808~-1, 1~9223372036854775807	TRUE

■ **Ціле число в бітовий рядок**

- Дані типу Integer можна перетворити на дані бітового рядка. Деякі інструкції наведено нижче.



Правило для перетворення цілочисельних даних у бітові дані таке ж, як і для перетворення бітових даних у бітові дані. Додаткову інформацію див. у розділі 8.13.2.

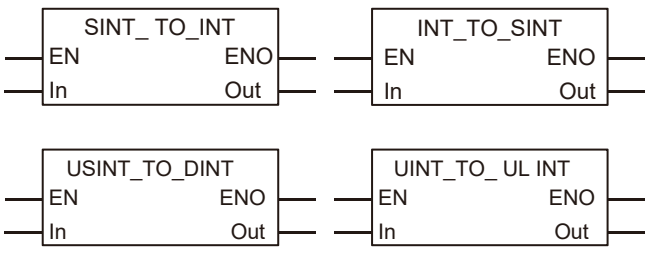
- Цілочисельні дані перетворюються на бітові дані, як показано в наведеній нижче таблиці.

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
USINT	BYTE	16#00~16#FF	16#00~16#FF
	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~ 16#0000_0000_0000_00FF
UINT	BYTE	16#**00~16#**FF	16#00~16#FF
	WORD	16#0000~16#FFFF	16#0000~16#FFFF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~ 16#0000_0000_0000_FFFF

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
UDINT	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	DWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
ULINT	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF
SINT	BYTE	16#00~16#FF	16#00~16#FF
	WORD	16#00~16#FF	16#0000~16#00FF
	DWORD	16#00~16#FF	16#0000_0000~16#0000_00FF
	LWORD	16#00~16#FF	16#0000_0000_0000_0000~ 16#0000_0000_0000_00FF
INT	BYTE	16#**00~16#**FF	16#00~16#FF
	WORD	16#0000~16#FFFF	16#0000~16#FFFF
	DWORD	16#0000~16#FFFF	16#0000_0000~16#0000_FFFF
	LWORD	16#0000~16#FFFF	16#0000_0000_0000_0000~ 16#0000_0000_0000_FFFF
DINT	BYTE	16#****_**00~16#****_**FF	16#00~16#FF
	WORD	16#****_0000~16#****_FFFF	16#0000~16#FFFF
	DWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000~16#FFFF_FFFF	16#0000_0000_0000_0000~ 16#0000_0000_FFFF_FFFF
LINT	BYTE	16#****_****_****_**00~ 16#****_****_****_**FF	16#00~16#FF
	WORD	16#****_****_****_0000~ 16#****_****_****_FFFF	16#0000~16#FFFF
	DWORD	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	16#0000_0000~16#FFFF_FFFF
	LWORD	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	16#0000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF

■ **Ціле число до цілого**

➢ Дані Integer можна перетворити на дані Integer. Деякі інструкції наведено нижче.



1. Правило для перетворення цілочисельних даних у цілочисельні дані таке ж, як і для перетворення бітових рядкових даних у бітові дані.
2. Дані меншої довжини перетворюються на дані більшої довжини шляхом запису значень усіх бітів даних меншої довжини у відповідні біти даних більшої довжини та встановлення значень решти бітів даних більшої довжини на 0.
3. Дані більшої довжини перетворюються на дані меншої довжини шляхом перегляду значень усіх бітів даних меншої довжини на значення відповідних бітів даних більшої довжини та значень решти бітів більшої довжини. дані про довжину не перетворюються і не впливають на перетворення.
4. Якщо довжини двох даних для перетворення рівні, усі значення всіх бітів *In* копіюються та вставляються у відповідні біти *Out* .

➢ Дані бітового рядка перетворюються на дані цілого числа, як показано в наведеній нижче таблиці.

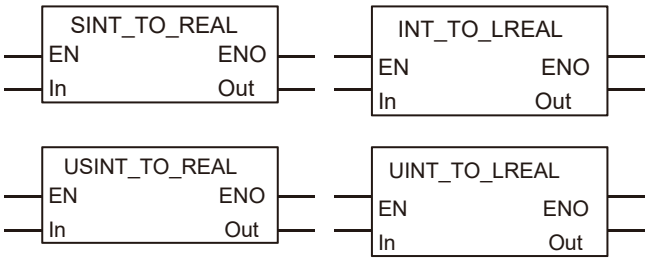
Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
USINT	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	- 128~ - 1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
UINT	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	- 128~ -1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	- 32768~ -1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
UDINT	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767
		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
ULINT	USINT	16#****_****_****_**00~ 16#****_****_****_**FF	0~255
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647
		16#****_****_8000_0000~ 16#****_****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0
SINT	USINT	16#00~16#FF	0~255
	UINT	16#00~16#FF	0~255
	UDINT	16#00~16#FF	0~255
	ULINT	16#00~16#FF	0~255
	SINT	16#00~16#7F	0~127
		16#80~16#FF	-128~-1
	INT	16#00~16#FF	0~255
	DINT	16#00~16#FF	0~255
	LINT	16#00~16#FF	0~255
INT	USINT	16#**00~16#**FF	0~255
	UINT	16#0000~16#FFFF	0~65535
	UDINT	16#0000~16#FFFF	0~65535
	ULINT	16#0000~16#FFFF	0~65535
	SINT	16#**00~16#**7F	0~127
		16#**80~16#**FF	-128~-1
	INT	16#0000~16#7FFF	0~32767
		16#8000~16#FFFF	-32768~-1
	DINT	16#0000~16#FFFF	0~65535
	LINT	16#0000~16#FFFF	0~65535
DINT	USINT	16#****_**00~16#****_**FF	0~255
	UINT	16#****_0000~16#****_FFFF	0~65535
	UDINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
	SINT	16#****_**00~16#****_**7F	0~127
		16#****_**80~16#****_**FF	-128~-1
	INT	16#****_0000~16#****_7FFF	0~32767

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
DINT		16#****_8000~16#****_FFFF	-32768~-1
	DINT	16#0000_0000~16#7FFF_FFFF	0~2147483647
		16#8000_0000~16#FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000~16#FFFF_FFFF	0~4294967295
LINT	USINT	16#****_****_****_**00~ 16#****_****_****_**FF	0~255
	UINT	16#****_****_****_0000~ 16#****_****_****_FFFF	0~65535
	UDINT	16#****_****_0000_0000~ 16#****_****_FFFF_FFFF	0~4294967295
	ULINT	16#0000_0000_0000_0000~ 16# FFFF_FFFF_FFFF_FFFF	0~18446744073709551645
	SINT	16#****_****_****_**00~ 16#****_****_****_**7F	0~127
		16#****_****_****_**80~ 16#****_****_****_**FF	-128~-1
	INT	16#****_****_****_0000~ 16#****_****_****_7FFF	0~32767
		16#****_****_****_8000~ 16#****_****_****_FFFF	-32768~-1
	DINT	16#****_****_0000_0000~ 16#****_****_7FFF_FFFF	0~2147483647
		16#****_****_8000_0000~ 16#****_****_FFFF_FFFF	-2147483648~-1
	LINT	16#0000_0000_0000_0000~ 16#7FFF_FFFF_FFFF_FFFF	0~9223372036854775807
		16#8000_0000_0000_0000~ 16#FFFF_FFFF_FFFF_FFFF	-9223372036854775808~0

■ **Ціле до дійсного числа**

➢ Цілі дані можна перетворити на дані дійсного числа. Деякі інструкції наведено нижче.

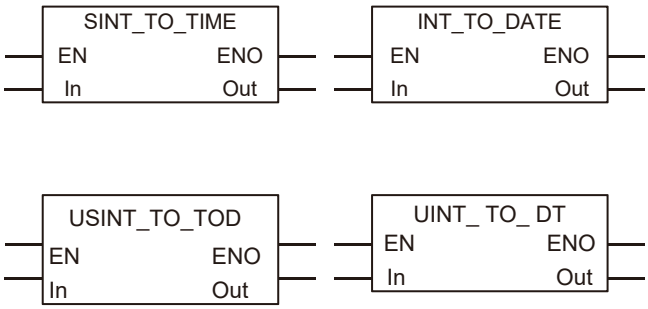


- Цілі дані перетворюються на дані дійсного числа, як показано в наступній таблиці.

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
USINT	REAL	0~255	0~2,55e+2
	LREAL	0~255	0~2,55e+2
UINT	REAL	0~65535	0~6,5535e+4
	LREAL	0~65535	0~6,5535e+4
UDINT	REAL	0~4294967295	0~4,294967e+9
	LREAL	0~4294967295	0~4,294967295e+9
ULINT	REAL	0~18446744073709551615	0~1,844674e+19
	LREAL	0~18446744073709551615	0~1,84467440737095e+19
SINT	REAL	-128~127	-1,28e+2~1,27e+2
	LREAL	-128~127	-1,28e+2~1,27e+2
INT	REAL	-32768~32767	-3,2768e+4~3,2767e+4
	LREAL	-32768~32767	-3,2768e+4~3,2767e+4
DINT	REAL	-2147483648~2147483647	-2,147483e+9~2,147483e+9
	LREAL	-2147483648~2147483647	-2,147483e+9~2,147483e+9
LINT	REAL	-9223372036854775808~ 9223372036854775807	-9,223372e+18~9,223372e+18
	LREAL	-9223372036854775808~ 9223372036854775807	-9,22337203685477e+18~ 9,22337203685477e+18

■ **Ціле число до часу або дати**

- Дані типу Integer перетворюються на дані часу або дати, а деякі інструкції наведено нижче.



Правило для перетворення цілих даних у дані про час або дату таке ж, як і для перетворення цілих даних у беззнакові цілі дані.

- Дані Integer перетворюються на дані часу або дати, як показано в наведеній нижче таблиці.

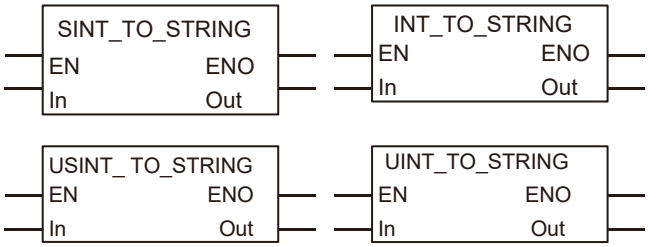
Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
USINT	TIME	16#00~16#FF	T#0ns~T#255ns
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0,255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15

Тип даних		**Значення *In* відповідає значенню *Out***	
In	Out	In	Out
UINT	TIME	16#0000~16#FFFF	T#0ns~T#65us535ns
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
UDINT	TIME	16#00000000~16#FFFFFFFF	T#0ns~ T#4s294ms967us295ns
	DATE	16#00000000~16#FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#00000000~16#05265BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#05265C00~16#0A4CB7FF	
		…….	
		16#FC579C00~16#FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#00000000~16#FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
ULINT	TIME	16#0000000000000000~ 16# FFFFFFFFFFFFFFFF	T#213503d23h34m33s709ms5 51us615ns
	DATE	16#********00000000~ 16#********FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#********00000000~ 16#********0A4CB7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#********05265C00~ 16#********0A4CB7FF	
		……	
		16#********00000000~ 16#********FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#********00000000~ 16#********FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15
SINT	TIME	16#00~16#FF	T#0ns~T#255ns
	DATE	16#00~16#FF	D#1970-1-1
	TOD	16#00~16#FF	TOD#0:0:0~ TOD#0:0:0,255
	DT	16#00~16#FF	DT#1970-1-1-0:0:0~ DT#1970-1-1-0:4:15
INT	TIME	16#0000~16#FFFF	T#0ns~T#65us535ns
	DATE	16#0000~16#FFFF	D#1970-1-1
	TOD	16#0000~16#FFFF	TOD#0:0:0~ TOD#0:1:5.535
	DT	16#0000~16#FFFF	DT#1970-1-1-0:0:0~ DT#1970-1-1-18:12:15
DINT	TIME	16#00000000~16#FFFFFFFF	T#0ns~ T#4s294ms967us295ns
	DATE	16#00000000~16#FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#00000000~16#05265BFF	TOD#0:0:0~ TOD#23:59:59.999
		16#05265C00~16#0A4CB7FF	
		…….	
		16#FC579C00~16#FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#00000000~16#FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
LINT	TIME	16#0000000000000000~ 16# FFFFFFFFFFFFFFFF	T#213503d23h34m33s709ms551us615ns
	DATE	16#********00000000~ 16#********FFFFFFFF	D#1970-1-1~D#2016-2-7
	TOD	16#********00000000~ 16#********0A4CB7FF	TOD#0:0:0~ TOD#23:59:59.999
		16#********05265C00~ 16#********0A4CB7FF	
		……	
		16#********00000000~ 16#********FFFFFFFF	TOD#0:0:0~ TOD#17:2:47.295
	DT	16#********00000000~ 16#********FFFFFFFF	DT#1970-1-1-0:0:0~ DT#2016-2-7-6:28:15

■ **Ціле число в рядок**

➢ Дані Integer можна перетворити на дані String, і деякі інструкції наведено нижче.



➢ Дані Integer перетворюються на дані String, як показано в наступній таблиці.

Тип даних		Значення *In* відповідає значенню *Out*	
In	Out	In	Out
USINT	STRING	0~255	'0'~'255'
UINT	STRING	0~65535	'0'~'65535'
UDINT	STRING	0~4294967295	'0'~'4294967295'
ULINT	STRING	0~18446744073709551615	'0'~'18446744073709551615'
SINT	STRING	-128~127	'-128' ~ '127'
INT	STRING	-32768~32767	'-32768'~'32767'
DINT	STRING	-2147483648~2147483647	'-2147483648'~'2147483647'
LINT	STRING	-9223372036854775808~ 9223372036854775807	'-9223372036854775808'~ '9223372036854775807'

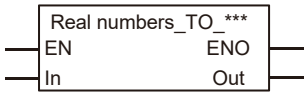
Коли бітові рядкові дані перетворюються на рядкові дані, довжина вихідних рядкових даних має відповідати довжині вхідного параметра.

● **Застереження щодо правильного використання**

Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

8.13.4 Реальні числа_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції Real numbers_TO_*** перетворюють дійсні числа в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In														●	●					
Out	Тип даних *Out* має збігатися з «***» назви інструкції.																			

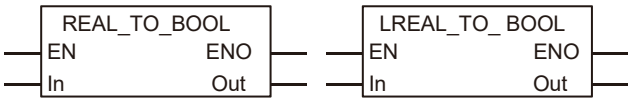
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ **Дійсне число до BOOL**

➢ Відповідні інструкції:



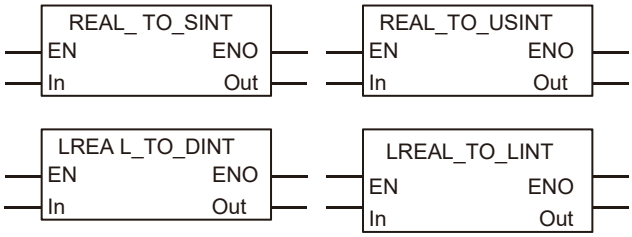
➢ Дійсні числа перетворюються на булеві дані, як показано в наведеній нижче таблиці. Якщо дійсне число дорівнює 0, результатом перетворення є ЛОЖЬ. Якщо дійсне число не дорівнює 0, результатом перетворення є ІСТИНА. Докладніше про правило див. у таблиці нижче.

Тип даних		Значення *In* відповідає значенню *Out*	
In	**Out**	**In**	**Out**
REAL	BOOL	-3,402823E+38~-1,175495E-38	TRUE
		0	FALSE

Тип даних		Значення *In* відповідає значенню *Out*	
In	**Out**	**In**	**Out**
		1,175495E-38~3,402823E+38	TRUE
LREAL	BOOL	-1,79769313486231E+308~ -2,22507385850721E-308	TRUE
		0	FALSE
		2.22507385850721E-308~ 1,79769313486231E+308	TRUE

■ **Дійсне число до цілого**

➢ Дійсні числа можна перетворити на цілі. Деякі інструкції наведено нижче.



➢ Для перетворення дійсного числа в ціле є два випадки, у яких дробова частина скорочується та округлюється, як зазначено нижче.

Випадок 1: якщо перше цифрове число дробової частини менше 5, дробову частину буде скорочено, а ціла частина не зміниться.

Випадок 2: якщо перше цифрове число дробової частини більше або дорівнює 5, дробову частину буде скорочено, а до цілої частини буде додано 1.

Вхідне значення		Вихідний результат	
		Тип даних	**Вихідне значення**
Випадок 1	1.36	SINT	1
		USINT	1
	-2,4	SINT	-2
		USINT	254
Випадок 2	1.6	SINT	2
		USINT	2
	-2,6	SINT	-3
		USINT	253

Примітка:

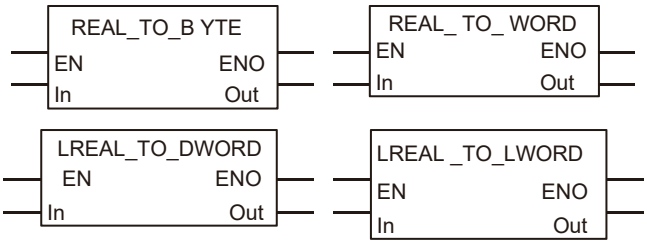
Для перетворення дійсного числа в ціле існує два випадки для значення дійсного числа.

1. Якщо кількість введених цифр дійсного числа перевищує допустиму, результатом буде невпевнене значення. Будь ласка, встановіть обмеження в програмі користувача, щоб отримати правильне значення.
   Наприклад: Тоді вхідне значення дорівнює 123456789 і кількість його цифр перевищує набір обмеження 7. Цифри, які виходять за межі, є ненормальними. Тоді вихідне значення дорівнює 1234567 ***92*** .

2. Якщо кількість введених цифр не перевищує встановленого ліміту, результат обчислюється за правилом перетворення.

■ **Рядок дійсне число в біт**

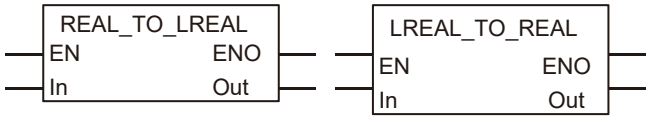
➢ Реальні числа можна перетворити на бітові рядки. Деякі інструкції наведено нижче.



Правило перетворення дійсних чисел у бітові рядки таке ж, як і для перетворення дійсних чисел у цілі без знаку.

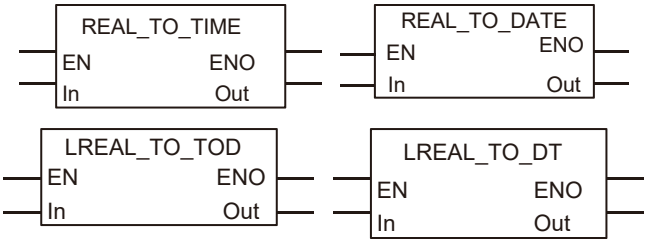
■ **Реальне число до дійсного числа**

➢ Дійсні числа можна перетворити на дійсні числа. Деякі інструкції наведено нижче.



■ **Справжнє число до часу або дати**

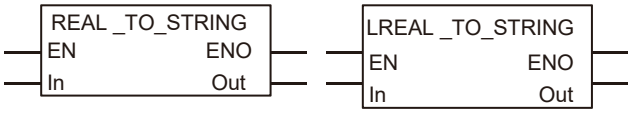
➢ Реальні числа можна перетворити на час або дату. Деякі інструкції наведено нижче.



Для перетворення дійсного числа в час або дату дійсне число спочатку перетворюється на ціле число, а потім ціле число перетворюється на час або дату. Для відповідного вмісту зверніться до перетворення дійсного числа в ціле число та перетворення цілого числа в час або дату.

■ **Дійсне число до рядка**

➢ Реальні числа можна перетворити на рядки. Деякі інструкції наведено нижче.



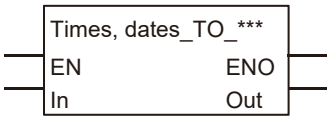
Правило для перетворення дійсного числа в рядок таке ж, як і для перетворення цілого числа в рядок. Додаткову інформацію див. у розділі 8.13.3.

● **Застереження щодо правильного використання**

Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

8.13.5 Час, дати_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції Times, dates_TO_*** перетворюють дані про час або дату в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																●	●	●	●	
Out	Тип даних *Out* має збігатися з «***» назви інструкції.																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

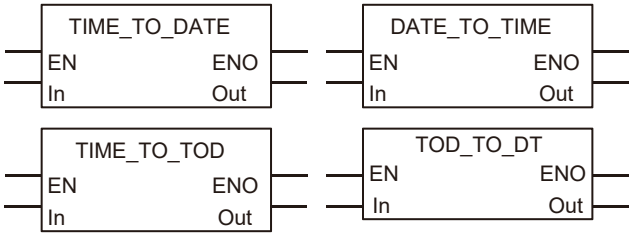
● **Пояснення функції**

■ **Час і дата для Bool, бітового рядка, цілого числа, дійсного числа та рядка**

Правило для перетворення часу та дати в логічний тип, рядок бітів, ціле число, дійсне число та рядок таке ж, як і для перетворення цілого числа без знаку в логічний рядок, рядок бітів, ціле число, дійсне число та рядок. Додаткову інформацію див. у розділі 8.13.5.

■ **Час і дата до часу і дати**

➢ Дані про час і дату можна конвертувати один в одного. Деякі інструкції наведено нижче.



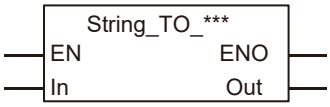
Правило для перетворення даних про час і дату в дані про час і дату таке ж, як і для перетворення цілих чисел без знака в цілі без знака. Під час перетворення одиниці повинні бути однорідними. Одиницею ЧАСУ є нс (наносекунда), а іншими одиницями є мс (мілісекунда).

- **Застереження щодо правильного використання**

  Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

8.13.6 Рядки_TO_***

FB/FC	Пояснення	Застосовна модель
FC	Інструкції Strings_TO_*** перетворюють дані String в дані базових типів даних. «***» може бути будь-яким базовим типом даних.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Дані для перетворення	Вхід	Дані для перетворення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат перетворення	Вихід	Результат перетворення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
Out	Тип даних *Out* має збігатися з «***» назви інструкції.																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ **Рядок до Bool**

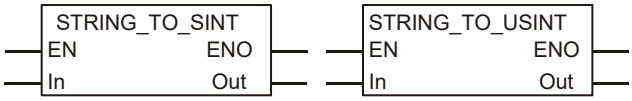
➢ Відповідні інструкції:

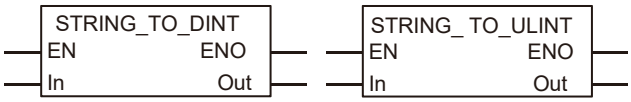


Правило для перетворення String-to-Bool полягає в тому, що вихідне значення Bool є TRUE, лише якщо значення рядка TRUE або true. В іншому випадку результат буде FALSE.

■ **Рядок до цілого**

➢ Рядки можна перетворювати на цілі числа. Деякі інструкції наведено нижче.



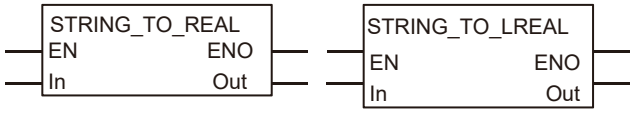


- Для перетворення рядка в ціле рядок має бути цілим значенням, таким як «123», «- 123» і «+123». Рядок на зразок "M123" не можна перетворювати на ціле число. Приклади перетворення наведено в наступній таблиці.

Вхідне значення	Вихідний результат	
	Тип даних	Вихідне значення
"123"	SINT	123
"+123"	SINT	123
'-123'	SINT	-123
"M123"	SINT	Перетворення заборонено, і початкове значення вихідної змінної зберігається.

■ **Рядок до дійсного числа**

- Рядки можна перетворювати на дійсні числа. Деякі інструкції наведено нижче.

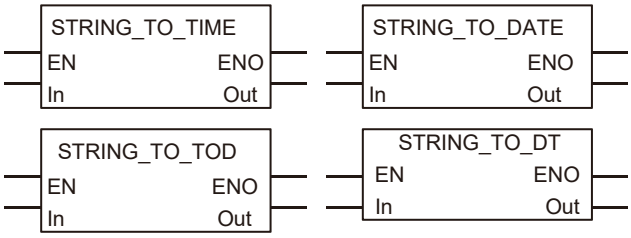


- Для перетворення рядка в дійсне число потрібно, щоб рядок був значенням дійсного числа, наприклад «123», «-123,123» і «1,23e+5». Приклади перетворення наведено в

Вхідне значення	Вихідний результат	
	Тип даних	Вихідне значення
"123"	REAL	123
'-123,123'	REAL	-123,123
'1,23e+5'	REAL	-1,23e+5
"M123.123"	REAL	Перетворення заборонено, і початкове значення вихідної змінної зберігається.

■ **Рядок часу або дати**

- Рядки можна перетворити на час і дату. Деякі інструкції наведено нижче.



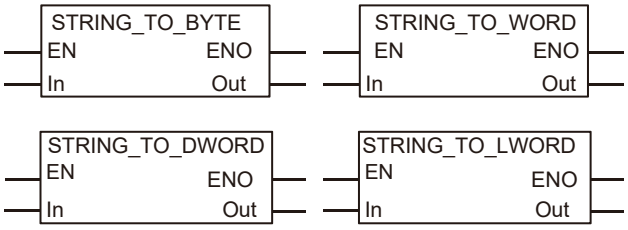
- Для перетворення рядка в час або дату рядок повинен представляти значення часу або дати, наприклад «T#1ns», «D#1970-1-1», «TOD#0:0:0» і "DT#1970-1-1-0:0:0". Приклади перетворення наведено в наступній таблиці.

Вхідне значення	Вихідний результат	
	Тип даних	Вихідне значення
"T#1ns"	TIME	T#1ns
"D#1970-1-1"	DATE	D#1970-1-1

Вхідне значення	Вихідний результат	
	Тип даних	Вихідне значення
'TOD#0:0:0'	TOD	TOD#0:0:0
'DT#1970-1-1-0:0:0'	DT	DT#1970-1-1-0:0:0

■ **Рядок до рядка біта**

➢ Рядки можна перетворити на бітові рядки. Деякі інструкції наведено нижче.



Правило для перетворення рядка в біт таке ж, як і для перетворення рядка в ціле число.

● **Застереження щодо правильного використання**

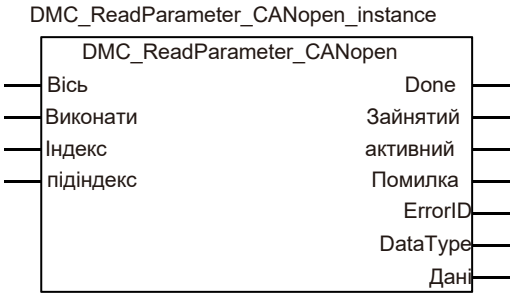
Вхідну змінну не можна пропускати. Під час компіляції програмного забезпечення виникне помилка, якщо вхідну змінну опущено. Але вихідну змінну можна опускати.

8.14 Інструкції зв’язку CANopen

8.14.1 Інструкції зв’язку

8.14.1.1 DMC_ReadParameter_CANopen

FB/FC	Пояснення	Застосовна модель
FB	DMC_ReadParameter_CANopen використовується для читання значення параметра підлеглого пристрою.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Вісь	Вкажіть підлеглий пристрій, яким має керувати інструкція	USINT	1~127 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
Індекс	Індекс параметра для читання	UINT	0	Коли *Execute* змінюється з FALSE на TRUE
Підіндекс	Підіндекс параметра для читання	USINT	0	Коли *Execute* змінюється з FALSE на TRUE

- **Вихідні**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Активний	ІСТИНА, коли вісь керується.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли є помилка.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	
Тип даних	Тип даних параметра читання. 1: Байт, 2: Слово, 4: Подвійне слово	USINT	
Дані	Значення параметра, яке було прочитано	UDINT	

■ **Індекс та підіндекс підлеглого параметра, який потрібно зчитати:**

1. Параметр, визначений користувачем, є параметром сервоприводу, який потрібно зчитати. Довжина даних визначається користувачами відповідно до типу даних параметра read. Довжина даних байтового параметра дорівнює 1, довжина даних параметра слова — 2, а довжина даних параметра подвійного слова — 4.

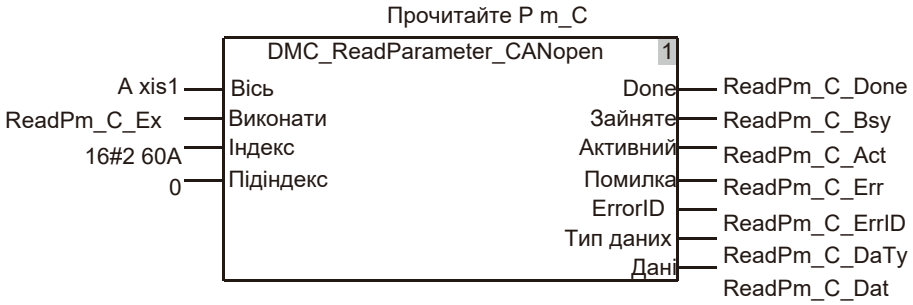
Метод розрахунку індексу та субіндексу параметра сервоприводу:
Індекс = значення параметра сервоприводу (Hex) + 2000 (Hex)
Субіндекс= 0.

приклад:

Індекс та субіндекс параметра сервоприводу P6-10 [2000 + 060A （ шістнадцяткове значення P6-10)] 260A та 0 відповідно.

➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
ReadPm_C	DMC_ReadParameter_CANopen	
ReadPm_C_Ex	BOOL	FALSE
ReadPm_C_Done	BOOL	
ReadPm_C_Bsy	BOOL	
ReadPm_C_Act	BOOL	
ReadPm_C_Err	BOOL	
ReadPm_C_ErrID	WORD	
ReadPm_C_DaTy	USINT	
ReadPm_C_Dat	UDINT	



2. Індекси та підіндекси інших підлеглих параметрів див. у посібнику CANopen для підлеглих пристроїв.

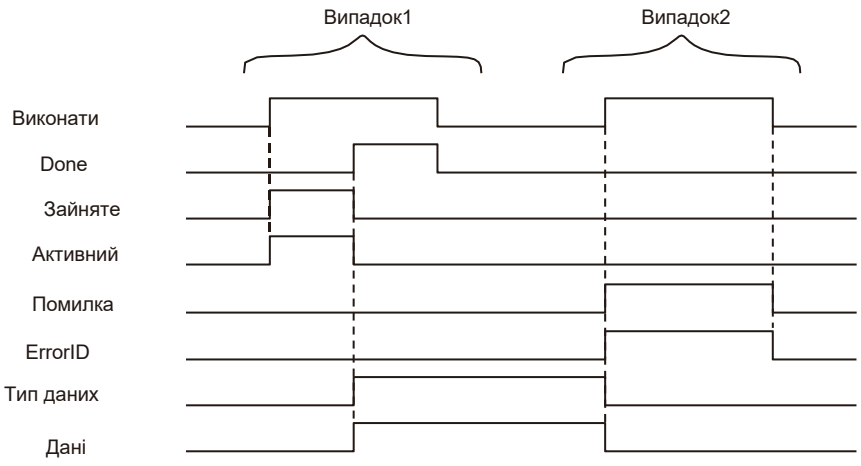
8

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли зчитування вмісту параметра завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Зайняте	◆ Коли *Execute* змінюється на TRUE	◆ Коли *помилка* змінюється на TRUE ◆ Коли *Done* змінюється з FALSE на TRUE
Активний	◆ Коли ведений починає керуватися інструкцією	◆ Коли помилка змінюється на TRUE ◆ Коли Done змінюється з FALSE на TRUE

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● Графік часу Вихід



Випадок 1 : *«Зайнятий»* і *«Активний»* змінюється на «ІСТИНА», коли *«Виконати»* змінюється з «ХИБНІСТЬ» на «ІСТИННО» і через один період, *«Готово»* змінюється на «ІСТИНА», а *тип даних* і *дані* показують відповідні дані. Коли *Done* змінюється на TRUE, *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE, а *тип даних* і *дані* зберігають вихідні значення.

Випадок 2 : перед виконанням DMC_ReadParameter_CANopen значення вхідного параметра, наприклад номер осі: 0, є недопустимим. Коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється з FALSE на TRUE, значення Datatype і Data очищаються до 0, а *ErrorID* показує відповідні коди помилок. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а вміст *ErrorID* очищається до 0.

● Функції

DMC_ReadParameter_CANopen використовується для читання значення параметра підлеглого пристрою. Користувачі можуть вказати індекс та піндекс параметра, який потрібно зчитати.

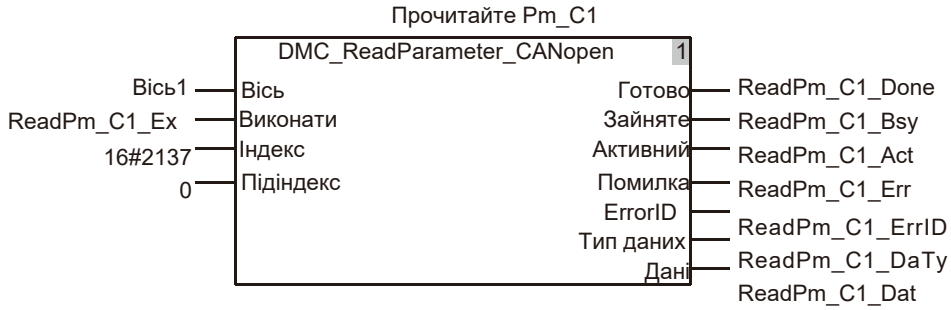
Приклад програмування

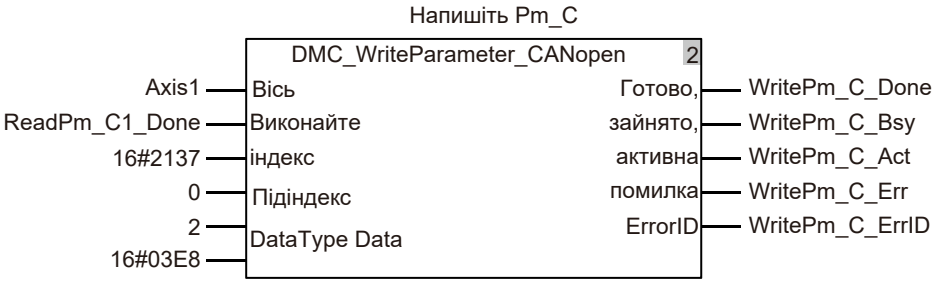
Нижче наведено приклад виконання інструкції DMC_ReadParameter_CANopen.

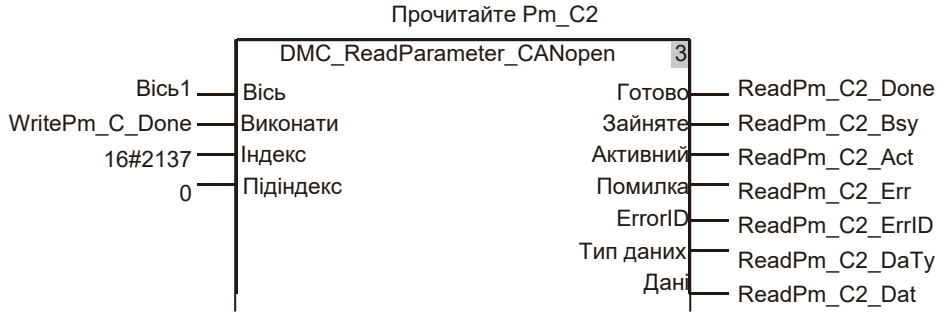
■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
Прочитайте Pm_C1	DMC_ReadParameter_CANopen	
Вісь1	USINT	1
ReadPm_C1_Ex	BOOL	TRUE
ReadPm_C1_Done	BOOL	TRUE
ReadPm_C1_Bsy	BOOL	FALSE
ReadPm_C1_Act	BOOL	FALSE
ReadPm_C1_Err	BOOL	FALSE
ReadPm_C1_ErrID	WORD	FALSE
ReadPm_C1_DaTy	USINT	2
ReadPm_C1_Dat	UDINT	5000
Напишіть Pm_C	DMC_WriteParameter_CANopen	

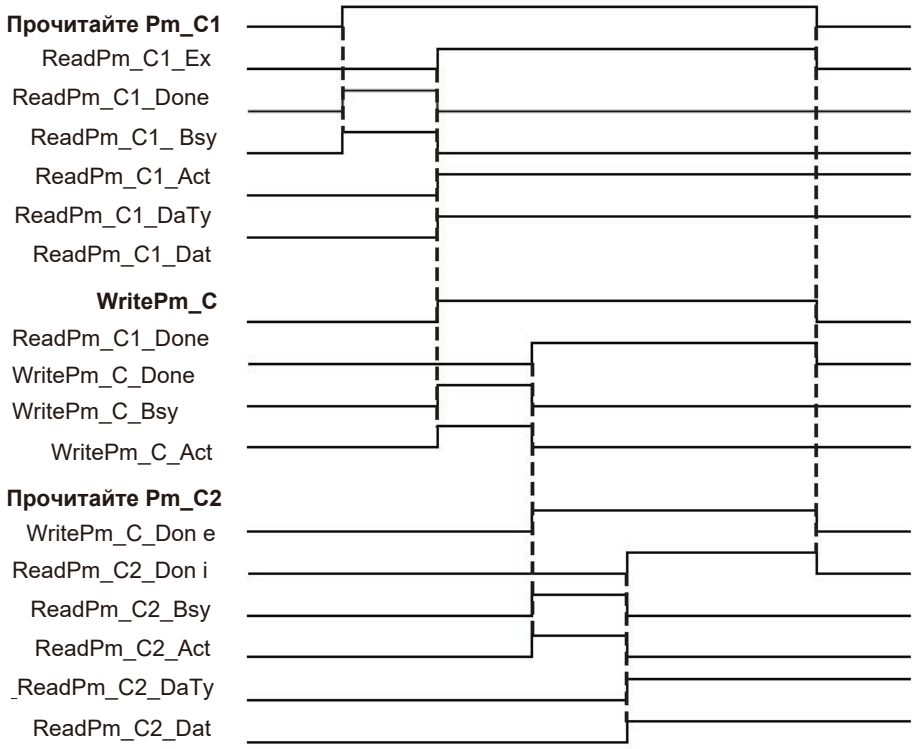
Ім'я змінної	Тип даних	Поточне значення
WritePm_C_Done	BOOL	TRUE
WritePm_C_Bsy	BOOL	FALSE
WritePm_C_Act	BOOL	FALSE
WritePm_C_Err	BOOL	FALSE
НапишітьPm_C_ErrID	WORD	FALSE
Прочитайте Pm_C2	DMC_ReadParameter_CANopen	
ReadPm_C2_Done	BOOL	TRUE
ReadPm_C2_Bsy	BOOL	FALSE
ReadPm_C2_Act	BOOL	FALSE
ReadPm_C2_Err	BOOL	FALSE
ReadPm_C2_ErrID	WORD	FALSE
ReadPm_C2_DaTy	USINT	2
ReadPm_C2_Dat	UDINT	1000







■ Графік часу



❖ Перший DMC_ReadParameter_CANopen починає виконуватися, коли ReadPm_C1_Ex змінюється з FALSE на TRUE. Коли виконання першого DMC_ReadParameter_CANopen завершено, ReadPm_C1_Done змінюється на TRUE, ReadPm_C1_DaTy = 2 і ReadPm_C1_Dat=5000.

Тобто вміст веденого параметра сервоприводу P1-55, який зчитується, становить 5000. (Максимальна швидкість сервоприводу обмежена 5000 об/хв.)

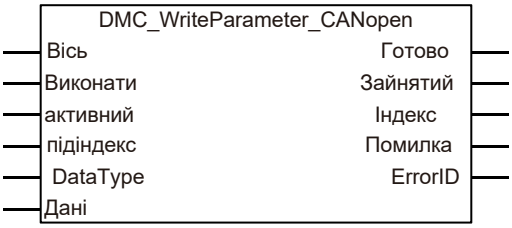
❖ Коли ReadPm_C1_Done змінюється з FALSE на TRUE, DMC_WriteParameter_CANopen починає виконуватися. Після завершення виконання інструкції DMC_WriteParameter_CANopen WritePm_C_Done змінюється на TRUE. Тобто 1000 записується як вміст параметра веденого сервоприводу P1-55. (Максимальна швидкість сервоприводу обмежена 1000 об/хв.)

❖ Другий DMC_ReadParameter_CANopen виконується, коли WritePm_C_Done змінюється з FALSE на TRUE. Коли виконання другого DMC_ReadParameter_CANopen завершено, ReadPm_C2_Done змінюється на TRUE, ReadPm_C2_DaTy = 2 і ReadPm_C2_Dat=1000. Тобто зчитований вміст параметра P1-55 веденого сервоприводу становить 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв.)

8.14.1.2 DMC_WriteParameter_CANopen

FB/FC	Пояснення	Застосовна модель
FB	DMC_WriteParameter_CANopen використовується для встановлення значення параметра підлеглого пристрою.	AS516E-B AS532EST-B AS564EST-B

DMC_WriteParameter_CANopen_instance



● **Вхідні**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Вісь	Вкажіть підлеглий пристрій, яким має керувати інструкція	USINT	1~127 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	-
Індекс	Індекс параметра, який встановлюється	UINT		
Підіндекс	Підіндекс встановленого параметра	USINT		
Тип даних	Тип даних параметра, який встановлюється 1 : байт, 2 : слово, 4 : подвійне слово.	USINT		
Дані	Значення вмісту параметра, який встановлюється	UDINT		

Примітки:

1. Значення *DataType* має вказувати тип даних параметра, який встановлено. Якщо заповнене значення неправильне, в інструкції виникне помилка.
2. Щоб отримати інформацію про метод обчислення індексу та субіндексу підлеглого параметра CANopen, зверніться до розділу «Вступ до параметрів осі» у розділі 9.

● **Вихідні параметри**

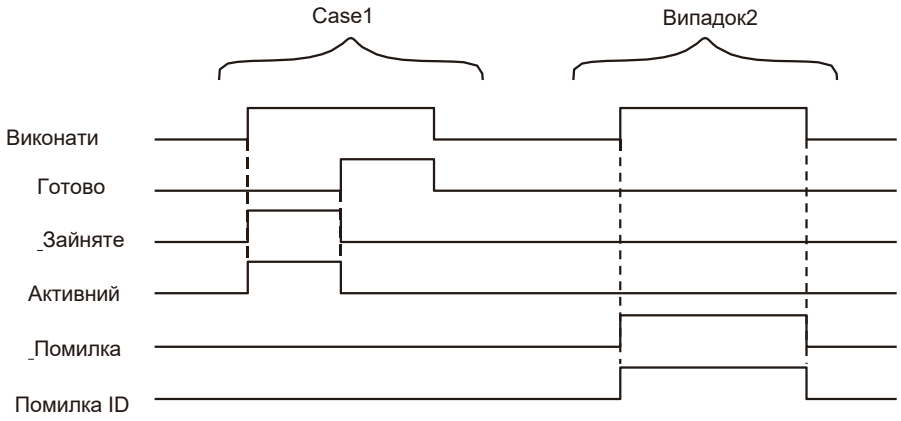
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Активний	ІСТИНА, коли вісь керується.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли є помилка.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли запис вмісту параметра завершено	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції
Зайняте	◆ Коли *Execute* змінюється на TRUE	◆ Коли *помилка* змінюється на TRUE ◆ Коли *Done* змінюється з FALSE на TRUE
Активний	◆ Коли ведений починає керуватися інструкцією	◆ Коли *помилка* змінюється на TRUE ◆ Коли *Done* змінюється з FALSE на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Графік часу**



Випадок 1 : *«Зайнятий»* і *«Активний»* змінюється на «ІСТИНА», коли *«Виконати»* змінюється з «ХИБНІСТЬ» на «ІСТИННО», а через один період « *Готово»* змінюється на «ІСТИННО». Коли *Done* змінюється на TRUE, *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

Випадок 2 : перед виконанням DMC_WriteParameter_CANopen значення вхідного параметра, наприклад номер осі: 0, є недопустимим. Після того як *Execute* змінюється з FALSE на TRUE, *Error* змінюється з FALSE на TRUE, а *ErrorID* відображає відповідні коди помилок. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а вміст *ErrorID* очищається до 0.

● **функція**

DMC_WriteParameter_CANopen використовується для встановлення значення параметра підлеглого пристрою. Користувачі можуть вказати індекс та підіндекс параметра, який потрібно встановити.

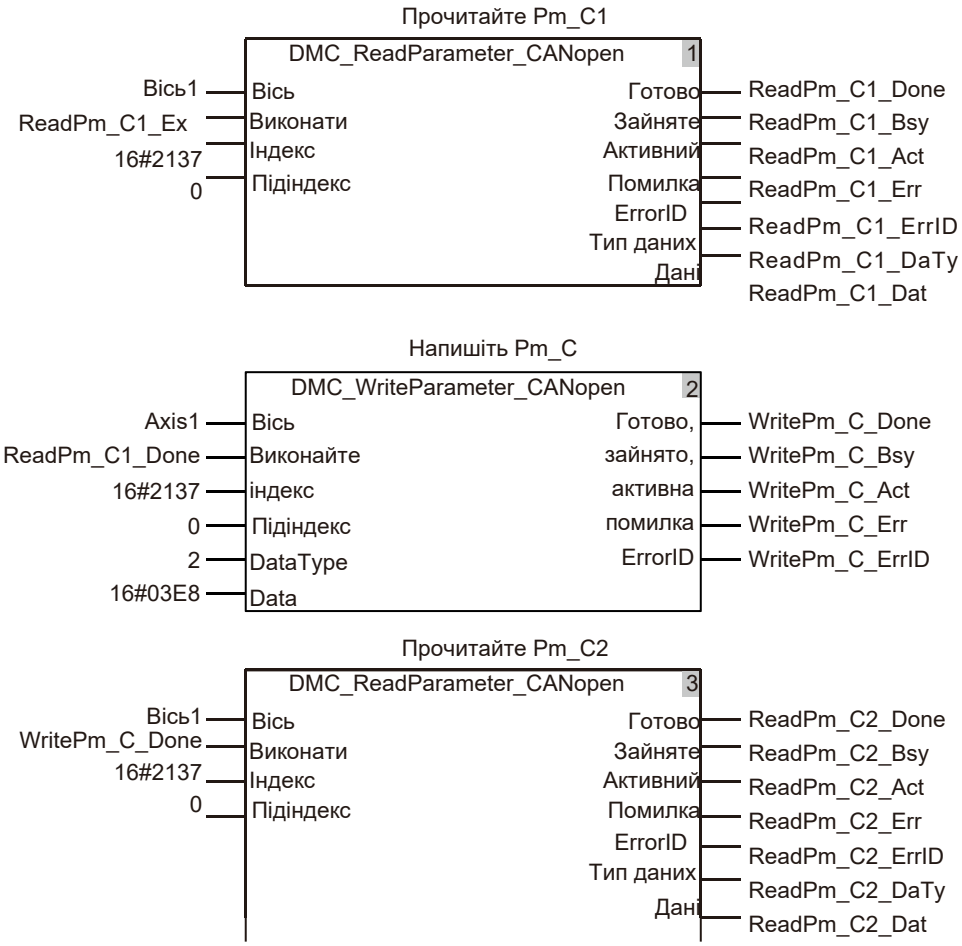
Приклад програмування

Нижче наведено приклад виконання однієї інструкції DMC_WriteParameter_CANopen.

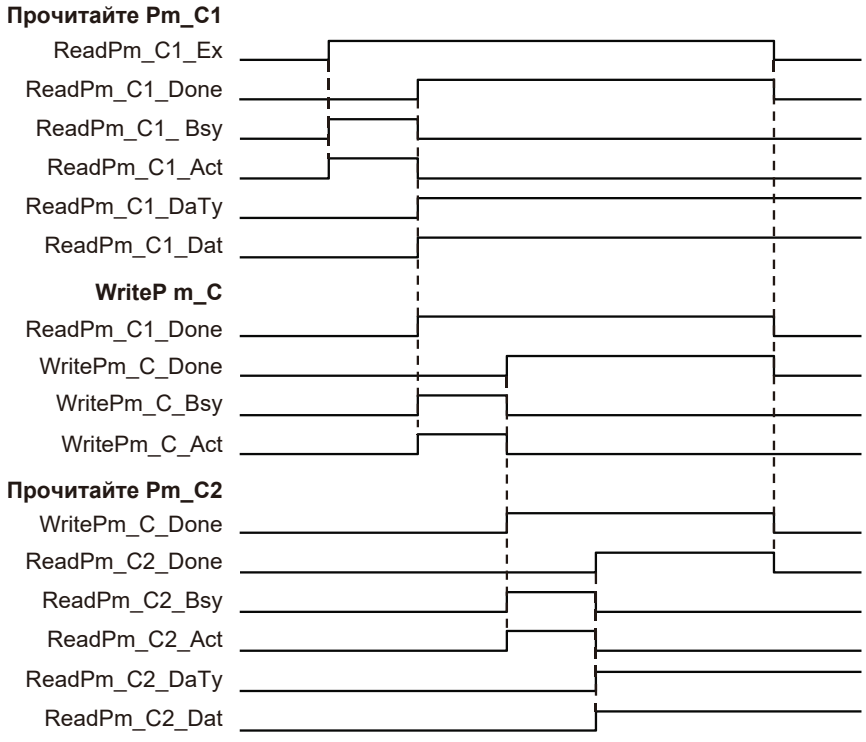
■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
Прочитайте Pm_C1	DMC_ReadParameter_CANopen	

Ім'я змінної	Тип даних	Початкове значення
Вісь1	USINT	1
ReadPm_C1_Ex	BOOL	TRUE
ReadPm_C1_Done	BOOL	TRUE
ReadPm_C1_Bsy	BOOL	FALSE
ReadPm_C1_Act	BOOL	FALSE
ReadPm_C1_Err	BOOL	FALSE
ReadPm_C1_ErrID	WORD	FALSE
ReadPm_C1_DaTy	USINT	2
ReadPm_C1_Dat	UDINT	5000
WritePm_C	DMC_WriteParameter_CANopen	
WritePm_C_Done	BOOL	TRUE
WritePm_C_Bsy	BOOL	FALSE
WritePm_C_Act	BOOL	FALSE
WritePm_C_Err	BOOL	FALSE
НапишітьPm_C_ErrID	WORD	FALSE
ПрочитайтеPm_C2	DMC_ReadParameter_CANopen	
ReadPm_C2_Done	BOOL	TRUE
ReadPm_C2_Bsy	BOOL	FALSE
ReadPm_C2_Act	BOOL	FALSE
ReadPm_C2_Err	BOOL	FALSE
ReadPm_C2_ErrID	WORD	FALSE
ReadPm_C2_DaTy	USINT	2
ReadPm_C2_Dat	UDINT	1000


Прочитайте Pm_C1
DMC_ReadParameter_CANopen
1
Вісь1
ReadPm_C1_Ex
16#2137
0
Вісь
Виконати
Індекс
Підіндекс
Готово
Зайняте
Активний
Помилка
ErrorID
Тип даних
Дані
ReadPm_C1_Done
ReadPm_C1_Bsy
ReadPm_C1_Act
ReadPm_C1_Err
ReadPm_C1_ErrID
ReadPm_C1_DaTy
ReadPm_C1_Dat
Напишіть Pm_C
DMC_WriteParameter_CANopen
2
Axis1
ReadPm_C1_Done
16#2137
0
2
16#03E8
Вісь
Виконайте
індекс
Підіндекс
DataType
Data
Готово,
зайнято,
активна
помилка
ErrorID
WritePm_C_Done
WritePm_C_Bsy
WritePm_C_Act
WritePm_C_Err
WritePm_C_ErrID
Прочитайте Pm_C2
DMC_ReadParameter_CANopen
3
Вісь1
WritePm_C_Done
16#2137
0
Вісь
Виконати
Індекс
Підіндекс
Готово
Зайняте
Активний
Помилка
ErrorID
Тип даних
Дані
ReadPm_C2_Done
ReadPm_C2_Bsy
ReadPm_C2_Act
ReadPm_C2_Err
ReadPm_C2_ErrID
ReadPm_C2_DaTy
ReadPm_C2_Dat

■ Графік часу


Прочитайте Pm_C1
ReadPm_C1_Ex
ReadPm_C1_Done
ReadPm_C1_ Bsy
ReadPm_C1_Act
ReadPm_C1_DaTy
ReadPm_C1_Dat
WriteP m_C
ReadPm_C1_Done
WritePm_C_Done
WritePm_C_Bsy
WritePm_C_Act
Прочитайте Pm_C2
WritePm_C_Done
ReadPm_C2_Done
ReadPm_C2_Bsy
ReadPm_C2_Act
ReadPm_C2_DaTy
ReadPm_C2_Dat

- Коли ReadPm_C1_Ex змінюється з FALSE на TRUE, перший DMC_ReadParameter_CANopen починає виконуватися. Після завершення виконання першого DMC_ReadParameter_CANopen ReadPm_C1_Done змінюється на TRUE, ReadPm_C1_DaTy =2 і ReadPm_C1_Dat=5000. Тобто вміст веденого параметра сервоприводу P1-55, який зчитується, становить 5000. (Максимальна швидкість сервоприводу обмежена 5000 об/хв.)

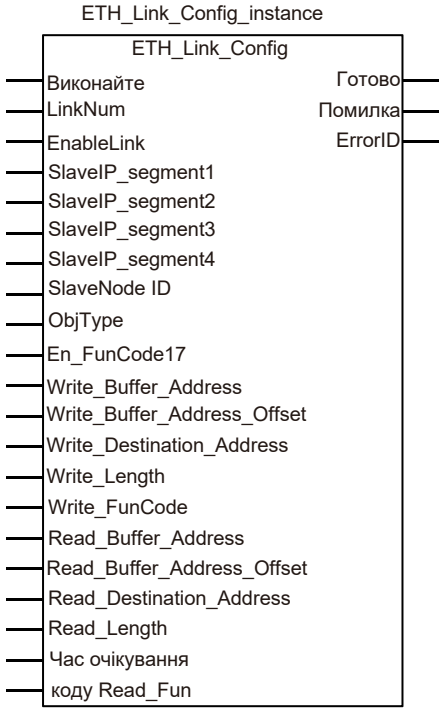
- Коли ReadPm_C1_Done змінюється з FALSE на TRUE, DMC_ WriteParameter Починається виконання інструкції _CANopen. Після завершення виконання інструкції DMC_WriteParameter_CANopen WritePm_C_Done змінюється на TRUE. Тобто записаний вміст веденого параметра сервоприводу P1-55 становить 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв.)

- Коли WritePm_C_Done змінюється з FALSE на TRUE, другий DMC_ ReadParameter Починається виконання інструкції _CANopen. Після виконання другої інструкції DMC_ ReadParameter _CANopen ReadPm_C2_Done змінюється на TRUE, ReadPm_C2_DaTy =2 і ReadPm_C2_Dat=1000. Тобто вміст веденого параметра сервоприводу P1-55, який зчитується, становить 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв.)

8.14.2 Інструкції по Ethernet

8.14.2.1 ETH_Link_Config

FB/FC	Пояснення	Застосовна модель
FB	ETH_Link_Config використовується для налаштування параметрів для обміну даними MODBUS TCP на порту Ethernet 2 контролера руху.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
LinkNum	Встановіть кількість обміну даними MODBUS TCP	UINT	1~16 (0)	Коли *Execute* змінюється з FALSE на TRUE
EnableLink	Увімкніть або вимкніть посилання	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
SlaveIP_segment1	Установіть перший сегмент цільової IP-адреси	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
SlaveIP_segment2	Встановіть другий сегмент цільової IP-адреси	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
SlaveIP_segment3	Встановіть третій сегмент цільової IP-адреси	USINT	0~255 (0)	Коли *Виконати* зміни від

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
				FALSE на TRUE
SlaveIP_segment4	Встановіть четвертий сегмент цільової IP-адреси	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
SlaveNodeID	Встановіть ідентифікатор вузла веденого пристрою MODBUS	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
ObjType	Встановіть тип підпорядкованих регістрів для читання та запису. 0 : реєстр слів 1 : Бітовий регістр	USINT	0~1 (0)	Коли *Execute* змінюється з FALSE на TRUE
En_FunCode17	Встановіть код функції 17 для використання чи ні.	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
Write_Buffer_Addres s	Укажіть початковий регістр головного, де зберігаються дані для надсилання.	UINT	%MW0~%MW32 767 %QW0~%QW63	Коли *Execute* змінюється з FALSE на TRUE
Write_Buffer_Address_Off set	Встановіть зміщення початкового регістру головних даних надсилання. Значення налаштування 1 означає, що зсув становить 1 слово, якщо дані записуються в регістри слів підлеглого пристрою. Значення параметра 1 означає, що зміщення становить 1 біт, якщо дані записуються в бітові регістри підлеглого пристрою.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
Write_Destination_Address	Вкажіть початкову адресу регістрів веденого пристрою MODBUS, який отримує дані від ведучого	UINT	16#0~16#FFFF (0)	Коли *Execute* змінюється з FALSE на TRUE
Написати_довжину	Довжина даних для запису	UINT	Реєстр слів: 0~100 Бітовий регістр: 0~256 Регістр слова або бітовий регістр можна встановити за допомогою значення *ObjType* (0)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Write_FunCode	Вкажіть код функції для запису даних у регістри.	USINT	Бітовий регістр： 05, 16#0F Реєстр слів： 06,16#10 (16#10)	Коли *Execute* змінюється з FALSE на TRUE
Read_Buffer_Address	Вкажіть початковий регістр головного, де зберігаються отримані дані.	UINT	%MW0~%MW32767 %QW0~%QW63	Коли *Execute* змінюється з FALSE на TRUE
Read_Buffer_Address_Offset	Встановіть зміщення початкового регістру головних даних прийому. Значення налаштування 1 означає, що зсув становить 1 слово, якщо зчитується регістр слів підлеглого пристрою. Значення параметра 1 означає, що зсув становить 1 біт, якщо зчитується бітовий регістр веденого пристрою.	USINT	0~255 （ 0 ）	Коли *Execute* змінюється з FALSE на TRUE
Read_Destination_Address	Вкажіть початкову адресу підлеглого пристрою MODBUS, який має зчитувати головний	UINT	16#0~16#FFFF （ 0 ）	Коли *Execute* змінюється з FALSE на TRUE
Read_Length	Довжина даних для читання	UINT	0~100 для ObjType: 0; 0~256 для ObjType: 1; （ 0 ）	Коли *Execute* змінюється з FALSE на TRUE
Read_FunCode	Встановіть код функції для читання регістрів save.	USINT	Бітовий регістр： 01~02 Реєстр слів： 03~04 （ 03 ）	Коли *Execute* змінюється з FALSE на TRUE
Час вийшов	Встановіть час, протягом якого головний пристрій очікує відповіді підлеглого. Тайм-аут підпорядкованого пристрою виникає, якщо підлеглий не відповідає на запит головного протягом встановленого часу. Одиниця: мс	UINT	0~65535 （ 0 ）	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. Вхідні параметри SlaveIP_segment1, SlaveIP_segment2, SlaveIP_segment3 і SlaveIP_segment4 є відповідно сегментом 1~сегментом 4 IP-адреси підлеглого пристрою. Наприклад, підлеглий IP-адреса 192.168.1.10. Отже, вхідне значення SlaveIP_segment1 дорівнює 192, SlaveIP_segment2 — 168; SlaveIP_segment3 дорівнює 1, а SlaveIP_segment4 — 10.

2. Вхідні параметри Write_Buffer_Address і Read_Buffer_Address означають початкові регістри головного модуля MODBUS TCP, де зберігаються надіслані та отримані дані. Потрібно ввести два вхідних значення. Ви можете визначати змінні та комбінувати для них адреси реєстрів, наприклад %MW0.

2. Вхідний параметр *ObjType* — це тип даних прочитаного/записаного параметра. Якщо *ObjType* дорівнює 0, це означає читання та запис даних у регістри слів підлеглого пристрою, а діапазони Write_Length і Read_Length становлять 0~100. Write_Length і Read_Length не можуть бути 0 одночасно.
Якщо *ObjType* дорівнює 1, це означає читання та запис даних у бітових регістрах підпорядкованого пристрою, а діапазони Write_Length і Read_Length становлять 0~256. Write_Length і Read_Length не можуть бути 0 одночасно.

4. Коли інструкція без введення Write_FunCode читає та записує дані в регістрі Word відповідно до вибору *ObjType* , код функції за замовчуванням для запису даних – 16#10, а код функції за замовчуванням для читання даних – 03. Коли *ObjType* визначає бітовий регістр, код функції за замовчуванням для запису даних – 16#0F, а код функції для читання даних, 01 або 02, вибирається залежно від типу бітових регістрів згідно з інструкцією до відповідного продукту.

4. Прошивка, пізніша за V1.02, підтримує введення Write_FunCode інструкції.

4. Для інструкції з введенням Write_FunCode бітовий регістр або регістр Word можна вибрати через *ObjType* , код функції для запису даних можна вказати через вхід Write_FunCode, а код функції для читання даних можна вказати через вхід Read_FunCode.

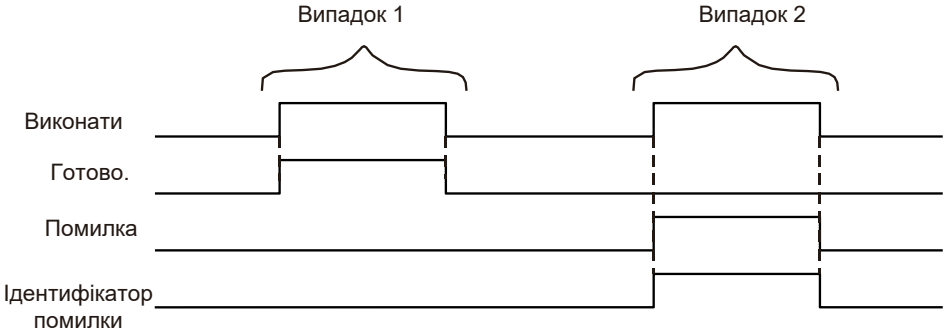
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE Коли налаштування параметрів завершено.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після завершення налаштування параметрів.	◆ Коли Execute змінюється з TRUE на FALSE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли Execute змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1: Коли *Execute* змінюється з FALSE на TRUE, *Done* стає TRUE, а параметри налаштовані успішно. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

Випадок 2: Коли *Execute* змінюється з FALSE на TRUE, *Error* стає TRUE, а *ErrorID* показує відповідний код помилки, якщо якийсь параметр є недопустимим. Коли *Execute* змінюється на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* скидається на 0.

- **функція**

ETH_Link_Config використовується для налаштування параметрів MODBUS TCP. Прошивка V1.01 і вище підтримує функцію.

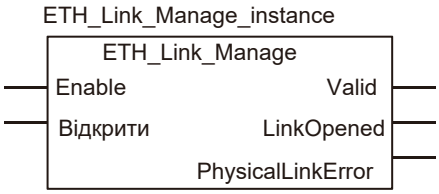
1. Коли використовується головна функція Modbus TCP контролера руху, лише порт Ethernet 2 підтримує цю функцію, а порт не підтримує цю функцію.
2. TCP-дані MODBUS повинні надсилатися разом за допомогою інструкцій ETH_Link_Config і ETH_Link_Manage
3. Якщо ви змінюєте параметри під час виконання інструкції, параметри не записуються. Після зміни параметрів інструкції параметр не буде записаний, доки знову не буде запущено Execute.
4. Якщо ви зміните параметри інструкції ETH_Link_Config, нові параметри не набудуть чинності негайно, доки інструкція ETH_Link_Manage не буде виконана повторно.
5. Коли регістри слів у підпорядкованому пристрої зчитуються та записуються, регістри %MW локального пристрою можна вибрати як регістри для зберігання прочитаних і записаних даних. Регістри зберігання варіюються від %MW0~%MW32767. Якщо регістри перевищують діапазон або використовується інший регістр, в інструкції виникне помилка.
   Коли бітові регістри підлеглого пристрою зчитуються та записуються, регістри %MW і %QW локального пристрою можна вибрати як регістри для зберігання прочитаних і записаних даних. Діапазони регістрів зберігання є %MW0~%MW32767 і %QW0~%QW63. Якщо регістри перевищують діапазони або використовується інший регістр, в інструкції виникне помилка.

6. Коли бітові регістри підпорядкованого пристрою зчитуються та записуються, а значення зміщення дорівнює 0, ПЛК почне читати або записувати біти даних Read_Length і Write_Length, починаючи з біта 0 початкового регістра, де зберігаються дані.
7. Коли бітові регістри підлеглого пристрою зчитуються та записуються, а значення зсуву становить n, яке не дорівнює 0, ПЛК почне читати або записувати біти даних Read_Length і Write_Length шляхом зміщення n бітів назад.
   Наприклад, Write_Buffer_Address_Offset дорівнює 0, ObjType дорівнює 1, регістр у поєднанні з Write_Buffer_Address дорівнює %MW0, а Write_Length дорівнює 5. Тоді ПЛК надішле дані в %MX0.0~%MX0.4 на підлеглий пристрій.
   Наприклад, Write_Buffer_Address_Offset дорівнює 8, ObjType дорівнює 1, регістр поєднується з Write_Buffer_Address дорівнює %MW0, а Write_Length дорівнює 5. Потім ПЛК надішле дані в %MX1.0~%MX1.4 на підлеглий пристрій.

8. Значення параметрів цієї інструкції дійсні лише під час роботи ПЛК. Коли контролер руху повторно вмикається після вимкнення живлення, всі параметри, налаштовані до вимкнення живлення, стають недійсними. Інструкцію ETH_Link_Config потрібно виконати ще раз, якщо для використання потрібна конфігурація перед вимкненням живлення.
9. Коли використовується головна функція MODBUS TCP контролера руху, контролер руху здійснює обмін даними з іншими підлеглими через порт Ethernet2 контролера руху.

8.14.2.2 ETH_Link_Manage

FB/FC	Пояснення	Застосовна модель
FB	ETH_Link_Manage використовується для ввімкнення та вимкнення обміну даними MODBUS TCP.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
ВІДЧИНЕНО	Увімкніть або вимкніть обмін даними MODBUS TCP.	BOOL	Правда чи брехня (FALSE)	Коли *Enable* змінюється з FALSE на TRUE

Примітка:

ETH_Link_Manage використовується для ввімкнення обміну даними MODBUS TCP після налаштування параметрів MODBUS TCP за допомогою інструкції ETH_Link_Config.

- **Вихідні параметри**

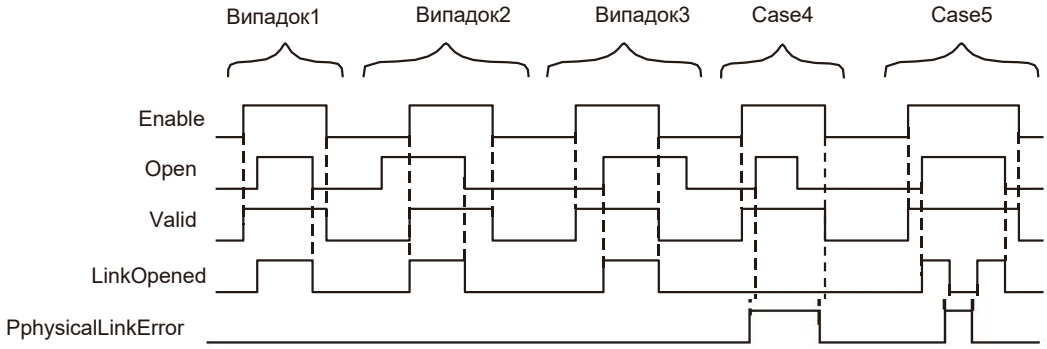
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
Посилання відкрито	ІСТИНА, коли ввімкнено обмін даними MODBUS TCP.	BOOL	ІСТИНА / НЕПРАВДА
PhysicalLinkError	ІСТИНА, коли фізичне з'єднання з інтерфейсом Ethernet ПЛК роз'єднано.	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE
Посилання відкрито	◆ Коли підключення до MODBUS TCP успішне.	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *Open* змінюється з TRUE на FALSE ◆ Коли фізичне з'єднання з інтерфейсом Ethernet центрального процесора розірвано.
PhysicalLinkError	◆ Коли *Open* має значення TRUE і фізичне з'єднання з інтерфейсом Ethernet ПЛК роз'єднано.	◆ Коли *Enable* змінюється з TRUE на FALSE

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
		◆ Коли *LinkOpened* змінюється з FALSE на TRUE ◆ Коли фізичне з'єднання з інтерфейсом Ethernet ПЛК відновлюється.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 Коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Коли *Open* змінюється з FALSE на TRUE, *LinkOpened* змінюється на TRUE. Коли *Open* змінюється з TRUE на FALSE, *LinkOpened* змінює FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE.

Випадок 2 Коли *Enable* змінюється на FALSE, *Open* змінюється на TRUE, і на даний момент результати інструкції не змінюються. Коли *Enable* змінюється з FALSE на TRUE, *Valid* і *LinkOpened* змінюються на TRUE. Коли *Open* змінюється на FALSE, *LinkOpened* змінюється на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE.

Випадок 3 Коли *Enable* змінюється на TRUE, *Valid* змінюється на TRUE. Коли *Open* змінюється з FALSE на TRUE, *LinkOpened* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *LinkOpened* змінюються на FALSE. Згодом зміна *Open* з TRUE на FALSE не вплине на вихідний результат.

Випадок 4 Коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Якщо в даний момент є проблема з підключенням до порту Ethernet 2 контролера руху, *PhysicalLinkError* змінюється на TRUE, якщо *Open* змінюється на TRUE; *PhysicalLinkError* не змінюється на FALSE, якщо *Open* змінюється на FALSE і *Valid* і *PhysicalLinkError* змінюється на FALSE одночасно, якщо *Enable* змінюється з TRUE на FALSE.

Випадок 5 Якщо зв'язок із портом Ethernet 2 контролера руху розривається під час виконання інструкції, *PhysicalLinkError* змінюється на TRUE, а *LinkOpened* змінюється на FALSE. Коли зв'язок із портом Ethernet 2 контролера руху відновлюється, *PhysicalLinkError* змінюється на FALSE, а *LinkOpended тим часом* змінюється на TRUE.

- **функція**

ETH_Link_Manage використовується для запуску зв'язку MODBUS TCP. Прошивка V1.01 і вище підтримує функцію.

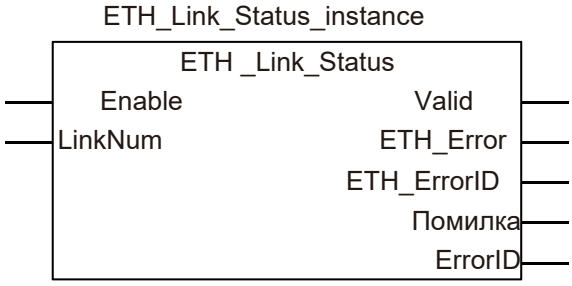
1. Після того, як вхід *Enable* для ETH_Link_Manage змінюється на TRUE, а *Open* встановлено на TRUE, дані MODBUS TCP починають надсилатися, якщо фізичний інтерфейс Ethernet ПЛК підключено нормально. Коли *Open* змінюється з TRUE на FALSE, ПЛК припиняє надсилати дані MODBUS TCP.

2. Якщо *для Enable* встановлено значення FALSE під час виконання інструкції, усі результати ETH_Link_Manage змінюються на FALSE. Але ПЛК не припинить надсилання даних MODBUS TCP. Посилання не буде закрито, доки для *параметра «Відкрити»* не буде встановлено значення FALSE, а для параметра « *Enable» встановлено* значення TRUE.

3. Повідомлення про помилку від підлеглого пристрою або тайм-аут зв'язку не впливають на виконання інструкції.

8.14.2.3 ETH_Link_Status

FB/FC	Пояснення	Застосовна модель
FB	ETH_Link_Status використовується, щоб спостерігати, чи виникає помилка в зв'язку MODBUS TCP, якому відповідає номер, або якщо підлеглий пристрій відповідає кодами помилок.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція перевіряється, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
LinkNum	Номер зв'язку MODBUS TCP, який буде відстежуватися.	UINT	1~16 (0)	Коли *Enable* змінюється з FALSE на TRUE

- **Вихідні параметри**

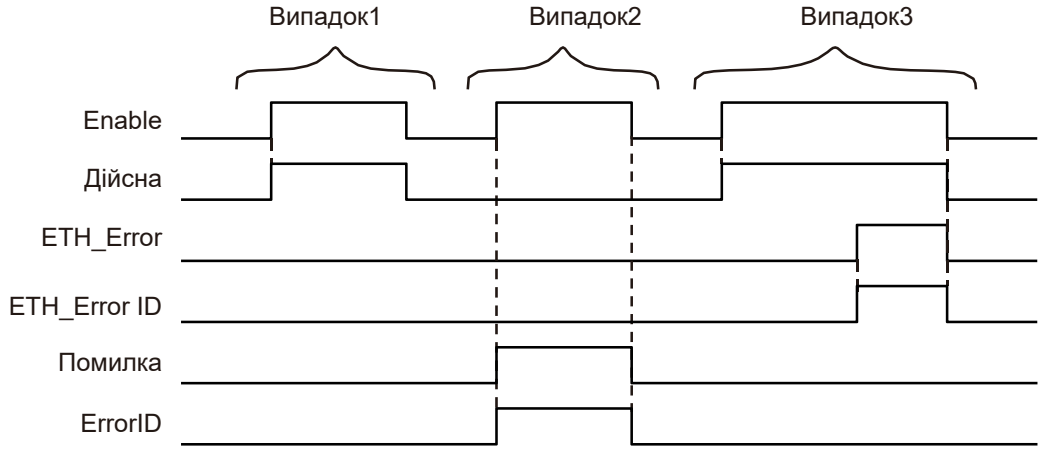
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
ETH_Error	TRUE, коли виникає помилка в обміні даними MODBUS TCP.	BOOL	ІСТИНА / НЕПРАВДА
ETH_ErrorID	Код помилки обміну даними MODBUS TCP. Зверніться до таблиці в наступному розділі **функцій** . *Я*кщо значення *ETH_ErrorID* дорівнює 1~10, зверніться до відповідного підпорядкованого пристрою для пояснення відповідного ідентифікатора помилки.	WORD	
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE.

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
ETH_Error	◆ Коли виникає помилка в обміні даними MODBUS TCP.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли обмін даними MODBUS TCP відновиться до нормального стану.
Помилка	◆ При виникненні помилки у виконанні інструкції.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли заповнено правильне значення параметра.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 Коли *Enable* змінюється на TRUE, *Valid* змінюється на TRUE. Коли *Enable* змінюється на FALSE, *Valid* змінюється на FALSE.

Випадок 2 Коли виникає помилка, коли *Enable* має значення TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. *Помилка* змінюється на FALSE, а значення *ErrorID* змінюється на 0, коли *Enable* змінюється на FALSE.

Випадок 3 Коли *Enable* змінюється на TRUE, *Valid* змінюється на TRUE. *ETH _ Error* змінюється на TRUE, а *ETH _ ErrorID* показує відповідний код помилки, коли надсилання даних MODBUS TCP не вдається або час очікування виникає під час виконання інструкції. Коли *Enable* змінюється на FALSE, *ETH _ Error* змінюється на FALSE, а *ETH _ ErrorID* відображає відповідний код помилки.

- **функція**

ETH_Link_Status використовується для спостереження, якщо виникає помилка у відповідному каналі MODBUS TCP або підлеглий пристрій відповідає кодами помилок. Прошивка V1.01 і вище підтримує цю функцію.

Код помилки	опис	Як боротися з
101	Посилання TCP вимкнено.	1. Перевірте, чи нормальне з'єднання Ethernet. 2. Повторно виконайте інструкцію ETH_Link_Manage, щоб увімкнути обмін даними MODBUS TCP.
102	Тайм-аут TCP-з'єднання	1. Перевірте, чи нормальне з'єднання Ethernet. 2. Перевірте, чи правильні налаштування для параметрів інструкції ETH_Link_Config.

Код помилки	опис	Як боротися з
103	Час очікування відповіді на повідомлення MODBUS TCP	1. Перевірте, чи нормальне з’єднання Ethernet. 2. Збільште час очікування, змінивши значення параметра *Timeout* інструкції ETH_Link_Config.
104	Зарезервований	-
105	Зарезервований	-
106	Помилка ідентифікатора транзакції (у заголовку повідомлення)	Переконайтеся, що формат відповідного повідомлення від підлеглого пристрою правильний.
107	Помилка ідентифікатора протоколу (у заголовку повідомлення)	Переконайтеся, що формат відповідного повідомлення від підлеглого пристрою правильний.
108	Помилка довжини повідомлення Modbus TCP (у заголовку повідомлення)	Переконайтеся, що довжина відповідного повідомлення від підлеглого пристрою правильна.
109	Зарезервований	-
110	Встановлення зв'язку виконується, коли зв'язок TCP не вимкнено.	Вимкніть поточне з’єднання перед відновленням з’єднання.

8.14.2.4 Приклад обміну даними MODBUS TCP

Приклад програмування 1

■ **Приклад читання та запису регістрів слів у раб**

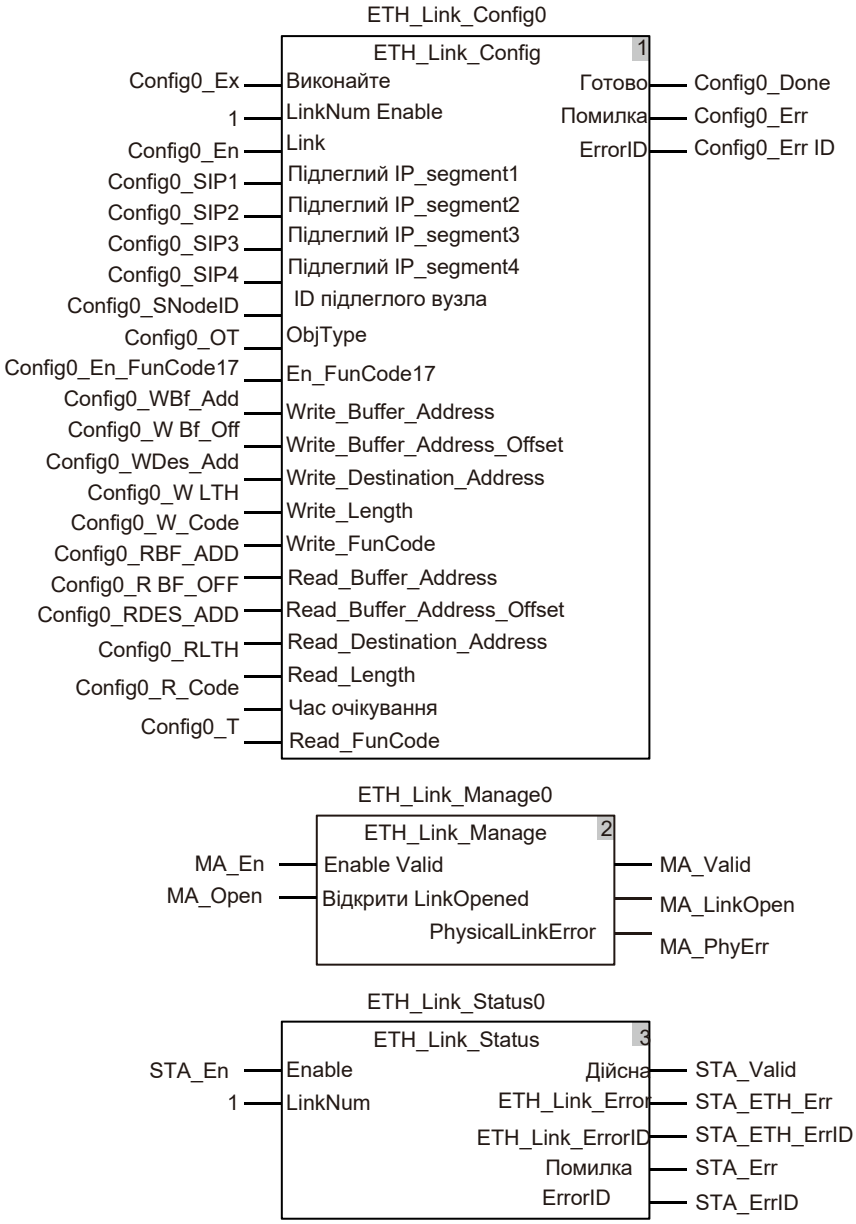
➢ **Приклад пояснення**

1. Контролер руху є провідним MODBUS TCP, DVP12SE є підлеглим MODBUS TCP, а IP-адреса DVP12SE становить 192.168.1.10.
2. Контролер руху записує значення в %MW10~%MW19 до D0~D9 у DVP12SE, потім зчитує значення в D100~D109 у DVP12SE та зберігає прочитані значення в %MW110~%MW119.

➢ **Таблиця змінних і програма**

Ім'я змінної	Адреса	Тип даних	Початкове значення
ETH_Link_Config0		ETH_Link_Config	
Config0_Ex		BOOL	
Config0_En		BOOL	TRUE
Config0_SIP1		USINT	192
Config0_SIP2		USINT	168
Config0_SIP3		USINT	1
Config0_SIP4		USINT	10
Config0_SNodeID		USINT	0
Config0_OT		USINT	0
Config0_En_FunCode17		BOOL	FALSE
Config0_WBf_Add	%MW0	UINT	
Config0_WBf_Off		USINT	10
Config0_WDes_Add		UINT	16#1000
Config0_WLTH		UINT	10
Config0_ W_Code		USINT	16#10
Config0_RBF_ADD	%MW100	UINT	
Config0_RBF_OFF		USINT	10
Config0_RDES_ADD		UINT	16#1064
Config0_RLTH		UINT	10
Config0_R_Code		USINT	16#03
Config0_T		UINT	1000
Config0_Done		BOOL	
Config0_Err		BOOL	
Config0_ErrID		WORD	
ETH_Link_Manage0		ETH_Link_Manage	
MA_En		BOOL	
MA_Open		BOOL	
MA_Valid		BOOL	
MA_LinkOpen		BOOL	

Ім'я змінної	Адреса	Тип даних	Початкове значення
MA_PhyErr		BOOL	
ETH_Link_Status0		ETH_Link_Status	
STA_En		BOOL	
STA_Valid		BOOL	
STA_ETH_Err		BOOL	
STA_ETH_ErrID		WORD	
STA_Err		BOOL	
STA_ErrID		WORD	



- **Пояснення операцій і обміну даними**
  1. Поєднайте Config0_WBf_Add і Config0_RBF_ADD з %MW0 і %MW100 відповідно. Початкові значення Config0_WBf_Off і Config0_RBf_OFF дорівнюють 10. Виконайте онлайн

після успішного компілювання та завантаження програми.

2. Встановіть Config_Ex на TRUE. Після завершення виконання інструкції ETH_Link_Config встановіть MA_En на TRUE, а потім MA_Open на TRUE. Після того, як вихід MA_LinkOpen інструкції ETH_Link_Manage зміниться на TRUE, контролер руху починає обмінюватися даними з DVP12SE.

За допомогою інструкції ETH_Link_Status можна спостерігати за поточним статусом зв'язку. Відповідні зв'язки між контролером руху та DVP12SE показані в наступній таблиці.

%MW реєструється в контролері руху		D прописується в DVP12SE
%MW10	⇨	D0
%MW11		D1
%MW12		D2
…….		……
%MW18		D8
%MW19		D9
%MW110	⇦	D100
%MW111		D101
%MW112		D102
……		……
%MW118		D108
%MW119		D109

Приклад програмування 2

■ Приклад читання та запису бітових регістрів у slave

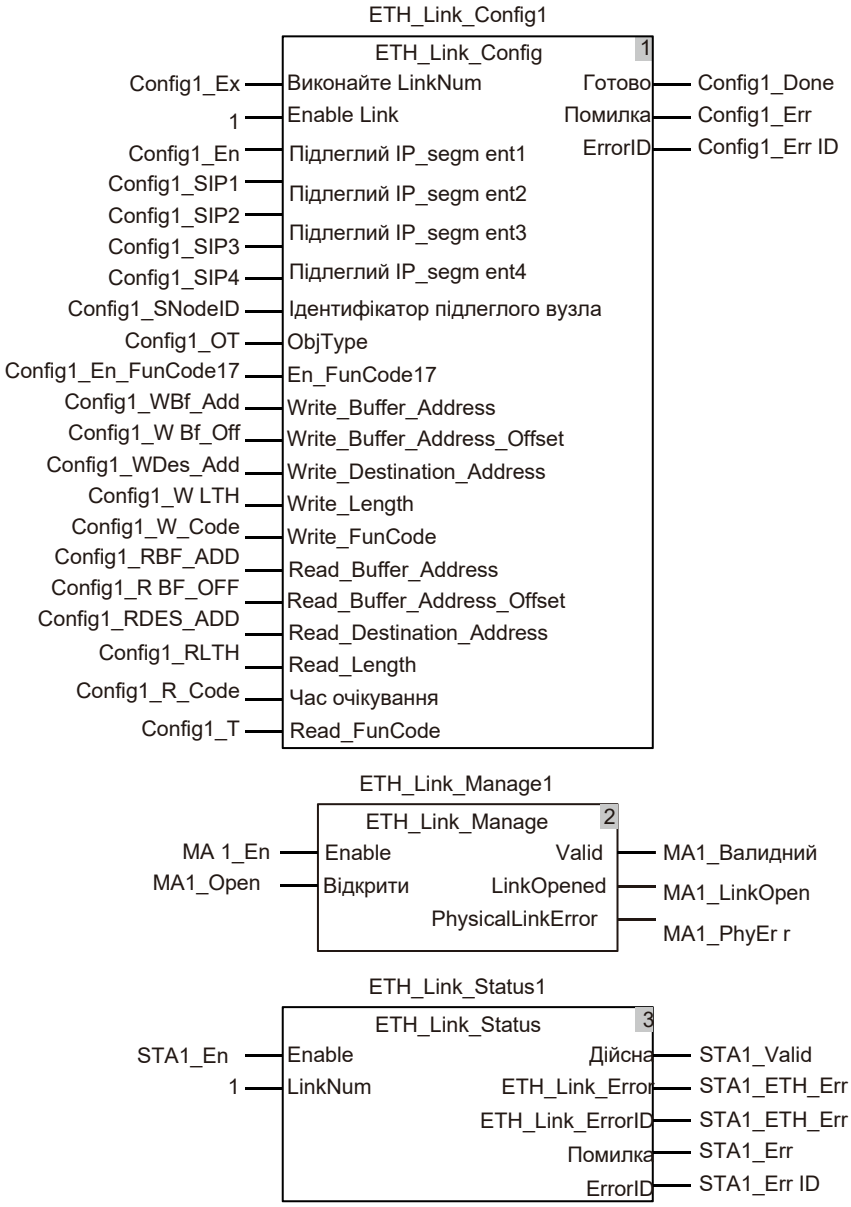
➢ Приклад пояснення

1. Контролер руху є провідним MODBUS TCP, DVP12SE є підлеглим MODBUS TCP, а IP-адреса DVP12SE становить 192.168.1.10.
2. Контролер руху записує значення в %MX0.0~%MX0.7 до Y0~Y7 у DVP12SE, потім зчитує значення в Y20~Y27 у DVP12SE та зберігає прочитані значення в %MX2.0~%MX2.7.

➢ Таблиця змінних і програма

Ім'я змінної	Адреса	Тип даних	Початкове значення
ETH_Link_Config1		ETH_Link_Config	
Config1_Ex		BOOL	
Config1_En		BOOL	TRUE
Config1_SIP1		USINT	192
Config1_SIP2		USINT	168
Config1_SIP3		USINT	1
Config1_SIP4		USINT	10
Config1_SNodeID		USINT	0
Config1_OT		USINT	1
Config1_En_FunCode17		BOOL	FALSE
Config1_WBf_Add	%MW0	UINT	

Ім'я змінної	Адреса	Тип даних	Початкове значення
Config1_WBf_Off		USINT	0
Config1_WDes_Add		UINT	16#0500
Config1_WLTH		UINT	8
Config1_ W_Code		USINT	16#0F
Config1_RBF_ADD	%MW1	UINT	
Config1_RBF_OFF		USINT	0
Config1_RDES_ADD		UINT	16#0510
Config1_RLTH		UINT	8
Config1_R_Code		USINT	1
Config1_T		UINT	1000
Config1_Done		BOOL	
Config1_Err		BOOL	
Config1_ErrID		WORD	
ETH_Link_Manage1		ETH_Link_Manage	
MA1_En		BOOL	
MA1_Open		BOOL	
MA1_Valid		BOOL	
MA1_LinkOpen		BOOL	
MA1_PhyErr		BOOL	
ETH_Link_Status1		ETH_Link_Status	
STA1_En		BOOL	
STA1_ Дійсний		BOOL	
STA1_ETH_Err		BOOL	
STA1_ETH_ErrID		WORD	
STA1_Err		BOOL	
STA10_ErrID		WORD	



➢ **Пояснення операцій і обміну даними**

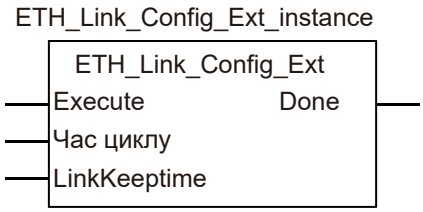
1. Поєднайте Config1_WBf_Add і Config1_RBF_ADD з %MW0 і %MW1 відповідно. Встановіть значення Config1_WLTH і Config1_RLTH на 8 і Config1_OT на 1. Встановіть значення Config1_WDes_Add і Config1_RDes_Add на 16#0500 і 16#0510 відповідно, а Config1_R_Code на 1.
2. Після успішного компілювання та завантаження програми виконайте онлайн. Встановіть для Config1_Ex значення TRUE. Після завершення виконання інструкції ETH_Link_Config встановіть MA1_En на TRUE, а потім MA1_Open на TRUE. Після того, як вихід MA1_LinkOpen інструкції ETH_Link_Manage зміниться на TRUE, контролер руху починає обмінюватися даними з DVP12SE.

За допомогою інструкції ETH_Link_Status можна спостерігати за поточним статусом зв'язку. Відповідні зв'язки між контролером руху та DVP12SE показані в наступній таблиці.

Пристрої в контролері руху		Розрядні пристрої Y в DVP12SE
%MX0,0	⇨	Y0
%MX0,1		Y1
%MX0,2		Y2
…….		……
%MX0,6		Y6
%MX0,7		Y7
%MX2.0	⇦	Y20
%MX2.1		Y21
%MX2.2		Y22
……		……
%MX2.6		Y26
%MX2.7		Y27

8.14.2.5 ETH_Link_Config_Ext

FB/FC	Пояснення	Застосовна модель
FB	ETH_Link_Config_Ext використовується для налаштування часу циклу та тривалості зв'язку для обміну даними MODBUS TCP.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Час циклу	Час циклу, коли головний пристрій надсилає дані MODBUS TCP. (Одиниця: мілісекунди)	МАСИВ [1..16] UINT	0~65535 (0)	Коли *Execute* змінюється з FALSE на TRUE.
LinkKeeptime	Тривалість зв'язку майстра (одиниці: секунди)	МАСИВ [1..16] UINT	0~65535 (0)	Коли *Виконати* змінюється з FALSE на TRUE.

Примітка:

1. Кожен елемент значень у *CycleTime* та *LinkKeeptime* відповідає номеру в інструкції ETH_Link_Config. Наприклад, перший елемент значення *CycleTime* відповідає часу циклу надсилання даних числа 1.
2. Одиницею *CycleTime* є мілісекунди (мс), а *LinkKeepTime* – секунди (с).
3. *LinkKeeptime* – це тривалість посилання (одиниці:секунди). Якщо обмін даними між головним і підлеглим не відбудеться протягом часу, визначеного *LinkKeeptime,* контролер автоматично від'єднає зв'язок із підлеглим.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **функція**

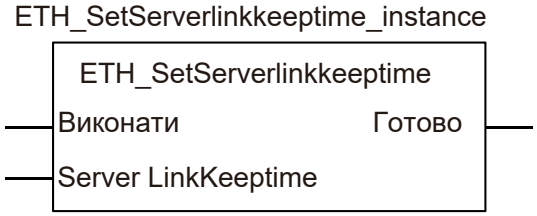
Коли використовується головна функція MODBUS TCP контролера, ETH_Link_Config_Ext можна використовувати для встановлення часу циклу та тривалості зв'язку для обміну даними MODBUS TCP. Прошивка V1.01 і вище

підтримує функцію.

1. Інструкцію ETH_Link_Config_Ext можна використовувати для встановлення часу циклу та тривалості зв’язку для обміну даними MODBUS TCP. Цю інструкцію також можна використовувати в конфігурації параметрів MODBUS TCP. І його все ще можна використовувати в головній функції MODBUS TCP контролера. Надсилання одного фрагмента даних MODBUS TCP завершено, а потім негайно буде надіслано інший. За замовчуванням тривалість посилання становить 30 секунд.
2. Значення параметра, змінене під час виконання інструкції, не буде записано. Нове значення параметра інструкції буде записано лише після повторного *запуску* інструкції «Виконати».
3. ETH_Link_Config_Ex і ETH_Link_Config не мають жодного відношення в послідовності виконання. Параметри ETH_Link_Config_Ext не набудуть чинності, якщо ETH_Link_Config_Ext не буде виконано до ETH_Link_Manage.

8.14.2.6 ETH_SetServerlinkkeeptime

FB/FC	Пояснення	Застосовна модель
FB	ETH_SetServerlinkkeeptime використовується для встановлення тривалості з'єднання, оскільки контролер служить підлеглим TCP MODBUS.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
ServerLinkKeeptime	Тривалість з'єднання, коли контролер працює як підлеглий пристрій MODBUS TCP (Одиниця вимірювання: секунди)	UINT	0~65535 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

Якщо для параметра *ServerLinkKeeptime* встановлено значення 0 або порожній, тривалість зв'язку за замовчуванням для підлеглого пристрою MODBUS TCP, який обслуговує контролер, становить 30 с.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE.

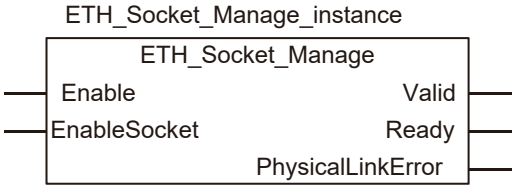
- **функція**

ETH_SetServerlinkkeeptime використовується для встановлення тривалості з'єднання, оскільки контролер служить підлеглим TCP MODBUS.

1. Значення *ServerLinkKeeptime* дійсне лише тоді, коли контролер працює як підлеглий пристрій MODBUS TCP. Коли контролер працює як головний MKODBUS TCP, виконання інструкції ETH_ SetServerlinkkeeptime не матиме впливу на головну функцію MODBUS TCP.
2. Якщо не відбувається обмін даними між підлеглим контролером і головним протягом часу, визначеного ServerLinkKeeptime, підлеглий контролер автоматично роз'єднає зв'язок із головним пристроєм MODBUS TCP.
3. Інструкція може встановлювати лише тривалість зв'язку підлеглого пристрою MODBUS TCP, з яким працює порт Ethernet 2 контролера руху.

8.14.2.7 ETH_Socket_Manage

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Manage використовується для керування Socket TCP/UDP.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
EnableSocket	Увімкніть Socket.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* змінюється з FALSE на TRUE

Примітка:
ETH_Socket_Manage — увімкнути функцію Socket. Інші інструкції, пов'язані з Socket, не можна використовувати, доки інструкція не буде виконана.

● **Вихідні параметри**

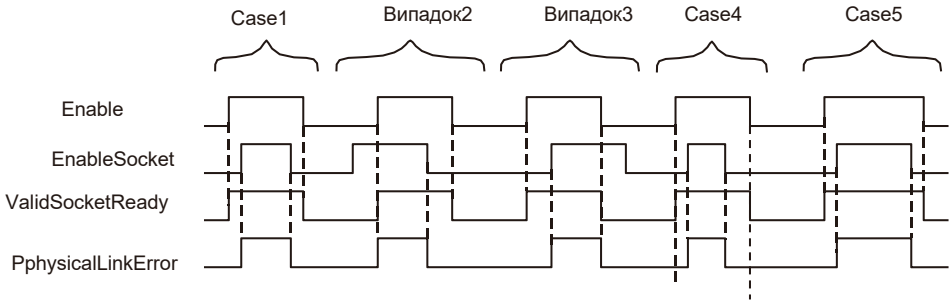
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли вихід інструкції дійсний.	BOOL	ІСТИНА/НЕПРАВДА
SokcetReady	TRUE, коли ввімкнення Socket успішне.	BOOL	ІСТИНА/НЕПРАВДА
PhysicalLinkError	ІСТИНА, коли фізичне підключення до порту Ethernet контролера розірвано.	BOOL	ІСТИНА/НЕПРАВДА

■ **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE.
SocketReady	◆ Коли ввімкнення Socket успішне.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *EnableSocket* змінюється з TRUE на FALSE.
PhysicalLinkError	◆ Коли *Enable* змінюється на TRUE і фізичне підключення до порту Ethernet 2 контролера встановлюється	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ При фізичному підключенні до

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
	відключений.	Порт Ethernet 2 контролера повертається до нормального стану.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Коли *EnableSocket* змінюється з FALSE на TRUE, *SocketReady* змінюється на TRUE. Коли *EnableSocket* змінюється з TRUE на FALSE, *SocketReady* змінює FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE.

Випадок 2 : коли *Enable* має значення FALSE, а *EnableSocket* змінює значення TRUE, вихідні дані інструкції залишаються незмінними. Коли *Enable* змінюється з FALSE на TRUE, *Valid* і *SocketReady* змінюються на TRUE. Коли *EnableSocket* змінюється на FALSE, *SocketReady* змінюється на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE.

Випадок 3 : коли *Enable* змінюється на TRUE, *Valid* змінюється на TRUE. Коли *EnableSocket* змінюється з FALSE на TRUE, *SocketReady* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *SocketReady* змінюються на FALSE. Після цього зміна *EnableSocket* з TRUE на FALSE не вплине на вихід інструкції.

Випадок 4 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Якщо в даний момент виникла проблема з підключенням до порту Ethernet 2 AS516E-B, *PhysicalLinkError* зміниться на TRUE, коли *параметр Enable* зміниться на TRUE. Коли *EnableSocket* змінюється на TRUE *, SocketReady* змінюється на TRUE, а стан *PhysicalLinkError* залишатиметься незмінним. Коли *EnableSocket* змінюється з TRUE на FALSE *, SocketReady* змінюється на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *PhysicalLinkError* змінюються на FALSE.

Випадок 5 : якщо з'єднання з портом Ethernet 2 контролера розривається під час виконання інструкції, *PhysicalLinkError* змінюється на TRUE, а стан *SocketReady* залишається незмінним. Коли підключення до Ethernet 2 контролера повертається до нормального стану, *PhysicalLinkError* змінюється на FALSE.

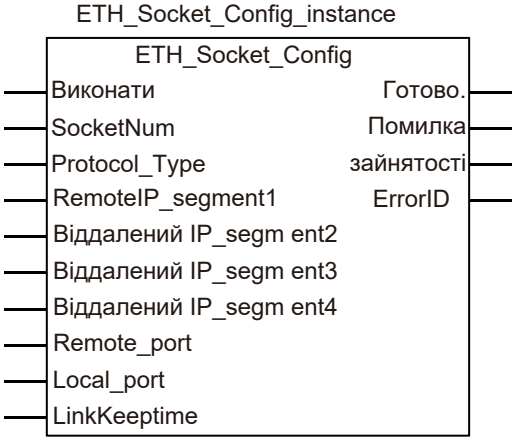
● **функція**

ETH_Socket_Manage використовується для ввімкнення Socket. Прошивка V1.02 і вище підтримує цю функцію.

1. Якщо *для Enable* встановлено значення FALSE під час виконання інструкції, усі виходи ETH_Link_Manage змінюються на FALSE. Але контролер не вимкне Socket. Socket можна вимкнути, встановивши для *EnableSocket* значення FALSE, лише якщо *Enable* має значення TRUE.
2. На виконання інструкції не вплине повідомлення про помилку або час очікування зв'язку, надісланий із цільового пристрою.

8.14.2.8 ETH_Socket_Config

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Config використовується для налаштування параметрів Socket.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
SocketNum	Вкажіть номер розетки.	USINT	1~8 (0)	Коли *Execute* змінюється з FALSE на TRUE.
Protocol_Type	Режим підключення до сокета: 0: Socket UDP 1: Сокет TCP	USINT	0: Сокет UDP 1: Сокет TCP (0)	Коли *Execute* змінюється з FALSE на TRUE.
RemoteIP_segment 1	Встановіть перший сегмент віддаленої IP-адреси.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE.
RemoteIP_segment 2	Встановіть другий сегмент віддаленої IP-адреси.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE.
RemoteIP_segment 3	Встановіть третій сегмент віддаленої IP-адреси.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE.
RemoteIP_segment 4	Встановіть четвертий сегмент віддаленої IP-адреси.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Віддалений_порт	Встановіть номер віддаленого порту.	UINT	0~65535 (0)	Коли *Execute* змінюється з FALSE на TRUE.
Локальний_порт	Встановіть номер локального порту.	UINT	0~65535 (0)	Коли *Execute* змінюється з FALSE на TRUE.
LinkKeeptime	Встановіть період часу для підключення (с) .	UINT	65535 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Вхідні параметри RemoteIP_segment1, RemoteIP_segment2, RemoteIP_segment3 і RemoteIP_segment4 відповідно представляють сегменти з першого по четвертий цільової IP-адреси. Наприклад, якщо цільовий IP – 192.168.1.10, вхідне значення SlaveIP_segment1 – 192, SlaveIP_segment2 – 168, SlaveIP_segment3 – 1 і SlaveIP_segment4 – 10.

2. Параметр *LinkKeeptime* означає тривалість з'єднання з одиницею вимірювання в секундах. Якщо протягом певного періоду часу немає передачі даних для побудованого зв'язку, контролер автоматично перерве з'єднання.

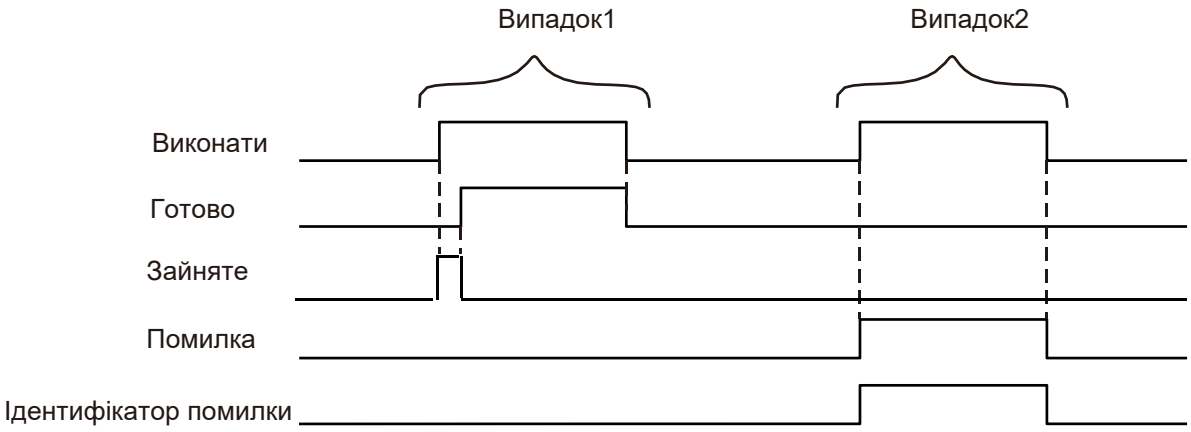
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли налаштування параметрів завершено.	BOOL	ІСТИНА/НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА/НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

■ **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після завершення налаштування параметрів.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Зайняте	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Done* змінюється з FALSE на TRUE.
Помилка	◆ Коли вхідні параметри інструкції недопустимі.	◆ Коли *Execute* змінюється з TRUE на FALSE.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1 Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і запис параметра вдається. Через один цикл *значення Done* змінюється на TRUE, а *Busy* — на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

Випадок 2. Коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки, якщо запис значень параметра є незаконним. Коли *Execute* змінюється на FALSE, *Error* змінюється на FALSE, а значення в *ErrorID* змінюється на 0.

● **функція**

ETH_Socket_Config використовується для налаштування параметрів Socket. Прошивка V1.02 і вище підтримує цю функцію.

1. Під час виконання ETH_Socket_Config змінені значення параметрів не будуть записані, якщо ви повторно не запустите введення *Execute* після того, як значення параметрів інструкції змінено.

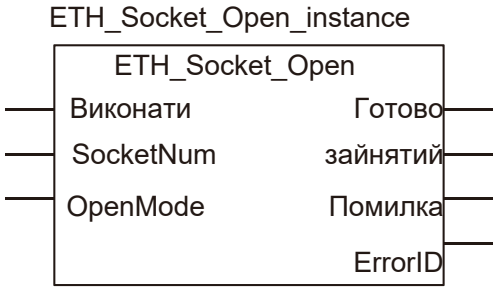
2. Якщо ви повторно запустите інструкцію для виконання, коли поточне з'єднання не роз'єднано, інструкція повідомить про помилку.

3. Контролер автоматично призначає локальний порт для поточного з'єднання, якщо значення *Local_port* встановлено на 0. Коли вибрано режим сервера Socket TCP, значення *Local_port* не може бути встановлено на 0, а один локальний порт, який не є портом 502, має бути встановлено. встановити для контролера контроль номера порту.

4. Коли для контролера вибрано режим сервера Socket TCP, контролер вибере віддалений хост відповідно до значень параметра RemoteIP_segment1~4 або Remote_port, якщо значення параметра *RemoteIP_segment1~4* або *Remote_port* не встановлено на 0. Якщо IP-адреса віддаленого хоста або номер порту не узгоджується з RemoteIP_segment1~4 або Remote_port, контролер роз'єднає з'єднання та припинить моніторинг локального порту, встановленого в поточному з'єднанні.

8.14.2.9 ETH_Socket_Open

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Open використовується для ввімкнення протоколу Socket TCP/UDP.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
SocketNum	Встановіть номер розетки	USINT	1~8 (Значення змінної має бути встановити.)	Коли *Execute* змінюється з FALSE на TRUE
OpenMode	Режим використання контролера TRUE: режим клієнта FALSE: режим сервера	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. ETH_Socket_Open використовується для ввімкнення Socket. Після виконання ETH_Socket_Open контролер спробує встановити з'єднання з іншим вузлом або зачекає, поки інший вузол надішле запит на зв'язок.
2. Коли *OpenMode* має значення TRUE, контролер перебуває в режимі **клієнта** . Після інструкції ETH_Socket_Open є
   виконується, контролер надсилає запит на зв'язок до цільового вузла.
   Коли OpenMode має значення FALSE, контролер перебуває в режимі **сервера** . Після виконання інструкції ETH_Socket_Open контролер очікує на запит на посилання від цільового вузла.
3. Якщо **Socket UDP** вибрано як режим Socket, вибір TRUE або FALSE для *OpenMode* не матиме вплив на використання інструкції.

- **Вихідні параметри**

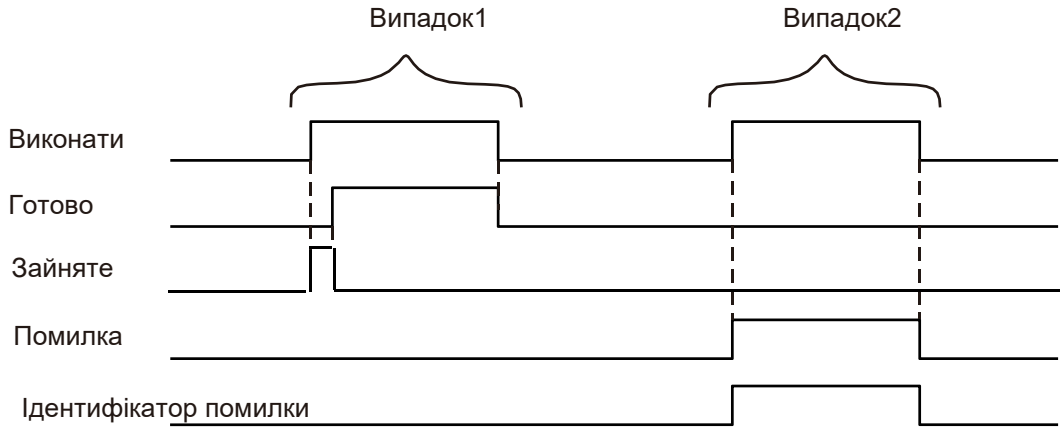
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	ІСТИНА/НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

■ **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ TRUE, коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Зайняте	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Done* змінюється з FALSE на TRUE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1 Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і запис параметрів успішний. Через один цикл *Done* змінюється на TRUE, а тим часом *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

Випадок 2. Коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки, якщо значення параметра є недопустимим. Коли *Execute* змінюється на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* змінюється на 0.

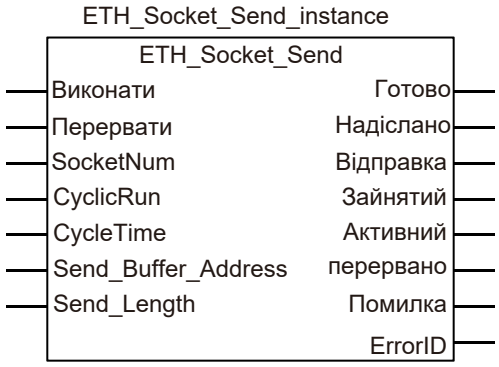
● **функція**

Інструкція ETH_Socket_Open використовується для створення TCP-з'єднання або ввімкнення функції UDP. Прошивка V1.02 і вище підтримує цю функцію.

1. Інструкції ETH_Socket_Send і ETH_Socket_Receive можуть бути виконані лише після нормального виконання інструкції ETH_Socket_Open.
2. Поточний стан підключення можна перевірити за допомогою інструкції ETH_Socket_Status після виконання інструкції ETH_Socket_Open.

8.14.2.10 ETH_Socket_Send

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Send використовується для надсилання даних Socket.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
Перервати	Перервати виконання інструкції.	BOOL	Правда чи брехня (FALSE)	
SocketNum	Вкажіть номер розетки.	USINT	1~8 (0)	Коли *Execute* змінюється з FALSE на TRUE
CyclicRun	Встановіть, чи надсилати дані циклічно. TRUE: циклічне надсилання, FALSE: лише одноразове надсилання даних	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
Час циклу	Встановіть час циклу. Одиниця: мс	UINT	0~65535 (0)	Коли *Execute* змінюється з FALSE на TRUE
Send_Buffer_Address	Вкажіть початковий регістр для збереження надісланих даних.	USINT	%MB0~%MB65 535	Коли *Execute* змінюється з FALSE на TRUE
Send_Length	Встановіть кількість байт даних для надсилання.	UINT	0~200 (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:
Вхідний параметр Send_Buffer_Address представляє адресу першого реєстру, де зберігаються дані, які надсилає контролер, а параметр Send_Length є довжиною даних, надісланих контролером. Необхідно встановити два вхідних значення.

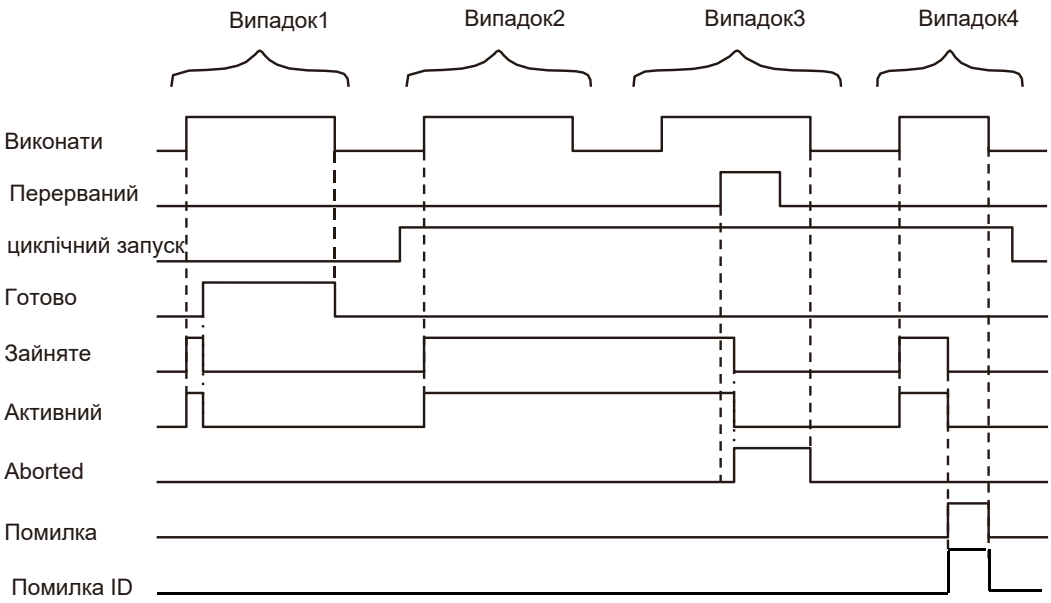
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли одноразове надсилання даних Socket завершено в режимі нециклічного надсилання.	BOOL	ІСТИНА/НЕПРАВДА
Надісланий	TRUE, коли надсилання даних Socket завершено.	BOOL	ІСТИНА/НЕПРАВДА
Відправка	TRUE, коли дані Socket надсилаються.	BOOL	ІСТИНА/НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА/НЕПРАВДА
Активний	ІСТИНА, коли інструкція керує контролером для надсилання даних.	BOOL	ІСТИНА/НЕПРАВДА
Перервано	TRUE, коли виконання інструкції перервано.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА/НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

■ **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли виконання інструкції закінчено.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Надісланий	◆ Коли надсилання однієї частини повідомлення Socket завершено.	◆ Коли контролер починає надсилати інший фрагмент даних.
Відправка	◆ Коли надсилається одна частина повідомлення Socket.	◆ Коли надсилання однієї частини даних завершено.
Зайняте	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Done* змінюється з FALSE на TRUE. ◆ Коли *Aborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
Активний	◆ Коли інструкція керує контролером для надсилання даних.	◆ Коли *Done* змінюється з FALSE на TRUE. ◆ Коли *Aborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
Перервано	◆ Коли виконання інструкції переривається.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1, коли *Execute* змінюється з FALSE на TRUE, *Busy* і *Active* змінюються на TRUE, якщо вибрано режим надсилання лише однієї частини даних. Коли надсилання однієї частини даних Socket завершено, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2, коли *Execute* змінюється з FALSE на TRUE, *Busy* та *Active* змінюються на TRUE, і інструкція починає керувати контролером для надсилання даних Socket, якщо ви вибрали режим циклічного надсилання даних. Коли *Execute* змінюється з TRUE на FALSE, стан TRUE для *Busy* і *Active* залишається незмінним.

Випадок 3. Вихідний стан залишиться незмінним, якщо знову встановити *Execute* з FALSE на TRUE після case 2. Встановлюючи *значення Abort* з FALSE на TRUE, через один цикл *Aborted* змінюється на TRUE, а *Busy* і *Active* змінюються на FALSE. Коли *Abort* змінюється з TRUE на FALSE, вихідні дані залишаться незмінними. Коли *Execute* змінюється з TRUE на FALSE, *Aborted* змінюється на FALSE.

Випадок 4, коли *Execute* змінюється з FALSE на TRUE, *Busy* і *Active* змінюються на TRUE. Коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а значення в *ErrorID* показує відповідний код помилки, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* змінюється на 0.

● **функція**

ETH_Socket_Send використовується для надсилання даних Socket. Прошивка V1.02 і вище підтримує цю функцію.

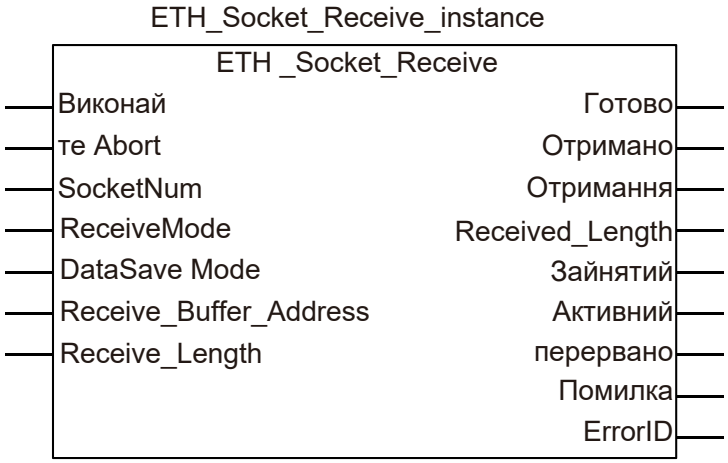
1. Вхідний параметр *CyclicRun* встановлює, чи надсилати дані циклічно чи ні, а значення *CycleTime є* час циклу. Коли *CyclicRun* має значення TRUE і виконується інструкція ETH_Socket_Send, контролер надсилає частину даних кожен період часу, який є значенням *CycleTime* . Коли *CyclicRun* має значення FALSE і виконується інструкція ETH_Socket_Send, контролер надсилає лише один фрагмент даних.
1. Коли значення *CycleTime* дорівнює 0, а *CyclicRun* — TRUE, контролер усе ще надсилає дані циклічно без будь-яких обмежень у часовому інтервалі.

3. Якщо значення вхідного параметра змінюється, а потім інструкція повторно запускається під час виконання інструкції, нове значення вхідного параметра не буде ефективним. Ви повинні використати вхідний параметр *Abort,* щоб спершу перервати інструкцію, потім повторно виконати інструкцію, після чого нове значення вхідного параметра набуде чинності.

3. Значення *Send_Length* не може бути встановлено рівним 0. Інакше під час виконання інструкції виникне помилка.

8.14.2.11 ETH_Socket_Receive

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Receive використовується для отримання даних Socket.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
Перервати	Перервати виконання інструкції.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
SocketNum	Вкажіть номер розетки.	USINT	1~8 (0)	Коли *Execute* змінюється з FALSE на TRUE
ReceiveMode	Встановіть режим отримання даних. 0 : продовжувати отримувати дані 1 : отримати лише одну частину даних	USINT	0, 1 (0)	Коли *Execute* змінюється з FALSE на TRUE
DataSaveMode	Встановіть режим збереження даних. 0 : Зварювання 1 : Покриття	USINT	0, 1 (0)	Коли *Execute* змінюється з FALSE на TRUE
Сукня Receive_Buffer_Ad	Вкажіть початковий регістр для збереження отриманих даних.	USINT	%MB0~%MB65 535	Коли *Execute* змінюється з FALSE на TRUE
Receive_Length	Встановіть кількість байт	UINT	0~200	Коли *Виконати*

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
	дані, які потрібно отримати.		(200)	змінюється з FALSE на TRUE

Примітка:
Вхідний параметр ***Receive_Buffer_Address*** представляє адресу першого реєстру, де зберігаються дані, які отримує контролер, а параметр ***Receive_Length*** є довжиною даних, отриманих контролером. Необхідно встановити два вхідних значення.

● **Вихідні параметри**

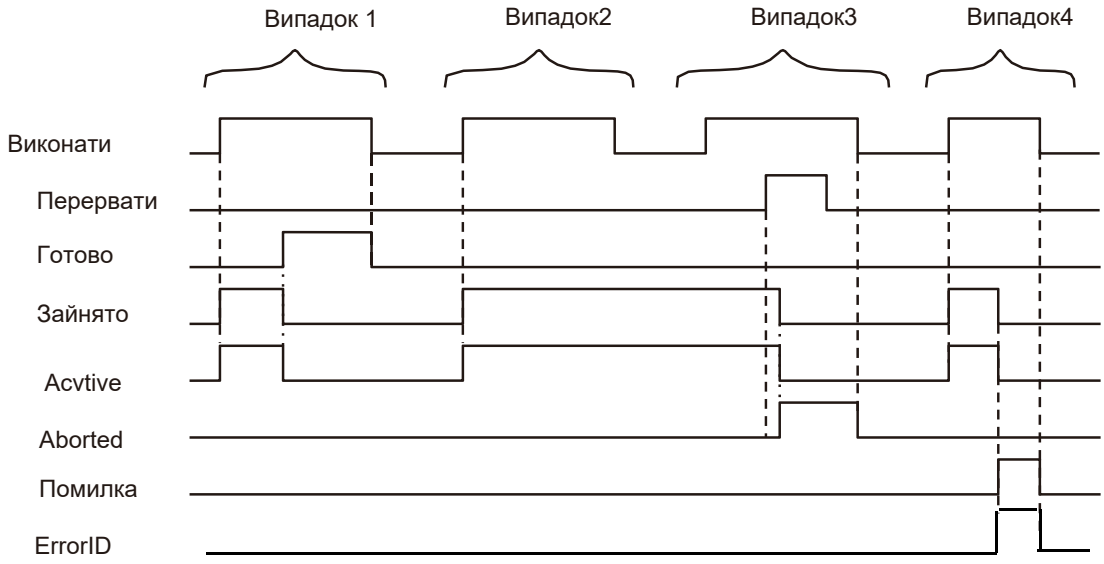
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	ІСТИНА, коли одноразове отримання даних сокета завершено в режимі одноразового отримання даних.	BOOL	ІСТИНА/НЕПРАВДА
Отримано	TRUE, коли отримання даних Socket завершено.	BOOL	ІСТИНА/НЕПРАВДА
Отримання	ІСТИНА, коли надходять дані сокета.	BOOL	ІСТИНА/НЕПРАВДА
Received_Length	Розмір фактично отриманих даних.	UINT	
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА/НЕПРАВДА
Активний	ІСТИНА, коли інструкція керує контролером для отримання даних.	BOOL	ІСТИНА/НЕПРАВДА
Перервано	TRUE, коли виконання інструкції перервано.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА/НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

■ **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Отримано	◆ Коли отримання однієї частини даних Socket завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли інструкція починає отримувати наступний фрагмент даних.
Отримання	◆ Під час отримання однієї частини даних Socket.	◆ Коли отримання однієї частини даних Socket завершено.
Зайняте	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Done* змінюється з FALSE на TRUE; ◆ Коли виконання інструкції перервано; ◆ Коли *помилка* змінюється з FALSE на TRUE.

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Активний	◆ Коли інструкція керує контролером для отримання даних.	◆ Коли *Done* змінюється з FALSE на TRUE. ◆ Коли виконання інструкції переривається. ◆ Коли *помилка* змінюється з FALSE на TRUE.
Перервано	◆ Коли виконання інструкції переривається.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1 Якщо інструкція використовується для отримання одного повідомлення або постійного отримання кількох повідомлень шляхом встановлення значення *DataSaveMode* на 0, *параметри Busy* та *Active* змінюються на TRUE, коли *Execute* змінюється з FALSE на TRUE. Значення *Done* змінюється на TRUE, а *Busy* та *Active* змінюються на FALSE, коли отримання частини даних закінчено або загальна довжина з'єднаних даних досягає або перевищує встановлену довжину. *Зроблено* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 2 *DataSaveMode* встановлено значення 1, *параметри Busy* і *Active* змінюються на TRUE, а *Execute* змінюється з FALSE на TRUE. Стани *«Зайнятий»* і *«Активний»* залишаються незмінними, а інструкція продовжує отримувати дані під час *виконання* змінюється з «ІСТИНА» на «ХИБНЯ».

Випадок 3 *Execute* знову встановлюється з FALSE на TRUE після випадку 2, і вихідний стан інструкції залишається незмінним. Встановіть *значення Abort* із FALSE на TRUE. Через один цикл *Aborted* змінюється з FALSE на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Abort* змінюється з TRUE на FALSE, вихід інструкції залишається незмінним. Коли *Execute* змінюється з TRUE на FALSE, *Aborted* змінюється на FALSE.

Випадок 4 Коли *Execute* змінюється з FALSE на TRUE, *Busy* і *Active* змінюються на TRUE. Коли виникає помилка, *Error* змінюється на TRUE, *ErrorID* показує відповідний код помилки, а тим часом *Busy*

і *Active* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* змінюється на 0.

- **функція**

ETH_Socket_Receive використовується для отримання даних Socket. Прошивка V1.02 і вище підтримує функцію.

1. Коли значення *ReceiveMode* встановлено на 0, контролер може постійно отримувати дані Socket. Коли значення *DataSaveMode* встановлено на 0, інструкція зберігає отримані дані в регістрах, починаючи з початкового регістра, зазначеного *Receive_Buffer_Address* у режимі сплайсингу. Контролер зберігає нові дані в регістрах після того, як зберігаються останні дані. Коли загальна довжина отриманих даних перевищує значення *Receive_Length* , вихід *Done* змінюється на TRUE, і виконання інструкції завершується.

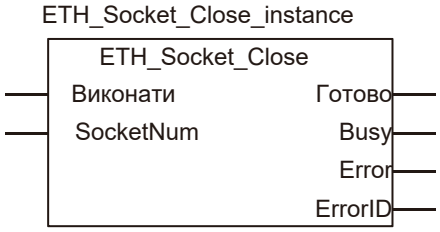
1. Коли значення вхідного параметра *ReceiveMode* дорівнює 0 і *DataSaveMode* дорівнює 1, інструкція зберігає отримані дані в регістрах, починаючи з початкового регістра, зазначеного *Receive_Buffer_Address* у режимі покриття. Щоразу, коли контролер зберігає отримані дані в регістрах, починаючи з початкового регістра, зазначеного *Receive_Buffer_Address* .

1. Коли довжина першого фрагмента отриманих даних перевищує довжину, визначену *Receive_Length* , контролер спочатку записує дані довжини, визначеної *Receive_Length* , на вказаний пристрій, дані, що перевищують встановлену довжину, відкидаються, а потім інструкція повідомляє про помилку .

1. У процесі виконання інструкції змініть значення вхідного параметра інструкції, а потім запустіть виконання інструкції, нове значення вхідного параметра не матиме чинності. Інструкцію потрібно перервати за допомогою вхідного параметра *Abort* , тоді нове значення вхідного параметра набуде чинності, коли інструкція буде виконана повторно.

8.14.2.12 ETH_Socket_Close

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Close використовується для відключення Socket.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
SocketNum	Вкажіть номер розетки.	USINT	1~8 (0)	Коли *Виконати* змінюється з FALSE на TRUE

Примітка. Функцію TCP-посилання та UDP можна вимкнути через ETH_Socket_Close.

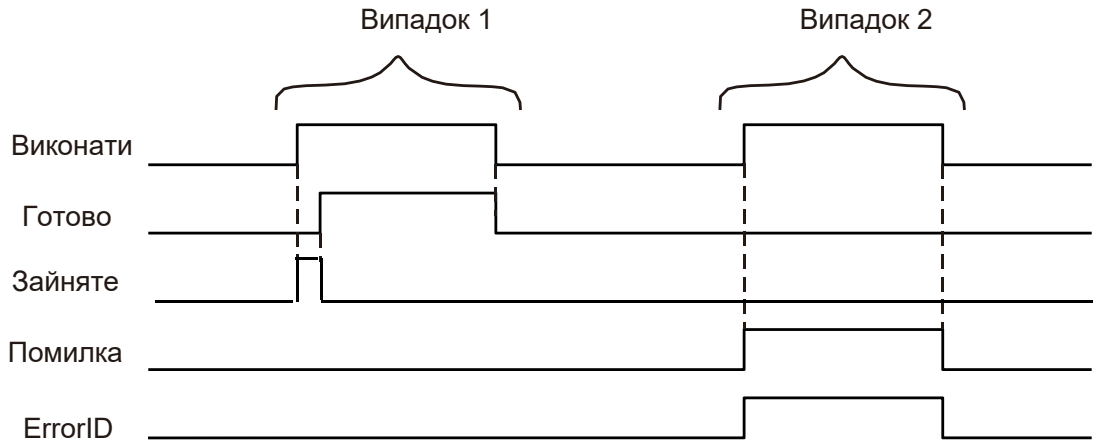
- **Час оновлення виводу**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	ІСТИНА/НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли виконання інструкції закінчено.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Зайняте	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Done* змінюється з TRUE на FALSE.
Помилка	◆ Коли виникає помилка під час виконання інструкції або введення параметри для інструкції є незаконними.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Час оновлення виводу**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і значення параметрів записуються успішно. Через один цикл *Done* змінюється на TRUE, а тим часом *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

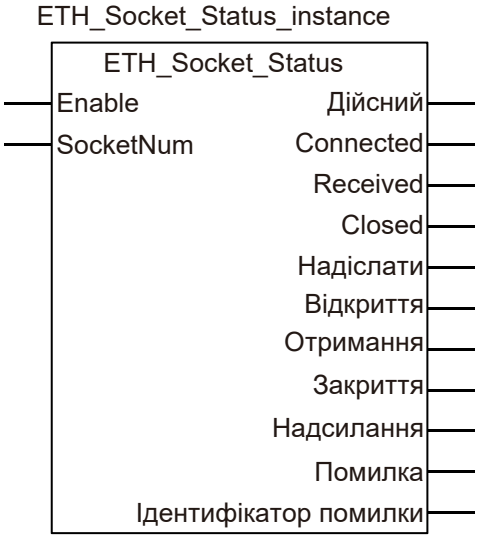
Випадок 2 : коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки, якщо записане значення параметра є недопустимим. Коли *Execute* змінюється на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* змінюється на 0.

- **функція**

ETH_Socket_Close використовується для роз'єднання TCP-з'єднання або вимкнення функції UDP. Прошивка V1.02 і вище підтримує цю функцію.

8.14.2.13 ETH_Socket_Status

FB/FC	Пояснення	Застосовна модель
FB	ETH_Socket_Status використовується для читання поточного стану вказаного сокета.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється на TRUE.	BOOL	Правда чи брехня (FALSE)	
SocketNum	Номер Сокета, який слід контролювати.	UINT	1~8 (0)	Коли *Увімкнути* змінюється на TRUE

Примітка: поточний стан Scocket вказаного номера та виникнення будь-якої помилки можна відстежувати за допомогою інструкції ETH_Socket_Status.

● **Час оновлення виводу**

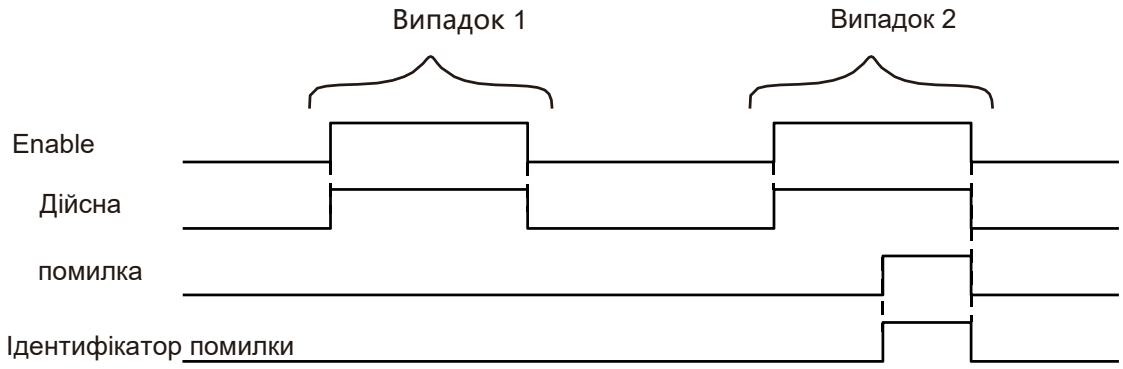
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли надсилання та отримання даних Socket завершено.	BOOL	ІСТИНА/НЕПРАВДА
Підключено	TRUE, коли підключення до цільового пристрою успішне.	BOOL	ІСТИНА/НЕПРАВДА
Отримано	TRUE, коли дані отримані успішно.	BOOL	ІСТИНА/НЕПРАВДА
ЗАЧИНЕНО	TRUE, коли посилання було закрито.	BOOL	ІСТИНА/НЕПРАВДА
Надіслати	TRUE, коли дані були надіслані.	BOOL	ІСТИНА/НЕПРАВДА
відкриття	ІСТИНА, коли здійснюється зв'язок із цільовим пристроєм.	BOOL	ІСТИНА/НЕПРАВДА
Отримання	TRUE, коли дані надходять.	BOOL	ІСТИНА/НЕПРАВДА
Закриття	TRUE, коли поточне посилання закривається.	BOOL	ІСТИНА/НЕПРАВДА
Відправка	ІСТИНА, коли надсилання або отримання даних закінчується.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли в інструкції виникає помилка	BOOL	ІСТИНА/НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
	виконання.		
ErrorID	Містить коди помилок, коли помилка виникає в виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється з FALSE на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE.
Підключено	◆ Коли Socket link завершено.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Closing* змінюється з FALSE на TRUE.
Отримано	◆ Після завершення отримання частини повідомлення.	◆ Під час *отримання* змінюється з FALSE на TRUE. ◆ Коли *Enable* змінюється з TRUE на FALSE.
ЗАЧИНЕНО	◆ Коли посилання Socket закрито.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *відкриття* змінюється з FALSE на TRUE.
Надіслати	◆ Коли надсилання частини повідомлення завершено.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *надсилання* змінюється з TRUE на FALSE.
відкриття	◆ Коли контролер починає створювати посилання Socket або очікує на підключення.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Connected* змінюється з FALSE на TRUE.
Отримання	◆ Коли контролер отримує частину повідомлення.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Received* змінюється з FALSE на TRUE.
Закриття	◆ Коли Socket вимкнено	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Closed* змінюється з FALSE на TRUE.
Відправка	◆ Коли надсилається фрагмент повідомлення.	◆ Коли *Enable* змінюється з TRUE на FALSE.
Помилка	◆ Коли виникає помилка виконання інструкції або помилка Socket.	◆ Коли *Enable* змінюється з TRUE на FALSE.

- **Час оновлення виводу**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* змінюється на FALSE.

Випадок 2 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. Коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, *ErrorID* показує коди помилок, а *Valid* зберігає стан TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *Error* змінюються на FALSE, а значення *ErrorID* змінюється на 0.

- **функція**

ETH_Socket_Status використовується для моніторингу поточного стану сокета вказаного номера. Прошивка V1.02 і вище підтримує функцію.

8.14.2.14 Приклад вільного протоколу Ethernet

Приклад програмування

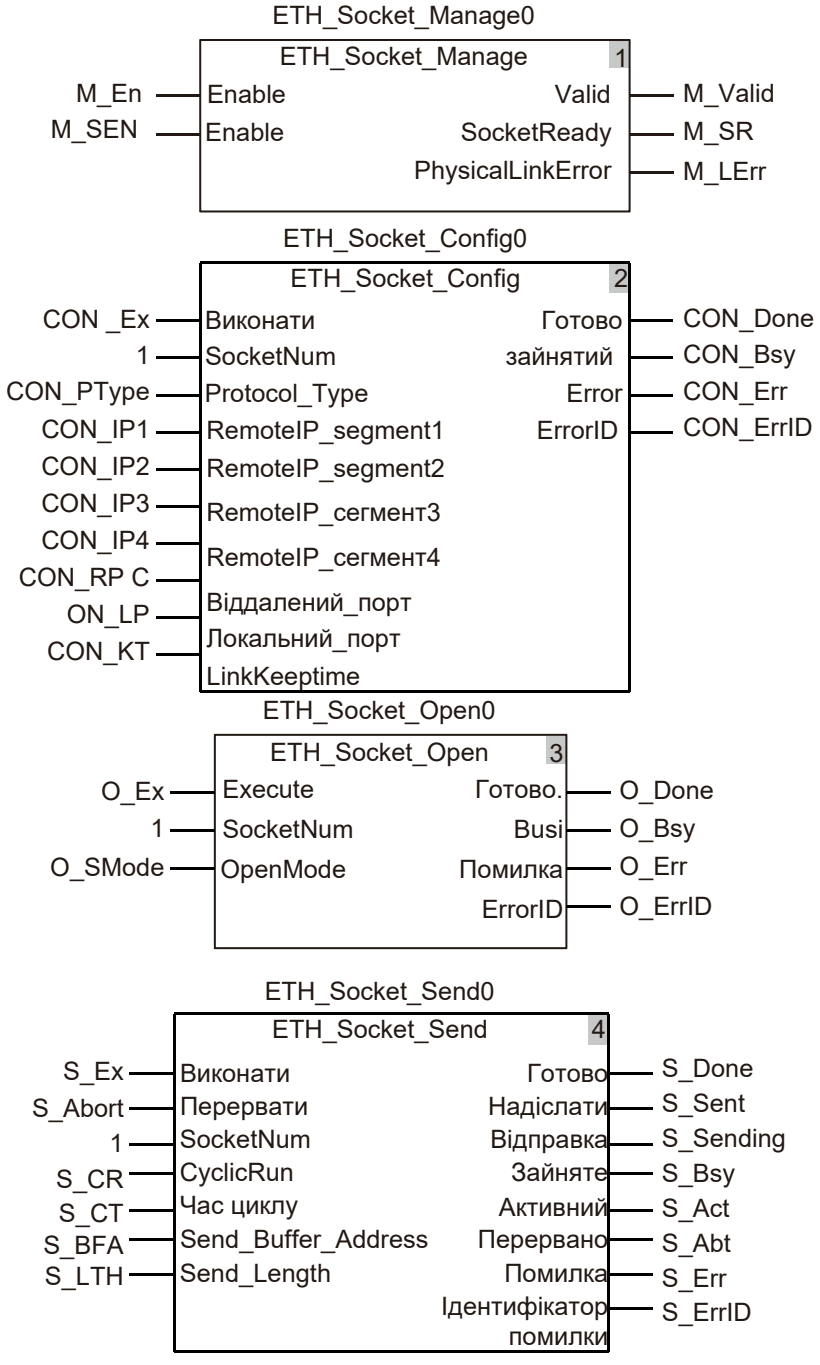
■ **Приклад використання інструкції ETH_Socket_Config:**

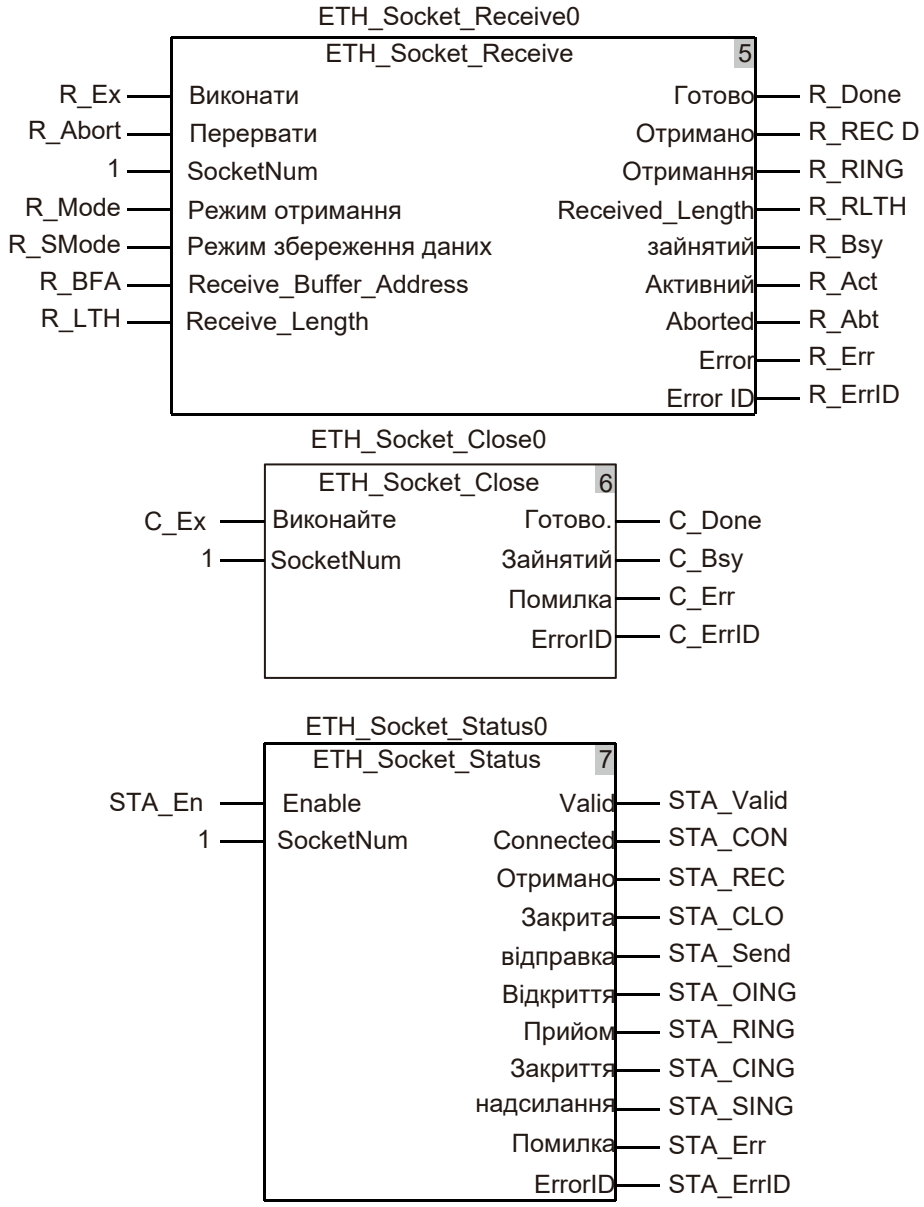
➢ **Таблиця змінних і програма**

Ім'я змінної	Адреса	Тип даних	Початкове значення
TEST_BEGIN		BOOL	
ETH_Socket_Manage0		ETH_Socket_Manage	
M_En		BOOL	
M_SEN		BOOL	
M_Valid		BOOL	
M_SR		BOOL	
M_LErr		BOOL	
ETH_Socket_Config0		ETH_Socket_Config	
CON_Ex		BOOL	
CON_PТип		USINT	1
CON_IP1		USINT	192
CON_IP2		USINT	168
CON_IP3		USINT	1
CON_IP4		USINT	10
CON_RP		UINT	502
CON_LP		UINT	502
CON_KT		UINT	30
CON_Готово		BOOL	
CON_Bsy		BOOL	
CON_Err		BOOL	
CON_ErrID		WORD	
ETH_Socket_Open0		ETH_Socket_Open	
O_Ex		BOOL	
O_SMode		BOOL	TRUE
O_Done		BOOL	
O_Bsy		BOOL	
O_Err		BOOL	
O_ErrID		WORD	
ETH_Socket_Send0		ETH_Socket_Send	
S_Ex		BOOL	
S_Abort		BOOL	
S_CR		BOOL	TRUE
S_CT		UINT	100

Ім'я змінної	Адреса	Тип даних	Початкове значення
S_BFA	%MB200	USINT	
S_LTH		UINT	17
S_Done		BOOL	
S_Sent		BOOL	
S_Sending		BOOL	
S_Bsy		BOOL	
S_Act		BOOL	
S_Abt		BOOL	
S_Err		BOOL	
S_ErrID		WORD	
ETH_Socket_Receive0		ETH_Socket_Receive	
R_Ex		BOOL	
R_Abort		BOOL	
R_Mode		USINT	0
R_SMode		USINT	1
R_BFA	%MB600	USINT	
R_LTH		UINT	100
R_Done		BOOL	
R_RECD		BOOL	
R_RING		BOOL	
R_RLTH		BOOL	
R_Bsy		BOOL	
R_Act		BOOL	
R_Abt		BOOL	
R_Err		BOOL	
R_ErrID		WORD	
ETH_Socket_Close0		ETH_Socket_Close	
C_Ex		BOOL	
C_Done		BOOL	
C_Bsy		BOOL	
C_Err		BOOL	
C_ErrID		WORD	
ETH_Socket_Status0		ETH_Socket_Status	
STA_En		BOOL	TRUE
STA_Valid		BOOL	
STA_CON		BOOL	
STA_REC		BOOL	
STA_CLO		BOOL	

Ім'я змінної	Адреса	Тип даних	Початкове значення
STA_Send		BOOL	
STA_OING		BOOL	
STA_RING		BOOL	
STA_CING		BOOL	
STA_SING		BOOL	
STA_Err		BOOL	
STA_ErrID		WORD	





■ **Опис операцій та обміну даними**

1. Змінні S_BFA і R_BFA відповідно прив'язані до адрес %MB200 і %MB600. Після успішної компіляції програми та завершення завантаження, програма

   виконується онлайн. Після активації онлайн стандартні дані MODBUS TCP « 00 00 00 00 00 0b 00 10 10 00 00 02 04 aa bb cc dd ″ записуються до 17 послідовних регістрів %MB, починаючи з %MB200.

2. Спочатку виконайте інструкцію ETH_Socket_Manage, встановивши для *M_En* значення TRUE, а потім установивши для *M_SEN* значення TRUE. Після того, як змінна *M_SR* зміниться на TRUE, виконайте ETH_Socket_Config, встановивши для змінної *CON_Ex* значення TRUE.

2. Після завершення виконання інструкції ETH_Socket_Config виконайте інструкцію ETH_Socket_Open, встановивши для *O_Ex* значення TRUE. Контролер починає створювати посилання на цільовий хост. Поточний стан підключення можна дізнатися, перевіривши вихід інструкції ETH_Socket_Status.

4. Після того, як контролер успішно побудує зв'язок із цільовим об'єктом, вихід *Connected* ETH_Socket_Status змінюється на TRUE. Потім виконайте ETH_Socket_Send, встановивши для *S_Ex* значення TRUE. Контролер циклічно надсилатиме 17 байт даних у регістрах, починаючи з %MB200, на цільовий хост.

5. Виконайте ETH_Socket_Receive, щоб отримати дані, які цільовий хост повертає або надсилає контролеру. У цьому прикладі контролер постійно отримуватиме та зберігатиме дані в режимі покриття.

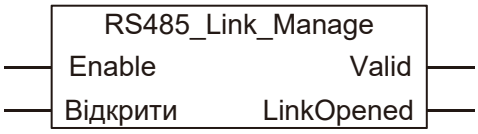
6. Інструкція ETH_Socket_Close використовується для відключення поточного посилання. Після того, як *C_Ex* встановлено на TRUE, контролер від'єднає з'єднання та припинить надсилання та отримання даних.

8.14.3 Інструкції щодо зв’язку RS485

8.14.3.1 RS485_Link_Manage

FB/FC	Пояснення	Застосовна модель
FB	RS485_Link_Manage використовується для вмикання або вимкнення зв’язку RS485.	AS516E-B AS532EST-B AS564EST-B

R S485_Link_Manage_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
ВІДЧИНЕНО	Зв'язок 485 увімкнено або вимкнено.	BOOL	Правда чи брехня (FALSE)	Коли *Enable* змінюється з FALSE на ІСТИНУ

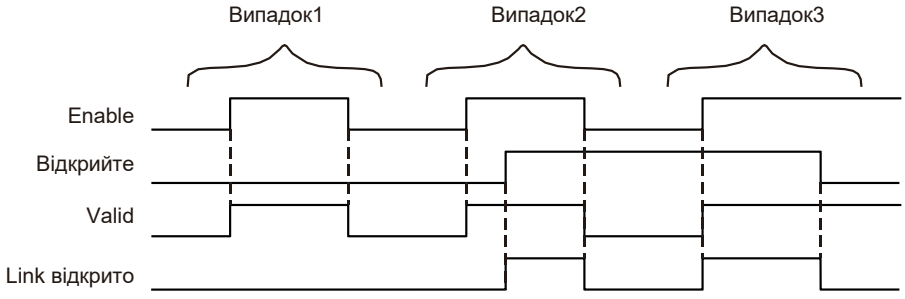
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
Посилання відкрито	ІСТИНА, коли зв’язок 485 увімкнено.	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE.
Посилання відкрито	◆ Коли *Enable* змінюється на TRUE, *Open* змінюється на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Open* змінюється з TRUE до FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, а Open — на FALSE, Valid змінюється на TRUE і

LinkOpened має значення FALSE.

Випадок 2 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. У цьому випадку LinkOpened змінюється на TRUE, оскільки Open змінюється з FALSE на TRUE. Коли Enable змінюється з TRUE на FALSE, Valid і LinkOpened змінюються на FALSE.

Випадок 3 : якщо Open має значення TRUE і Enable змінюється з FALSE на TRUE, значення Valid і *LinkOpened* змінюються на TRUE. Коли *Open* змінюється на FALSE, *LinkOpened* змінюється на FALSE.

- **функція:**

RS485_Link_Manage використовується для ввімкнення або вимкнення зв’язку RS485.
Прошивка V1.01 і вище підтримує цю функцію.

1. Вхід *Увімкнути* використовується для контролю того, чи вказівка набере чинності чи ні. Коли *Enable* змінюється на FALSE, робота інших параметрів інструкції буде недійсною. Коли *параметр «Увімкнути»* змінюється на «ІСТИНА», інструкція набуде чинності, а потім дія іншого параметра інструкції стане дійсною.

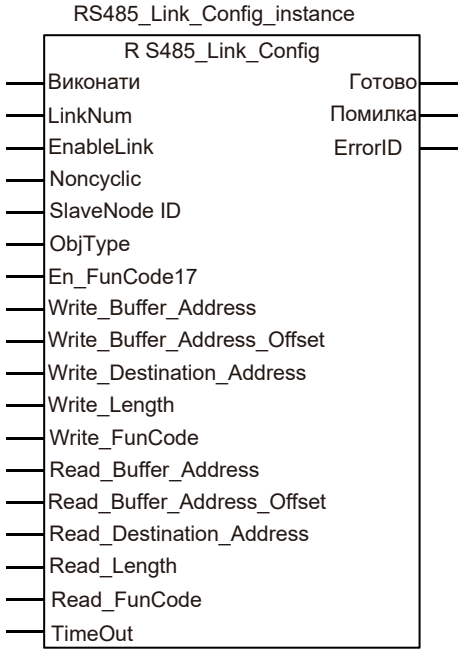
1. У випадку, якщо *Enable* має значення TRUE, зв’язок RS485 увімкнено, коли *Open* змінюється на TRUE. Зв'язок RS485 вимкнено, оскільки *Open* має значення FALSE.

1. Перш ніж увімкнути функцію зв'язку RS485, параметри інструкції RS485_Link_Config повинні бути налаштовані.
   В іншому випадку налаштовані параметри не будуть ефективними, якщо функцію RS485 увімкнути до того, як параметри інструкції RS485_Link_Config будуть налаштовані, якщо тільки функцію зв’язку RS485 не буде повторно ввімкнено.

4. Після налаштування кількох параметрів *LinkNum* за допомогою інструкцій RS485_Link_Config достатньо лише однієї інструкції RS485_Link_Manage, щоб увімкнути функцію зв’язку RS485. І всі конфігурації, визначені інструкціями RS485_Link_Config, призначені для обміну даними з підлеглим.

8.14.3.2 RS485_Link_Config

FB/FC	Пояснення	Застосовна модель
FB	RS485_Link_Config використовується для налаштування 485 параметрів зв'язку контролера руху.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
LinkNum	Вкажіть номер Посилання.	UINT	1~24 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
EnableLink	Увімкніть або вимкніть посилання.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Виконати* змінюється з FALSE на TRUE
Нециклічні	Встановіть режим зв'язку Link FALSE: циклічні, TRUE: нециклічні	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
SlaveNodeID	Укажіть ідентифікатор вузла підлеглого пристрою 485.	USINT	0~255 (0)	Коли *Виконати* змінюється з FALSE на TRUE
ObjType	Встановіть тип підлеглого регістра, який потрібно читати та записувати 0: реєстр слів 1: Бітовий регістр	USINT	0~1 (0)	Коли *Execute* змінюється з FALSE на TRUE
En_FunCode17	Встановіть код функції 17 для використання чи ні.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Виконати* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Write_Buffer_Address	Початковий регістр майстра, де зберігаються надіслані дані	UINT	%MW0 ~ %MW32767 %QW0 ~ %QW63	Коли *Виконати* змінюється з FALSE на TRUE
Write_Buffer_Address_Offset	Встановіть зміщення початкового регістру головних даних надсилання. Значення параметра 1 означає, що зсув становить 1 слово, якщо дані записуються на адресу слова підлеглого пристрою. Значення параметра 1 означає, що зміщення становить 1 біт, якщо дані записуються на бітову адресу веденого пристрою.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
Write_Destination_Address	Початкова адреса підлеглого пристрою MODBUS, який отримує дані від головного	UINT	Стандартна адреса Modbus	Коли *Execute* змінюється з FALSE на TRUE
Написати_довжину	Довжина даних для запису	UINIT	Реєстр слів: 0~100 Бітовий регістр: 0~256	Коли *Execute* змінюється з FALSE на TRUE
Write_FunCode	Вкажіть код функції для запису даних у регістри.	USINT	Бітовий регістр: 05, 16#0F Реєстр слів: 06, 16#10 （ 16#10 ）	Коли *Execute* змінюється з FALSE на TRUE
Read_Buffer_Address	Початковий регістр майстра, де зберігаються отримані дані.	UINT	%MW0~%MW327 67 %QW0~%QW63	Коли *Execute* змінюється з FALSE на TRUE
Read_Buffer_Address_Offset	Встановіть зміщення початкового регістру головних даних прийому. Значення налаштування 1 означає, що зсув становить 1 слово, якщо зчитується регістр слів підлеглого пристрою. Значення параметра 1 означає, що зсув становить 1 біт, якщо зчитується бітовий регістр веденого пристрою.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
Read_Destination_Address	Встановіть початкову адресу підлеглого пристрою MODBUS для зчитування провідним.	UINT	Стандартна адреса Modbus	Коли *Виконати* змінюється з FALSE на TRUE
Read_Length	Довжина даних для читання	UINT	Реєстр слів: 0~100 Бітовий регістр: 0~256	Коли *Execute* змінюється з FALSE на TRUE
Read_FunCode	Встановіть код функції для читання даних у регістрах.	USINT	Бітовий регістр: 01~02 Регістр слів: 03~04 （ 03 ）	Коли *Execute* змінюється з FALSE на TRUE
Час вийшов	Встановіть час, протягом якого головний пристрій очікує відповіді підлеглого. Тайм-аут підпорядкованого пристрою виникає, якщо підлеглий не відповідає на запит головного протягом встановленого часу. Одиниця: мс	UINT	Ціле число більше 0 (300)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. Коли інструкція без введення Write_FunCode читає та записує дані в регістрі Word відповідно до вибору *ObjType* , код функції за замовчуванням для запису даних – 16#10, а код функції за замовчуванням для читання даних – 03. Коли *ObjType* визначає бітовий регістр, код функції за замовчуванням для запису даних – 16#0F, а код функції для читання даних, 01 або 02, вибирається залежно від типу бітових регістрів згідно з інструкцією до відповідного продукту.
2. Прошивка, пізніша за V1.02, підтримує введення Write_FunCode інструкції.
3. Для інструкції з введенням Write_FunCode бітовий регістр або регістр Word можна вибрати через *ObjType* , код функції для запису даних можна вказати через вхід Write_FunCode, а код функції для читання даних можна вказати через вхід Read_FunCode.

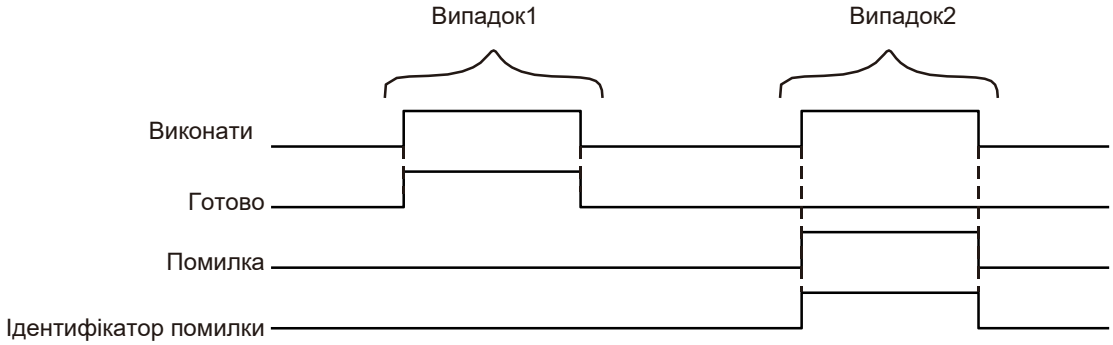
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE Якщо конфігурація параметрів Link успішна.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	ІСТИНА / НЕПРАВДА
Ідентифікатор помилки	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли конфігурація параметрів правильна, і *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALE.
Помилка	◆ Якщо конфігурація параметрів неправильна, і *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALE. ◆ Коли *Execute* запускається повторно після правильної зміни конфігурації параметра.

- **Діаграма часу оновлення вихідних даних**



- Якщо конфігурація параметрів правильна, *Done* змінюється з FASLE на TRUE як *Execute* змінюється з FASLE на TRUE.
- Якщо в конфігурації параметрів виникає помилка, *Done* означає FASLE, а *Error* змінюється з FALSE на TRUE, а *Execute* змінюється з FASLE на TRUE.

- **функція**

RS485_Link_Config використовується для налаштування параметрів для каналу зв’язку 485. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

1. *ObjType* — це тип даних прочитаного та записаного параметра. Коли *ObjType* дорівнює 0, це означає читання та запис реєстру слів підлеглого. Діапазон Write_Length і Read_Length становить 0~100, а значення Write_Length і Read_Length не можуть бути 0 одночасно.
   Коли *ObjType* дорівнює 1, це означає читання та запис бітового регістру підлеглого. Діапазон Write_Length і Read_Length становить 0~256, а значення Write_Length і Read_Length не можуть бути 0 одночасно. Для локальних адрес ви можете безпосередньо заповнювати адреси, змінні, поєднані з адресами, масиви, поєднані з початковими адресами.

2. Щоб локальна адреса заповнювала адресу %MB, ви можете заповнити лише молодший байт адреси %MW, а не старший байт.
3. Для адреси реєстру підлеглого пристрою ви можете безпосередньо заповнити адресу та змінну MODBUS.
4. Для зміщення в слововій операції фактична адреса обчислюється за словом. Для зсуву в бітовій операції фактична адреса обчислюється по біту.
   Наприклад:
   - Фактична адреса, розрахована через зсув слова:
     Коли адреса дорівнює %MW0 і зміщення дорівнює 15, фактична адреса операції дорівнює %MW15=%MW (0+15).
   - Фактична адреса, розрахована за допомогою бітового зсуву:
     Якщо адреса дорівнює %QW0, а зміщення дорівнює 7, фактична адреса операції дорівнює %QX0.7.
5. Якщо ви вирішите читати та записувати регістр слів підлеглого пристрою, регістр %MW локального пристрою можна розглядати як регістр зберігання для читання та запису даних. Діапазон регістра зберігання становить %MW0~%MW32767. Якщо діапазон перевищено або використовується інший регістр, в інструкції виникне помилка.

6. Якщо ви вирішите читати та записувати бітовий регістр підлеглого пристрою, регістри %MW та %QW локального пристрою можна розглядати як регістри зберігання для читання та запису даних. Діапазони регістра зберігання: %MW0~%MW32767 і %QW0~%QW63. Якщо діапазони перевищені або використовується інший регістр, в інструкції виникне помилка.

7. Встановлені значення параметрів цієї інструкції дійсні лише під час роботи ПЛК. Коли контролер руху повторно вмикається після вимкнення живлення, всі параметри, налаштовані до вимкнення живлення, стають недійсними. Інструкцію ETH_Link_Config потрібно виконати повторно, якщо потрібно використовувати конфігурацію перед вимкненням живлення.

8. Встановіть режим зв'язку через вхідний параметр *Noncyclic* . Режим зв’язку є циклічним, якщо для *параметра Noncyclic* встановлено значення FALSE. Режим зв’язку є нециклічним, якщо для *параметра «Нециклічний»* встановлено значення «ІСТИНА».

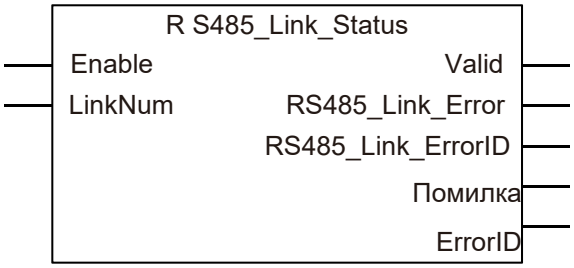
   a. Режим циклічного зв’язку: коли поточне з’єднання встановлено як режим циклічного зв’язку. Конфігурація для зв'язку набуде чинності негайно, а обмін даними між головним і підлеглим буде здійснюватися циклічно після ввімкнення функції зв'язку RS485 за допомогою інструкції RS485_Link_Manage.
   Конфігурація для каналу не буде виконана, і обмін даними з підлеглим припиняється після того, як функцію зв’язку RS485 буде вимкнено за допомогою інструкції RS485_Link_Manage.

   a. Режим нециклічного зв’язку: коли поточне з’єднання встановлено як режим нециклічного зв’язку. Конфігурація для зв'язку набуде чинності негайно, а обмін даними між головним і підлеглим буде проведено лише один раз після ввімкнення функції зв'язку RS485 за допомогою інструкції RS485_Link_Manage.
   Якщо обмін даними між головним і підлеглим має бути здійснений ще раз, повторно ввімкніть функцію зв’язку RS485 після того, як функцію зв’язку RS485 буде вимкнено за допомогою інструкції RS485_Link_Manage.

8.14.3.3 RS485_Link_Status

FB/FC	Пояснення	Застосовна модель
FB	RS485_Link_Status використовується, щоб спостерігати за тим, чи виникає помилка в посиланні 485, якому відповідає номер, або якщо підлеглий пристрій відповідає кодами помилок.	AS516E-B AS532EST-B AS564EST-B

RS485_Link_Status_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція перевіряється, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
LinkNum	Номер посилання для перегляду.	UINT	1~24 (Потрібно встановити значення змінної)	Коли *Enable* змінюється з FALSE на TRUE

- **Вихідні параметри**

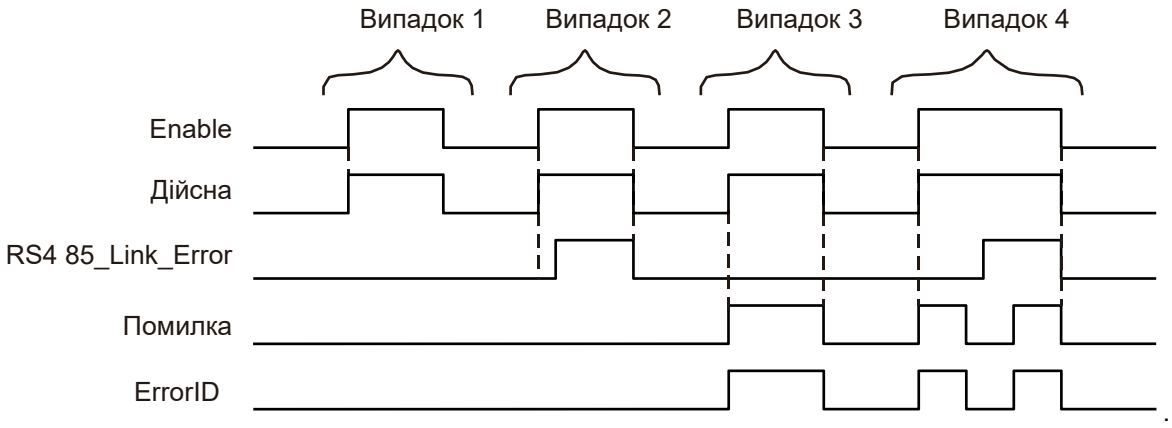
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
RS485_Link_Error	TRUE, коли виникає помилка у відповідному посиланні. Він дійсний, якщо *помилка* має значення FALSE.	BOOL	ІСТИНА / НЕПРАВДА
RS485_Link_ErrorID	Виводить код помилки зв'язку. Нижче наведено коди помилок та їх значення: 1: Невідомий код функції 2: Адреса у повідомленні відповіді від підпорядкованого пристрою відрізняється від налаштованої адреси. 3: Довжина отриманих даних у повідомленні-відповіді від підлеглого пристрою не відповідає налаштованій довжині. 4: Час очікування отримання даних 5: Помилка контрольної суми 6: налаштована довжина даних для читання та запису дорівнює 0. 7: Довжина фактично отриманих даних перевищує макс. довжина отриманих даних. 8: Час очікування надсилання даних 16#80+ код винятку: код винятку з підлеглого пристрою	WORD	
Помилка	TRUE, коли виникає помилка у введених інструкціях.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється з FALSE на TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE
RS485_Link_Error	◆ Коли під час зв’язку виникає помилка, параметр *Enable* має значення TRUE.	◆ Зв'язок відновлюється в нормальному режимі. (*Помилка* FALSE.) ◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Open* RS485_Link_Manage змінюється з TRUE на FALSE. (*Помилка* FALSE.)
Помилка	◆ Коли виникає помилка у вхідних інструкціях, *Enable* має значення TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли заповнено правильне значення параметра.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE. (Немає помилок як у зв’язку, так і в інструкціях.)

Випадок 2 : якщо під час зв’язку сталася помилка, *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE, а через деякий час *RS485_Link_Error* змінюється на TRUE.

Випадок 3 : якщо є помилка у введених інструкціях, коли *Enable* змінюється з FALSE на TRUE, *дійсний* змінюється з FALSE на TRUE, а *Error* змінюється на TRUE.

Випадок 4 : якщо є помилка як у зв’язку, так і в інструкціях, коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *RS485_Link_Error* все ще має значення FALSE. Після того, як значення вхідних параметрів змінено правильно, *Error* змінюється на FALSE, а через деякий час *RS485 _ Link _ Error* змінюється на TRUE. Коли помилка вхідного параметра виникає знову, *Error* змінюється на TRUE. На даний момент не буде змін у *RS485 _ Link _ Error* незалежно від того, чи зв'язок працює нормально. Коли *включено*

TRUE і *помилка* TRUE, стан *RS485 _ Link _ Error* не буде оновлено і буде недійсним.

- **функція**

  RS485_Link_Status використовується для перегляду стану зв'язку відповідного каналу. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

  1. Стан *RS485_Link_Error* не оновлюватиметься та буде недійсним, якщо *Enable* має значення TRUE і *Помилка* TRUE.
  2. Вхідне значення *LinkNum* може бути числом або змінною.

8.14.3.4 Приклад обміну даними RS485

Приклад програмування

- **Адреса вузла підлеглого DVP32ES2 дорівнює 1. Значення в 10 регістрах слів %MW100~%MW109 головного пристрою записуються в D0~D9 підлеглого пристрою. Потім значення в 20 регістрах слів D100~D119 підлеглого пристрою зчитуються та зберігаються в %MW0~%MW19 у головному пристрої, як показано в наступній таблиці змінних 1.**
  - **Таблиця змінних 1**

Ім'я змінної	Адреса	Тип даних	Початкове значення
Mng		RS485_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid		BOOL	
Mng_LOpen		BOOL	
Confg		RS485_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	0
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW100	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#1000
Confg _WL		UINT	10
Confg _WFC		USINT	16#10
Confg _RBA	%MW0	UINT	
Confg_RBAO		USINT	0
Confg _RDA		UINT	16#44C
Confg _RL		UINT	20
Confg _RFC		USINT	16#03
Confg _TOut		UINT	1000
Confg _Done		BOOL	
Confg _Err		BOOL	
Confg _ErrID		WORD	
LStatus		RS485_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid		BOOL	
LStatus_RLE		BOOL	

Ім'я змінної	Адреса	Тип даних	Початкове значення
LStatus_RLEID		BOOL	
LStatus_Err		BOOL	
LStatus_ErrID		WORD	

Після налаштування цих параметрів установіть Confg _Ex на TRUE, а потім Mng_Open на TRUE. Якщо конфігурацію необхідно змінити під час зв'язку, вхід Confg_Ex має бути викликаний наростаючим фронтом, Mng_Open має бути встановлено на FALSE, а потім на TRUE після встановлення нових конфігураційних даних.

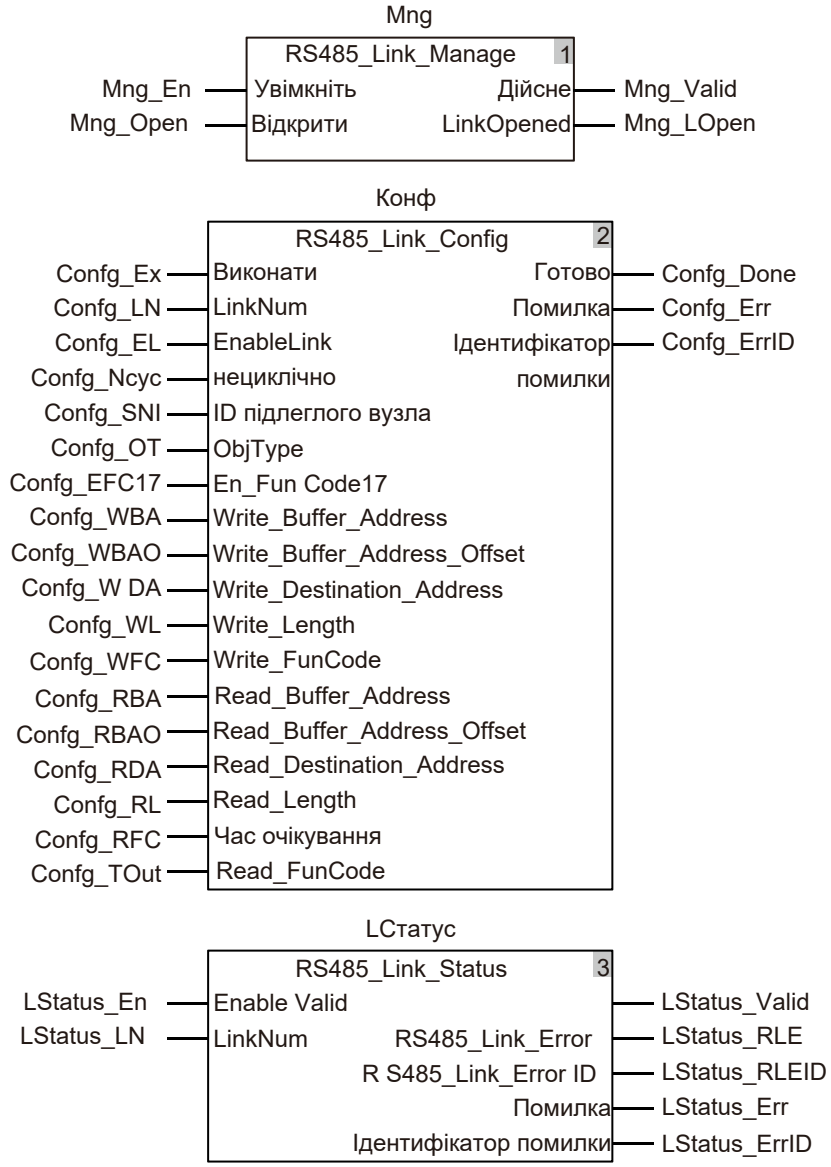
■ **Адреса вузла підлеглого DVP32ES2 дорівнює 1. Значення в 31-розрядних пристроях %MX0.0~%MX3.6 головного пристрою записуються в Y0~Y30 у підлеглому пристрої. Потім значення в Y0~Y30 підлеглого пристрою зчитуються та зберігаються в 31-розрядних пристроях %QX0.1~%QX3.7 у головному пристрої, як показано в наступній таблиці змінних 2.**

➢ **Таблиця змінних 2**

Ім'я	Адреса	Тип даних	Початкове значення
Mng		RS485_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid			
Mng_LOpen			
Confg		RS485_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	1
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW0	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#0400
Confg _WL		UINT	31
Confg _WFC		USINT	16#0F
Confg _RBA	%QW0	UINT	
Confg_RBAO		USINT	1
Confg _RDA		UINT	16#0400
Confg _RL		UINT	31
Confg _RFC		USINT	1
Confg _TOut		UINT	1000
Confg _Done			
Confg _Err			

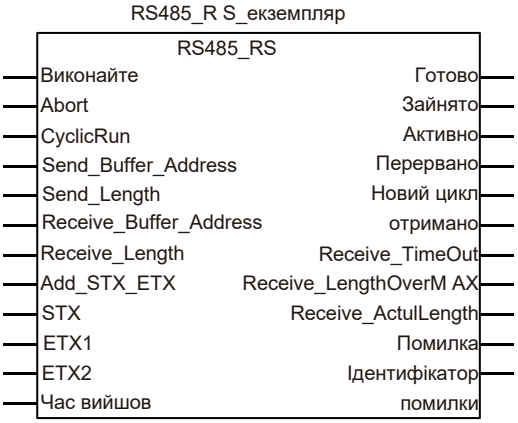
Ім'я	Адреса	Тип даних	Початкове значення
Confg _ErrID			
LCтатус		RS485_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid			
LStatus_RLE			
LStatus_RLEID			
LStatus_Err			
LStatus_ErrID			

➢ **програма**



8.14.3.5 RS485_RS

FB/FC	Пояснення	Застосовна модель
FB	RS485_RS використовується для налаштування параметрів зв’язку вільного протоколу RS485 для контролера руху.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Перервати	Зв’язок за вільним протоколом припиняється, коли *Abort* змінюється на TRUE.	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
CyclicRun	Налаштування режиму зв'язку для інструкції. НЕПРАВИЛЬНО: нециклічний; ІСТИНА: циклічний	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
Send_Buffer_Address	Вкажіть початковий регістр майстра, де зберігаються надіслані дані.	USINT	%MB0-%MB32767	Коли *Execute* змінюється з FALSE на TRUE
Send_length	Встановіть довжину надісланих даних.	UINT	0-200 (Байт) (0)	Коли *Виконати* змінюється з FALSE на TRUE
Receive_Buffer_Address	Вкажіть початковий регістр майстра, де зберігаються отримані дані.	USINT	%MB0-%MB32767	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Receive_Length	Встановіть довжину отриманих даних.	UINT	0-200 (байт) (0)	Коли *Execute* змінюється з FALSE на TRUE
Add_STX_ETX	Встановіть, чи додаватимуться початковий і кінцевий коди до надісланих повідомлень. НЕПРАВИЛЬНЕ: коди початку та кінця не додаються. TRUE: Додайте початковий і кінцевий коди.	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
STX	Встановити початок повідомлень.	USINT	0-16#7F (16#3A)	Коли *Виконати* змінюється з FALSE на TRUE
ETX1	Встановіть перше закінчення повідомлень.	USINT	0-16#7F (16#0D)	Коли *Виконати* змінюється з FALSE на TRUE
ETX2	Встановити другий кінець повідомлень.	USINT	0-16#7F (16#0A)	Коли *Виконати* змінюється з FALSE на TRUE
Час вийшов	Встановіть час очікування під час отримання даних.	UINT	0-32767 (0)	Коли *Виконати* змінюється з FALSE на TRUE

- **Вихідні параметри**

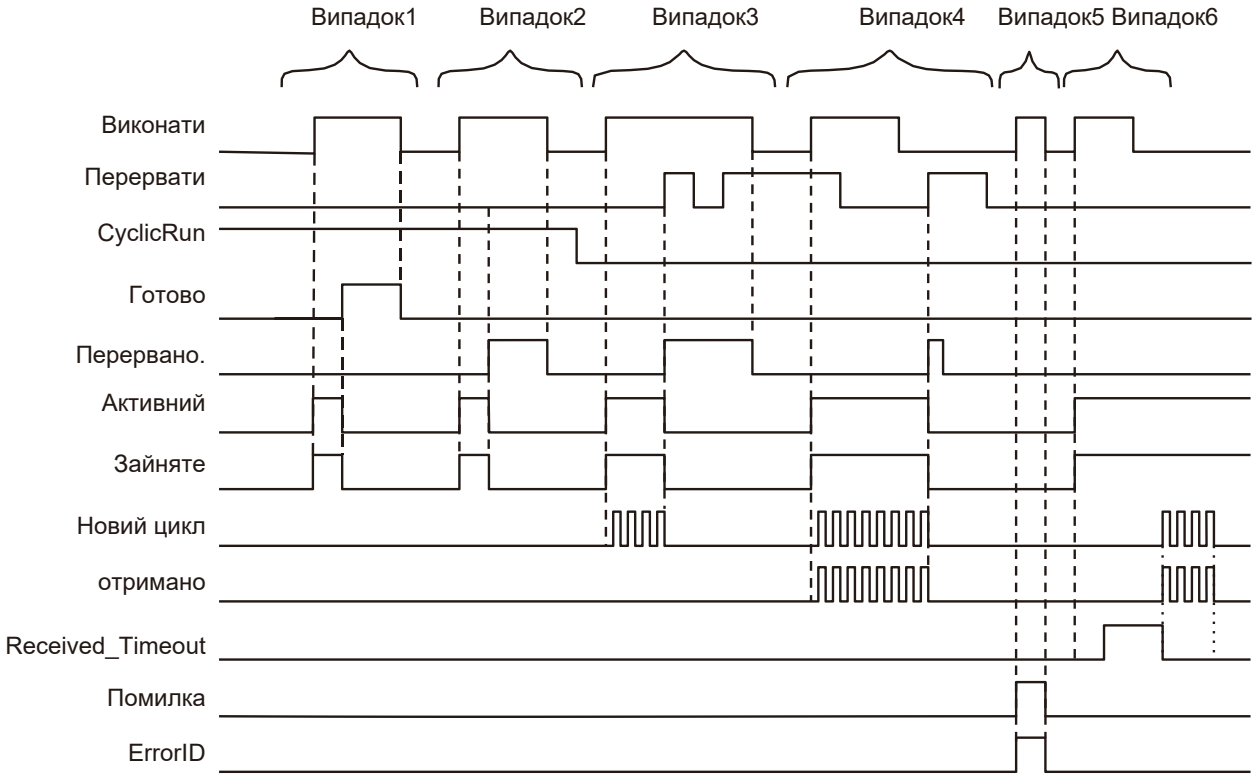
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли спілкування триває.	BOOL	ІСТИНА / НЕПРАВДА
Перервано	TRUE, коли поточне виконання інструкції перервано.	BOOL	ІСТИНА / НЕПРАВДА
Активний	ІСТИНА, коли портом керує інструкція.	BOOL	ІСТИНА / НЕПРАВДА
Новий цикл	ІСТИНА, коли виконання інструкції завершується один раз (у багатоцикловому стані)	BOOL	ІСТИНА / НЕПРАВДА
Отримано	ІСТИНА, коли прийом успішний.	BOOL	ІСТИНА / НЕПРАВДА
Receive_TimeOut	ІСТИНА, коли настає час очікування отримання повідомлення.	BOOL	ІСТИНА / НЕПРАВДА
Receive_LengthOverMAX	TRUE, коли довжина отриманих даних перевищує макс. дозволена довжина.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
Receive_ActulLength	Довжина фактично отриманих даних	UINT	0-200
Помилка	TRUE, коли виникає помилка конфігурації інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після того, як *Execute* змінюється з FALSE на TRUE, а *Busy* змінюється на TURE у робочому стані одного циклу	◆ Коли *Execute* змінюється з TRUE на FALSE у робочому стані одного циклу.
Зайняте	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Abort* змінюється на TRUE у багатоцикловому стані. ◆ Коли *Done* змінюється на TRUE у стані одного циклу.
Перервано	◆ Коли *Abort* змінюється з FALSE на TRUE вперше. ◆ Коли поточна інструкція переривається іншою інструкцією в багатоцикловому стані.	◆ Коли *Abort* змінюється з TRUE на FALSE. ◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли *Execute* має значення FALSE і *Abort* змінюється з FALSE на TRUE, *Aborted* змінюється на TRUE, а через один цикл змінюється на FALSE.
Активний	◆ Коли Execute змінюється з FALSE на TRUE.	◆ Коли Execute змінюється з TRUE на FALSE ◆ Коли поточна інструкція переривається іншою інструкцією в багатоцикловому стані. ◆ Коли Done змінюється на TRUE у стані одного циклу.
Новий цикл	◆ Коли ПЛК завершує один цикл роботи в багатотактовому стані.	◆ Коли інструкція переходить до наступного циклу отримання та відправлення даних.
Отримано	◆ Коли інструкція отримує відповідне повідомлення.	◆ Коли вводиться наступний цикл після інструкції, отримує відповідне повідомлення.
Receive_TimeOut	◆ Коли конфігурацію параметрів для отримання даних виконано, а відповідне повідомлення ні отримано протягом встановленого часу очікування.	◆ Коли отримано відповідне повідомлення.
Receive_LengthOverMAX	◆ Коли довжина фактично отриманих даних перевищує макс. дозволена довжина.	◆ Якщо довжина отриманих даних менша за макс. дозволена довжина.
Помилка	◆ Коли виникає помилка в конфігурації параметрів інструкції.	◆ Коли дані конфігурації інструкції правильні.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : у стані роботи з одним циклом *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Коли виконання інструкції завершено, *Busy* змінюється на FALSE, а *Done* змінюється на TRUE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

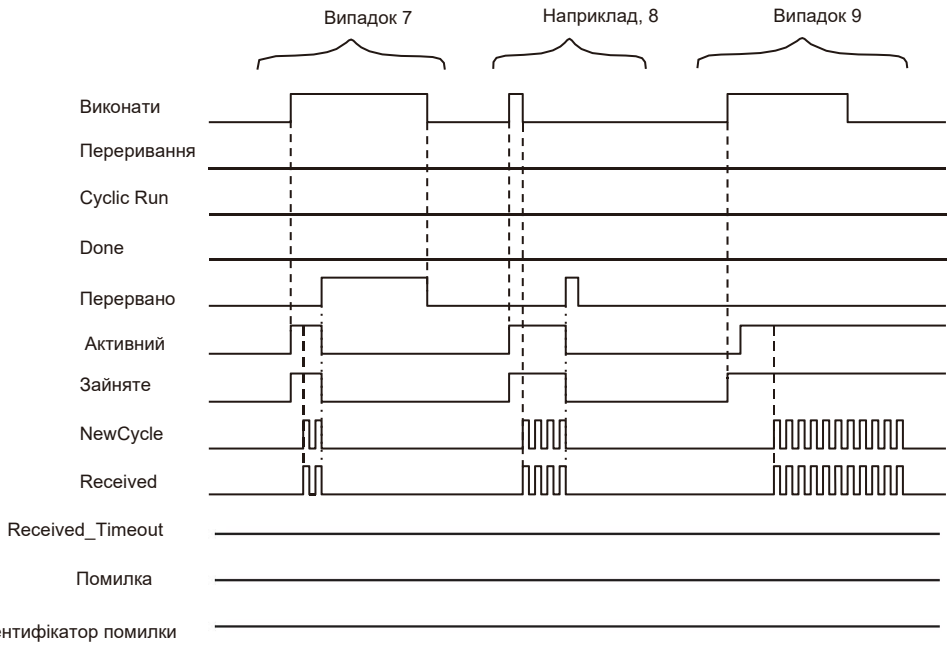
Випадок 2 : у робочому стані з одним циклом *значення Busy* і *Active* змінюються на FALSE, а *Abort* змінюється на TRUE, коли інструкція переривається іншою інструкцією. Тоді *Aborted* змінюється на FALSE, коли *Execute* змінюється на FALSE. Якщо *Execute* змінюється на FALSE до того, як інструкцію буде перервано, *Aborted* змінюється на TRUE протягом одного циклу.

Випадок 3 : у багатоцикловому стані, де налаштовано лише параметри надсилання даних, коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через деякий час *Newcycle* по черзі змінює значення TRUE і FALSE. Коли *Abort* змінюється на TRUE, *Busy* змінюється на FALSE, а *Aborted* на TRUE. Коли *Aborted* змінюється на FALSE, *Aborted* залишається TRUE. *Aborted* залишається TRUE, коли *Aborted* знову змінюється на TRUE. *Перервано* зміни на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : у стані з кількома циклами, коли параметри для отримання та надсилання даних налаштовано, а *параметр Abort* має значення TRUE, *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через деякий час як *Newcycle , так* і *Received* починають по черзі *змінюватись* між TRUE та FALSE. Під час повторного запуску *Abort після Execute* змінюється з TRUE на FALSE, *Aborted* змінюється на TRUE, через один цикл змінюється на FALSE, а інші змінюються на FALSE.

Випадок 5 : якщо є помилка в налаштованих параметрах інструкції, *Error* змінюється на TRUE, а *Execute* змінюється з FALSE на TRUE. *Помилка* змінюється на FALSE, коли *Execute* змінюється на FALSE.

Випадок 6 : у разі закінчення часу очікування отримання даних, коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через деякий час *Receive _ Timeout* змінюється на TRUE. Коли зв'язок відновлюється до нормального стану, *NewCycle* та *Received* починають по черзі змінювати значення TRUE та FALSE.



Випадок 7 : у стані з кількома циклами, коли *Execute* має значення TRUE, коли інструкція переривається іншою інструкцією, *Aborted* змінюється на TRUE, *Execute* змінюється на FALSE і *Aborted* змінюється на FALSE.

Випадок 8 : у багатоцикловому стані, де *Execute* має значення FALSE, коли інструкція переривається іншою інструкцією, *Aborted* змінюється на TRUE, а через один цикл змінюється на FALSE.

Випадок 9 : коли інструкція переривається іншою інструкцією, *Busy* змінюється на TRUE. Через деякий час *активні* зміни на TRUE та інші виходи дають відповідний вихід відповідно до наведених вище випадків.

- **функція**

RS485_RS використовується для конфігурації параметрів зв'язку за вільним протоколом RS485 і спостереження за станом зв'язку. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

1. RS485_RS не додає контрольну суму автоматично. У режимі ASCII надіслані дані мають бути повідомленням ASCII, у яке було перетворено.
2. Загальна довжина надісланих даних становить 200 байт. Довжина, встановлена в інструкції, не включає початок і кінець надісланих повідомлень.
3. Add_STX_ET встановлює, чи додаються коди верхнього та нижнього колонтитулів у надіслане повідомлення. Функція активна лише тоді, коли коди верхнього та нижнього колонтитулів надісланого повідомлення ідентичні кодам отриманого повідомлення. В іншому випадку функція не може бути ввімкнена, якщо коди верхнього та нижнього колонтитулів надісланого повідомлення відрізняються від кодів отриманого повідомлення, і в цьому випадку виникне помилка, якщо функцію ввімкнено.
   a. Код заголовка та код нижнього колонтитула, встановлені в інструкції, призначені для надісланих повідомлень. Якщо надіслане повідомлення правильне, але код заголовка та код нижнього колонтитула в отриманому повідомленні відрізняються від коду надісланого повідомлення, контролер не зможе отримати повідомлення, і час очікування відбудеться під час отримання.
   b. Код заголовка та код нижнього колонтитула, встановлені в інструкції, призначені для надісланого повідомлення. Тому підлеглий пристрій не дасть відповіді, якщо надіслане повідомлення не відповідає вимогам до підлеглого повідомлення.

   Таким чином, функція не може бути ввімкнена, якщо коди верхнього та нижнього колонтитулів надісланого повідомлення не ідентичні кодам отриманого повідомлення.
4. Коли RS485_RS і RS_485_Link_Config існують разом,
   a) Коли інструкція *Execute* of RS485_RS змінюється з FALSE на TRUE, функція 485Link ПЛК починає діяти.

b) Коли інструкція *Abort* RS485_RS змінюється з FALSE на TRUE, функція 485Link, яка виконується, знову набуде чинності.

c) Коли *Execute* інструкції RS485_RS змінюється з FALSE на TRUE, *Enable* RS485_Link_Manage змінюється на TRUE, *Open* інструкції може керувати функціями зв'язку 485 і вільного протоколу.

d) Якщо в програмі є декілька інструкцій RS485_RS, діятиме лише одна з них, яка була запущена востаннє.

e) Для повідомлення, яке надсилається через інструкцію RS485_RS, адреса, де зберігаються отримані дані, повинна бути встановлена в інструкції RS485_RS, якщо підлеглий пристрій надсилає відповідне повідомлення.

8.14.3.6 Приклад безкоштовного протоколу RS485

Приклад програмування 1

◆ **Включення функції початку і кінця повідомлень**

Безкоштовний протокол застосовується для надсилання стандартного повідомлення Modbus у режимі ASCII із початковою адресою %MB4000, де зберігаються надіслані дані, та %MB5000, де зберігаються отримані дані, і отримує дані відповіді від підлеглого пристрою.
Зміст повідомлення: 01 10 15 00 00 01 02 00 08.
Спочатку обчислюється контрольна сума CF. Потім повідомлення перетворюється на код ASCII.
Вміст повідомлення після перетворення: 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46.

Якщо параметри інструкції RS485_RS налаштовано на основі таблиці змінних 1, дані в %MB4000~%MB4019 є такими: 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46.

Дані повідомлення на шині: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A.

■ **Таблиця змінних 1**

Ім'я змінної	Адреса	Тип даних	Початкове значення
RS		RS485_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#30
RS_SL		UINT	20
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	1
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Abt		BOOL	
RS_Act		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

◆ **Відключення функції початку і кінця повідомлень**

1. Безкоштовний протокол застосовується для надсилання стандартного повідомлення Modbus у режимі ASCII із початковою адресою %MB4000, де зберігаються надіслані дані, та %MB5000, де зберігаються отримані дані, і отримує дані відповіді від підлеглого пристрою.

Зміст повідомлення: 01 10 15 00 00 01 02 00 08 ·

Спочатку обчислюється контрольна сума CF. Потім повідомлення перетворюється на код ASCII. Вміст повідомлення після перетворення: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38

43 46 0D 0A

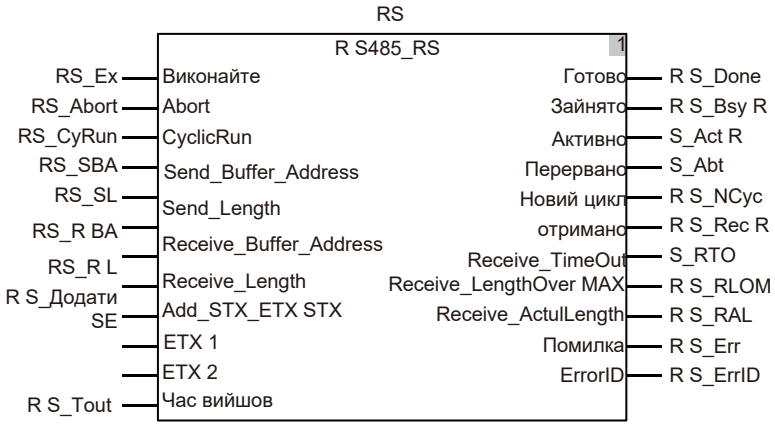
2. Якщо параметри інструкції RS485_RS налаштовані на основі таблиці змінних 2, дані в %MB4000~%MB4022 є 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

Дані повідомлення на шині: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

- **Таблиця змінних 2**

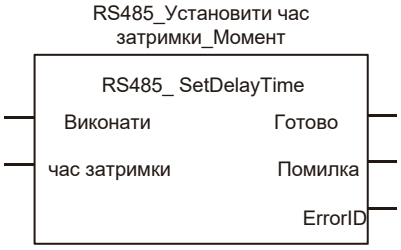
Ім'я змінної	Адреса	Тип даних	Початкове значення
RS		RS485_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#3A
RS_SL		UINT	23
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	0
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Abt		BOOL	
RS_Act		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

➢ **програма**


RS
R S485_RS
1
RS_Ex
Виконайте
RS_Abort
Abort
RS_CyRun
CyclicRun
RS_SBA
Send_Buffer_Address
RS_SL
Send_Length
RS_R BA
Receive_Buffer_Address
RS_R L
Receive_Length
R S_Додати SE
Add_STX_ETX STX
ETX 1
ETX 2
R S_Tout
Час вийшов
Готово
R S_Done
Зайнято
R S_Bsy R
Активно
S_Act R
Перервано
S_Abt
Новий цикл
R S_NCyc
отримано
R S_Rec R
Receive_TimeOut
S_RTO
Receive_LengthOver MAX
R S_RLOM
Receive_ActulLength
R S_RAL
Помилка
R S_Err
ErrorID
R S_ErrID

8.14.3.7 RS485_SetDelayTime

FB/FC	Пояснення	Застосовна модель
FB	RS485_SetDelayTime встановлює час затримки відповіді на комунікаційному порту RS485 контролера.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
Час затримки	Час затримки відповіді на комунікаційному порту RS485 контролера. Він дійсний незалежно від того, чи контролер служить головним чи підлеглим. (Одиниця: мс)	UINT	0~65535 (0)	Коли *Execute* змінюється на TRUE

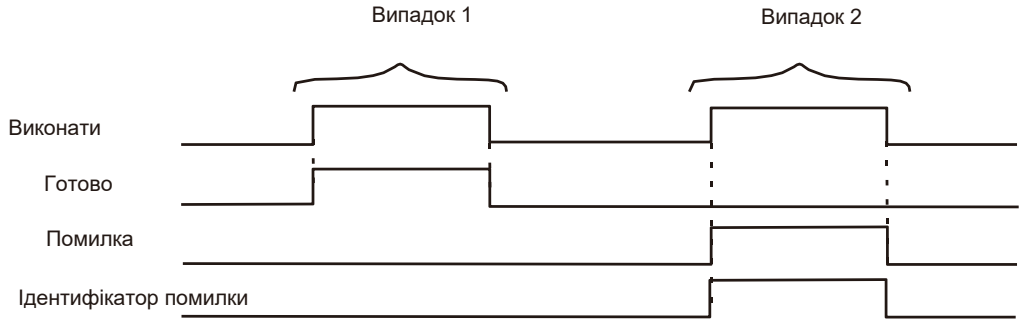
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли є помилка.	BOOL	ІСТИНА / НЕПРАВДА
Ідентифікатор помилки	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *DONE* змінюється з FALSE на TRUE.

Випадок 2 : коли виникає помилка, коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок. *Помилка* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

- **функція**

Інструкція RS485_SetDelayTime використовується для встановлення часу затримки відповіді комунікаційного порту RS485 контролера. Час налаштування дійсний, коли комунікаційний порт RS485 контролера працює як головний або підлеглий.

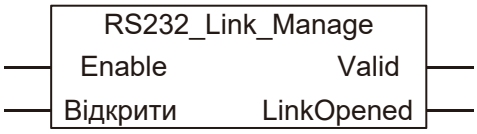
Коли комунікаційний порт RS485 працює як головний, він не надсилатиме наступне повідомлення, доки не мине встановлений час затримки після отримання даних відповіді від підлеглого. Коли комунікаційний порт RS485 працює як підлеглий, він не відповідатиме головному, доки не мине встановлений час затримки після отримання даних від головного.

8.14.4 Інструкції щодо зв’язку RS232

8.14.4.1 RS232_Link_Manage

FB/FC	Пояснення	Застосовна модель
FB	RS232_Link_Manage використовується для вмикання або вимкнення зв’язку RS232.	AS516E-B AS532EST-B AS564EST-B

RS232_Link_Manage_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
ВІДЧИНЕНО	232 зв'язок увімкнено або вимкнено.	BOOL	Правда чи брехня (FALSE)	Коли *Enable* змінюється з FALSE на ІСТИНУ

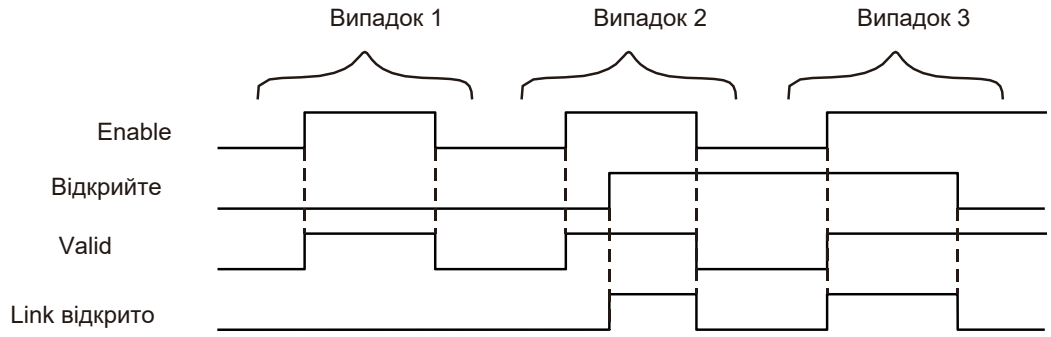
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
Посилання відкрито	ІСТИНА, коли зв’язок 232 увімкнено.	BOOL	ІСТИНА / НЕПРАВДА

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE
Посилання відкрито	◆ Коли *Enable* змінюється на TRUE і *Open* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *Open* змінюється з TRUE до FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Open* змінюється на FALSE, *Valid* змінюється на TRUE і *LinkOpened* змінюється на FALSE.

Випадок 2 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється на TRUE. У цьому випадку *LinkOpened* змінюється на TRUE, оскільки *Open* змінюється з FALSE на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *LinkOpened* змінюються на FALSE.

Випадок 3 : якщо *Open* має значення TRUE і *Enable* змінюється з FALSE на TRUE, *значення Valid* і *LinkOpened* змінюються на TRUE. Коли *Open* змінюється на FALSE, *LinkOpened* змінюється на FALSE.

- **функція**

RS232_Link_Manage використовується для ввімкнення або вимкнення зв’язку RS232. Прошивка V1.01 і вище підтримує цю функцію.

0. Вхід *Увімкнути* використовується для контролю того, чи вказівка набере чинності чи ні. Коли *Enable* змінюється на FALSE, робота інших параметрів інструкції буде недійсною. Коли *Enable* змінюється на TRUE, інструкція набуває чинності, а потім дія іншого параметра інструкції стає дійсною.

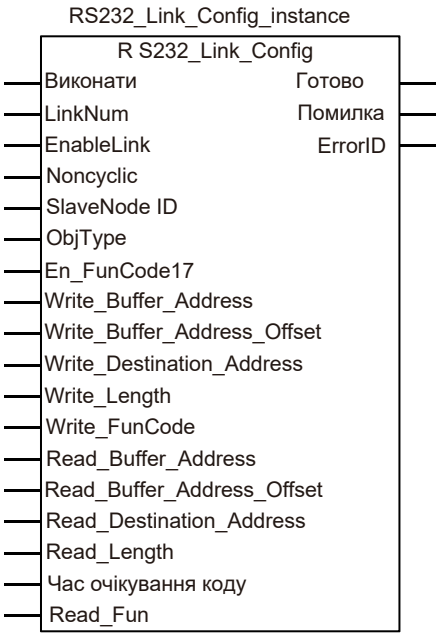
0. У випадку, якщо *Enable* має значення TRUE, зв’язок RS232 увімкнено, коли *Open* змінюється на TRUE. Зв'язок RS232 вимкнено, оскільки *Open* має значення FALSE.

0. Перш ніж увімкнути функцію зв'язку RS232, параметри інструкції RS232_Link_Config повинні бути налаштовані.
   В іншому випадку налаштовані параметри не будуть ефективними, якщо функцію RS232 увімкнено до того, як параметри інструкції RS232_Link_Config налаштовані, якщо тільки функцію зв’язку RS232 не буде повторно ввімкнено.

3. Після налаштування кількох параметрів *LinkNum* за допомогою інструкцій RS232_Link_Config достатньо лише однієї інструкції RS232_Link_Manage, щоб увімкнути функцію зв’язку RS232. І всі конфігурації, визначені інструкціями RS232_Link_Config, призначені для обміну даними з підлеглим.

8.14.4.2 RS232_Link_Config

FB/FC	Пояснення	Застосовна модель
FB	RS232_Link_Config використовується для налаштування параметрів зв'язку RS232 контролера руху.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
LinkNum	Вкажіть номер Посилання.	UINT	1~24 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
EnableLink	Увімкніть або вимкніть посилання.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Виконати* змінюється з FALSE на TRUE
Нециклічні	Встановіть режим зв'язку Link FALSE: циклічні, TRUE: нециклічні	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
SlaveNodeID	Встановіть ідентифікатор вузла підлеглого пристрою 232	USINT	0~255 (0)	Коли *Виконати* змінюється з FALSE на TRUE
ObjType	Встановіть тип підлеглого регістра, який потрібно читати та записувати. 0: реєстр слів 1: Бітовий регістр	USINT	0~1 (0)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
En_FunCode17	Встановіть, чи використовується код функції 17 чи ні.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Виконати* змінюється з FALSE на TRUE
Write_Buffer_Address	Початковий регістр майстра, де зберігаються надіслані дані	UINT	%MW0 ~ %MW32767 %QW0 ~ %QW63	Коли *Виконати* змінюється з FALSE на TRUE
Write_Buffer_Address_Offset	Встановіть зміщення початкового регістру головних даних надсилання. Значення налаштування 1 означає, що зміщення становить 1 слово, якщо записується адреса підлеглого слова. Значення параметра 1 означає, що зсув становить 1 біт, якщо записана бітова адреса веденого пристрою.	USINT	0~255 （ 0 ）	Коли *Execute* змінюється з FALSE на TRUE
Write_Destination_Адреса	Встановіть початкову адресу підлеглого пристрою MODBUS, який отримує дані від головного	UINT	Стандартна адреса Modbus	Коли *Execute* змінюється з FALSE на TRUE
Написати_довжину	Довжина даних для запису	UINIT	Регістр слів: 0~100 Бітовий пристрій: 0~256	Коли *Виконати* змінюється з FALSE на TRUE
Write_FunCode	Вкажіть код функції для запису даних у регістри.	USINT	Бітовий регістр: 05, 16#0F Регістр слів: 06, 16#10 （ 16#10 ）	Коли *Execute* змінюється з FALSE на TRUE
Read_Buffer_Address	Початковий регістр майстра, де зберігаються отримані дані.	UINT	%MW0~%MW32767 %QW0~%QW63	Коли *Виконати* змінюється з FALSE на TRUE
Read_Buffer_Address_Offset	Встановіть зміщення початкового регістру головних даних прийому. Значення налаштування 1 означає, що зсув становить 1 слово, якщо зчитується регістр слів підлеглого пристрою. Значення параметра 1 означає, що зсув становить 1 біт, якщо зчитується бітовий регістр веденого пристрою.	USINT	0~255 (0)	Коли *Execute* змінюється з FALSE на TRUE
Read_Destination_Адреса	Встановіть початкову адресу підлеглого пристрою MODBUS для зчитування провідним	UINT	Стандартна адреса Modbus	Коли *Execute* змінюється з FALSE на TRUE
Read_Length	Довжина даних для читання	UINT	Регістр слів: 0~100 Бітовий регістр: 0~256	Коли *Виконати* змінюється з FALSE на TRUE
Read_FunCode	Встановіть код функції для читання даних у регістрах.	USINT	Бітовий регістр: 01~02 Регістр слів: 03~04 （ 03 ）	Коли *Execute* змінюється з FALSE на TRUE
Час вийшов	Встановіть час очікування головним на підлеглих	UINT	Ціле число більше 0	Коли *Виконати* зміни від

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
	відповідь. Тайм-аут підпорядкованого пристрою виникає, якщо підлеглий не відповідає на запит головного протягом встановленого часу. Одиниця: мс		(300)	FALSE на TRUE

Примітка:

1. Коли інструкція без введення Write_FunCode читає та записує дані в регістрі Word відповідно до вибору *ObjType* , код функції за замовчуванням для запису даних – 16#10, а код функції за замовчуванням для читання даних – 03. Коли *ObjType* визначає бітовий регістр, код функції за замовчуванням для запису даних – 16#0F, а код функції для читання даних, 01 або 02, вибирається залежно від типу бітових регістрів згідно з інструкцією до відповідного продукту.
2. Прошивка, пізніша за V1.02, підтримує введення Write_FunCode інструкції.
3. Для інструкції з введенням Write_FunCode бітовий регістр або регістр Word можна вибрати через *ObjType* , код функції для запису даних можна вказати через вхід Write_FunCode, а код функції для читання даних можна вказати через вхід Read_FunCode.

- **Вихідні параметри**

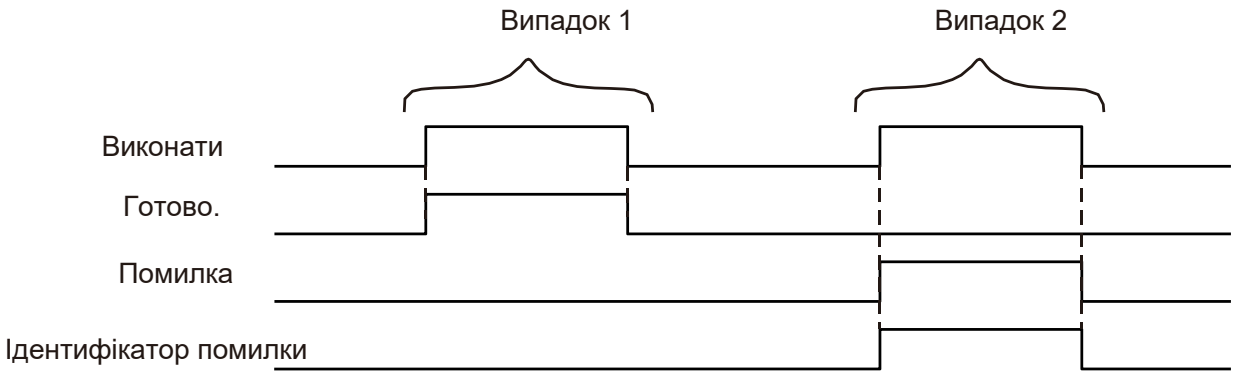
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE Якщо конфігурація параметрів Link успішна.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	ІСТИНА / НЕПРАВДА
Ідентифікатор помилки	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли конфігурація параметрів правильна, і *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE
Помилка	◆ Якщо конфігурація параметрів неправильна, і *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ Коли *Execute* запускається повторно після правильної зміни конфігурації параметра.

8

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : якщо конфігурація параметрів правильна, а *параметр «Виконати»* змінюється з FALSE на TRUE, *Виконано* зміни з FALSE на TRUE.

Випадок 2 : якщо конфігурація параметрів неправильна, а параметр « *Виконати»* змінюється з «ЛОЖЬ» на «ІСТИНА», *Готово* означає ЛОЖЬ, а *помилка* змінюється з ЛОЖЬ на ІСТИНУ.

- **функція**

RS232_Link_Config використовується для налаштування параметрів для каналу зв'язку 232. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

1. *ObjType* — це тип даних прочитаного та записаного параметра. Коли *ObjType* дорівнює 0, це означає читання та запис реєстру слів підлеглого. Діапазон Write_Length і Read_Length становить 0~100, а значення Write_Length і Read_Length не можуть бути 0 одночасно.
   Коли *ObjType* дорівнює 1, це означає читання та запис бітового регістру підлеглого. Діапазон Write_Length і Read_Length становить 0~256, а значення Write_Length і Read_Length не можуть бути 0 одночасно. Для локальних адрес ви можете безпосередньо заповнювати адреси, змінні, поєднані з адресами, масиви, поєднані з початковими адресами.

2. Щоб локальна адреса заповнювала адресу %MB, ви можете заповнити лише молодший байт адреси %MW, а не старший байт.
3. Для адреси реєстру підлеглого пристрою ви можете безпосередньо заповнити адресу та змінну MODBUS.
4. Для зміщення в слововій операції фактична адреса обчислюється за словом. Для зсуву в бітовій операції фактична адреса обчислюється по біту.
   Наприклад:
   Випадок 1 Фактична адреса, розрахована за допомогою зсуву слова:
   Коли адреса дорівнює %MW0 і зміщення дорівнює 15, фактична адреса операції дорівнює %MW15=%MW (0+15).
   Випадок 2. Фактична адреса, обчислена за допомогою бітового зсуву:
   Якщо адреса дорівнює %QW0, а зміщення дорівнює 7, фактична адреса операції дорівнює %QX0.7.

5. Якщо ви вирішите читати та записувати регістр слів підлеглого пристрою, регістр %MW локального пристрою можна розглядати як регістр зберігання для читання та запису даних. Діапазон регістра зберігання становить %MW0~%MW32767. Якщо діапазон перевищено або використовується інший регістр, в інструкції виникне помилка.
6. Якщо ви вирішите читати та записувати бітовий регістр підлеглого пристрою, регістри %MW та %QW локального пристрою можна розглядати як регістри зберігання для читання та запису даних. Діапазони регістра зберігання: %MW0~%MW32767 і %QW0~%QW63. Якщо діапазони перевищені або використовується інший регістр, в інструкції виникне помилка.

7. Встановлені значення параметрів цієї інструкції дійсні лише під час роботи ПЛК. Коли контролер руху повторно вмикається після вимкнення живлення, всі параметри, налаштовані до вимкнення живлення, стають недійсними. Інструкцію ETH_Link_Config потрібно виконати повторно, якщо потрібно використовувати конфігурацію перед вимкненням живлення.

8. Встановіть режим зв'язку через вхідний параметр *Noncyclic* . Режим зв’язку є циклічним, якщо для *параметра Noncyclic* встановлено значення FALSE. Режим зв’язку є нециклічним, якщо для *параметра «Нециклічний»* встановлено значення TRUE.

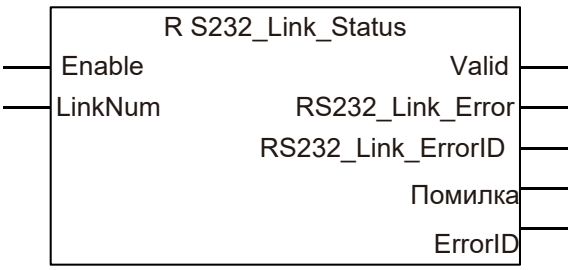
   a. Режим циклічного зв’язку: коли поточне з’єднання встановлено як режим циклічного зв’язку. Конфігурація для зв'язку набуде чинності негайно, а обмін даними між головним і підлеглим буде здійснюватися циклічно після ввімкнення функції зв'язку RS232 за допомогою інструкції RS232_Link_Manage.
   Конфігурація для каналу не буде виконана, а обмін даними з підлеглим припиняється після того, як функцію зв’язку RS232 буде вимкнено за допомогою інструкції RS232_Link_Manage.

   b. Режим нециклічного зв’язку: коли поточне з’єднання встановлено як режим нециклічного зв’язку. Конфігурація для зв’язку набуде чинності негайно, а обмін даними між головним і підлеглим буде проведено лише один раз після ввімкнення функції зв’язку RS232 за допомогою інструкції RS232_Link_Manage.
   Якщо обмін даними між головним і підлеглим має бути здійснений ще раз, повторно ввімкніть функцію зв’язку RS232 після того, як функцію зв’язку RS232 буде вимкнено за допомогою інструкції RS232_Link_Manage.

8.14.4.3 RS232_Link_Status

FB/FC	Пояснення	Застосовна модель
FB	RS232_Link_Status використовується, щоб спостерігати, чи виникає помилка в 232 посиланні, якому відповідає номер, або підлеглий пристрій відповідає кодами помилок.	AS516E-B AS532EST-B AS564EST-B

RS232_Link_Status_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція перевіряється, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
LinkNum	Номер посилання для перегляду.	UINT	1~24 (Потрібно встановити значення змінної)	Коли *Enable* змінюється з FALSE на TRUE

- **Вихідні параметри**

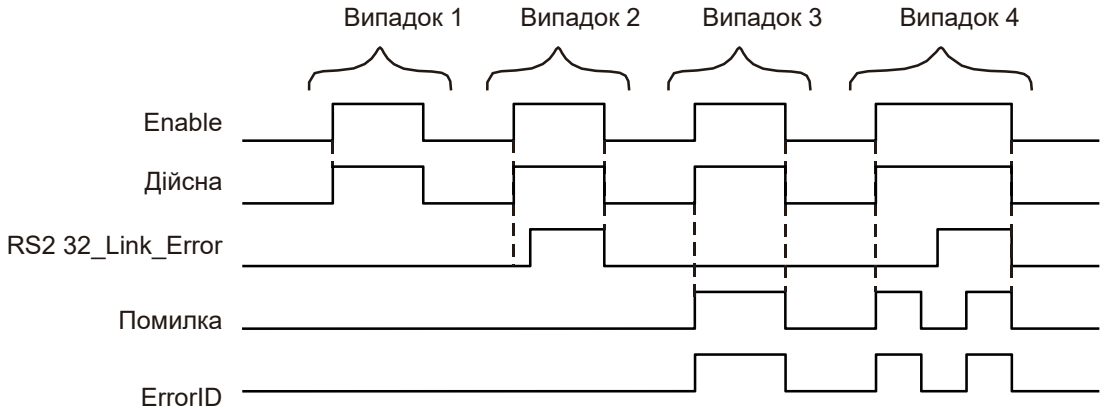
Назва параметра	функція	Тип даних	Допустимий діапазон
Дійсний	TRUE, коли результати інструкції дійсні.	BOOL	ІСТИНА / НЕПРАВДА
RS232_Link_Error	TRUE, коли виникає помилка у відповідному посиланні. Він дійсний, якщо *помилка* має значення FALSE.	BOOL	ІСТИНА / НЕПРАВДА
RS232_Link_ErrorID	Виводить код помилки зв'язку. Нижче наведено коди помилок та їх значення: 1: Невідомий код функції 2: Адреса у повідомленні відповіді від підпорядкованого пристрою відрізняється від налаштованої адреси. 3: Довжина отриманих даних у повідомленні-відповіді від підлеглого пристрою не відповідає налаштованій довжині. 4: Час очікування отримання даних 5: Помилка контрольної суми 6: налаштована довжина даних для читання та запису дорівнює 0. 7: Довжина фактично отриманих даних перевищує макс. довжина отриманих даних. 8: Час очікування надсилання даних 16#80+ код винятку: код винятку з підлеглого пристрою	WORD	
Помилка	TRUE, коли виникає помилка у введених інструкціях.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Дійсний	◆ Коли *Enable* змінюється з FALSE на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE.
RS232_Link_Error	◆ Коли під час зв'язку виникає помилка, параметр *Enable* має значення TRUE.	◆ Зв'язок відновлюється в нормальному режимі. (*Помилка* FALSE.) ◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли *Open* RS232_Link_Manage змінюється з TRUE на FALSE. (*Помилка* FALSE.)
Помилка	◆ Коли виникає помилка у вхідних інструкціях, *Enable* має значення TRUE.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ Коли заповнено правильне значення параметра.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE. (Немає помилок як у зв'язку, так і в інструкціях.)

Випадок 2 : якщо під час зв'язку сталася помилка, *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE, а через деякий час *RS232_Link_Error* змінюється на TRUE.

Випадок 3 : якщо є помилка у введених інструкціях, коли *Enable* змінюється з FALSE на TRUE, *дійсний* змінюється з FALSE на TRUE, а *Error* змінюється на TRUE.

Випадок 4 : якщо є помилка як у зв'язку, так і в інструкціях, коли *Enable* змінюється з FALSE на TRUE, *Valid* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *RS232_Link_Error* усе ще FALSE. Після того, як значення вхідних параметрів змінено правильно, *Error* змінюється на FALSE, а через деякий час *RS232 _ Link _ Error* змінюється на TRUE. Коли помилка вхідного параметра виникає знову, *Error* змінюється на TRUE. На даний момент не буде жодних змін у *RS232 _ Link _ Error* незалежно від того, чи зв'язок працює нормально. Якщо *Enable* має значення TRUE, а *Error* — TRUE, стан *RS232 _ Link _ Error* не оновлюватиметься та буде недійсним.

- **функція**

  RS232_Link_Status використовується для перегляду стану зв'язку відповідного каналу. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

  1. Стан *RS232_Link_Error* не оновлюватиметься та буде недійсним, якщо *Enable* має значення TRUE і *Помилка* TRUE.
  2. Вхідне значення *LinkNum* може бути числом або змінною.

8.14.4.4 Приклад обміну даними RS232

Приклад програмування

- **Адреса вузла підлеглого DVP32ES2 дорівнює 1. Значення в 10 регістрах слів %MW100~%MW109 головного пристрою записуються в D0~D9 підлеглого пристрою. Потім значення в 20 регістрах слів D100~D119 підлеглого пристрою зчитуються та зберігаються в %MW0~%MW19 у головному, як показано в наступній таблиці змінних 1.**

  - **Таблиця змінних 1**

Ім'я	Адреса	Тип даних	Початкове значення
Mng		RS232_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid		BOOL	
Mng_LOpen		BOOL	
Confg		RS232_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	0
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW100	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#1000
Confg _WL		UINT	10
Confg _WFC		USINT	16#10
Confg _RBA	%MW0	UINT	
Confg_RBAO		USINT	0
Confg _RDA		UINT	16#44C
Confg _RL		UINT	20
Confg _RFC		USINT	16#03
Confg _TOut		UINT	1000
Confg _Done		BOOL	
Confg _Err		BOOL	
Confg _ErrID		WORD	
LStatus		RS232_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid		BOOL	

Ім'я	Адреса	Тип даних	Початкове значення
LStatus_RLE		BOOL	
LStatus_RLEID		BOOL	
LStatus_Err		BOOL	
LStatus_ErrID		WORD	

Після налаштування цих параметрів установіть для Confg _Ex значення TRUE, а потім для Mng_Open значення TRUE. Якщо конфігурацію необхідно змінити під час зв'язку, вхід Confg_Ex має бути викликаний наростаючим фронтом, Mng_Open має бути встановлено на FALSE, а потім на TRUE після встановлення нових конфігураційних даних.

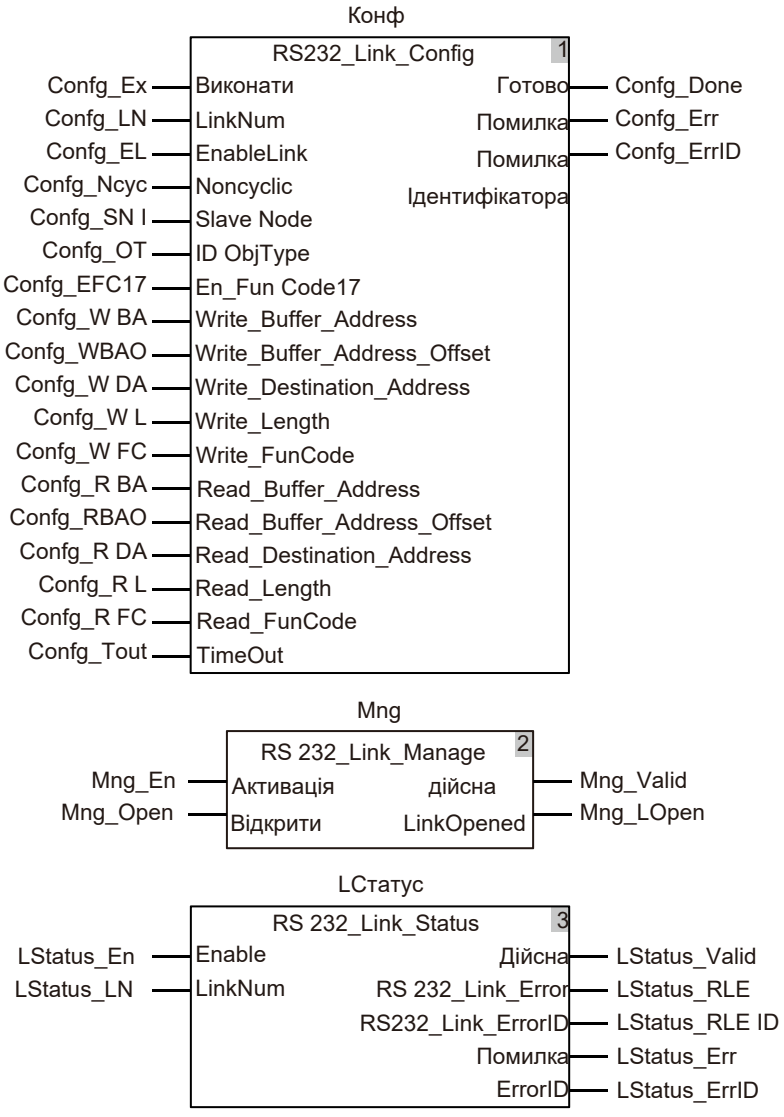
■ **Адреса вузла підлеглого DVP32ES2 дорівнює 1. Значення в 31-розрядних пристроях %MX0.0~%MX3.6 головного пристрою записуються в Y0~Y30 у підлеглому пристрої. Потім значення в Y0~Y30 підлеглого пристрою зчитуються та зберігаються в 31-розрядних пристроях %QX0.1~%QX3.7 у головному пристрої, як показано в наступній таблиці змінних 2.**

➢ **Таблиця змінних 2**

Ім'я	Адреса	Тип даних	Початкове значення
Mng		RS232_Link_Manage	
Mng_En		BOOL	TRUE
Mng_Open		BOOL	FALSE
Mng_Valid			
Mng_LOpen			
Confg		RS232_Link_Config	
Confg _Ex		BOOL	FALSE
Confg _LN		UINT	1
Confg _EL		BOOL	TRUE
Confg _Ncyc		BOOL	FALSE
Confg _SNI		USINT	1
Confg _OT		USINT	1
Confg _EFC17		BOOL	FALSE
Confg _WBA	%MW0	UINT	
Confg _WBAO		USINT	0
Confg _WDA		UINT	16#0400
Confg _WL		UINT	31
Confg _WFC		USINT	16#0F
Confg _RBA	%QW0	UINT	
Confg_RBAO		USINT	1
Confg _RDA		UINT	16#0400
Confg _RL		UINT	31
Confg _RFC		USINT	1
Confg _TOut		UINT	1000
Confg _Done			

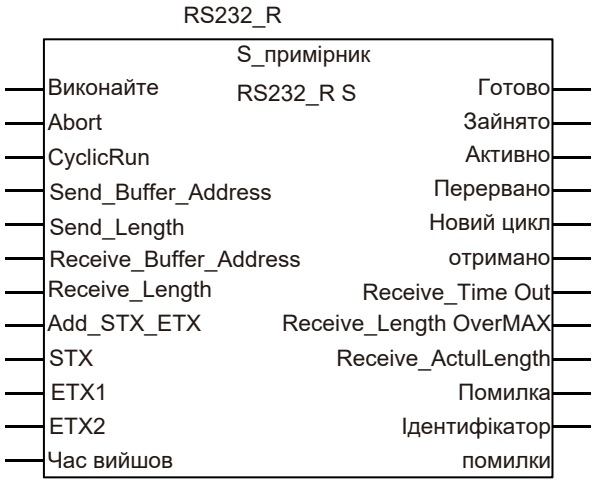
Ім'я	Адреса	Тип даних	Початкове значення
Confg _Err			
Confg _ErrID			
LСтатус		RS232_Link_Status	
LStatus_En		BOOL	TRUE
LStatus_LN		UINT	1
LStatus_Valid			
LStatus_RLE			
LStatus_RLEID			
LStatus_Err			
LStatus_ErrID			

➢ **програма**



8.14.4.5 RS232_RS

FB/FC	Пояснення	Застосовна модель
FB	RS232_RS використовується для налаштування параметрів зв'язку вільного протоколу RS232 для контролера руху.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Перервати	Зв'язок за вільним протоколом припиняється, коли *Abort* змінюється на TRUE.	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
CyclicRun	Налаштування режиму зв'язку для інструкції. НЕПРАВИЛЬНО: нециклічний; ІСТИНА: циклічний	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
Send_Buffer_Address	Вкажіть початковий регістр майстра, де зберігаються надіслані дані.	USINT	%MB0-%MB32767	Коли *Execute* змінюється з FALSE на TRUE
Send_length	Встановіть довжину надісланих даних.	UINT	0-200 (Байт) (0)	Коли *Виконати* змінюється з FALSE на TRUE
Receive_Buffer_Address	Вкажіть початковий регістр майстра, де зберігаються отримані дані.	USINT	%MB0-%MB32767	Коли *Execute* змінюється з FALSE на TRUE
Receive_Length	Встановіть довжину отриманих даних.	UINT	0-200 (байт) (0)	Коли *Виконати* змінюється з FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
				на ІСТИНУ
Add_STX_ETX	Встановіть, чи додаватимуться початковий і кінцевий коди до надісланих повідомлень. НЕПРАВИЛЬНЕ: коди початку та кінця не додаються. TRUE: Додайте початковий і кінцевий коди.	BOOL	Правда чи брехня (FALSE)	Коли *Execute* змінюється з FALSE на TRUE
STX	Встановити початок повідомлень.	USINT	0-16#7F (16#3A)	Коли *Виконати* змінюється з FALSE на TRUE
ETX1	Встановіть перше закінчення повідомлень.	USINT	0-16#7F (16#0D)	Коли *Виконати* змінюється з FALSE на TRUE
ETX2	Встановити другий кінець повідомлень.	USINT	0-16#7F (16#0A)	Коли *Execute* змінюється з FALSE на TRUE
Час вийшов	Встановіть час очікування під час отримання даних.	UINT	0-32767 (0)	Коли *Виконати* змінюється з FALSE на TRUE

- **Вихідні параметри**

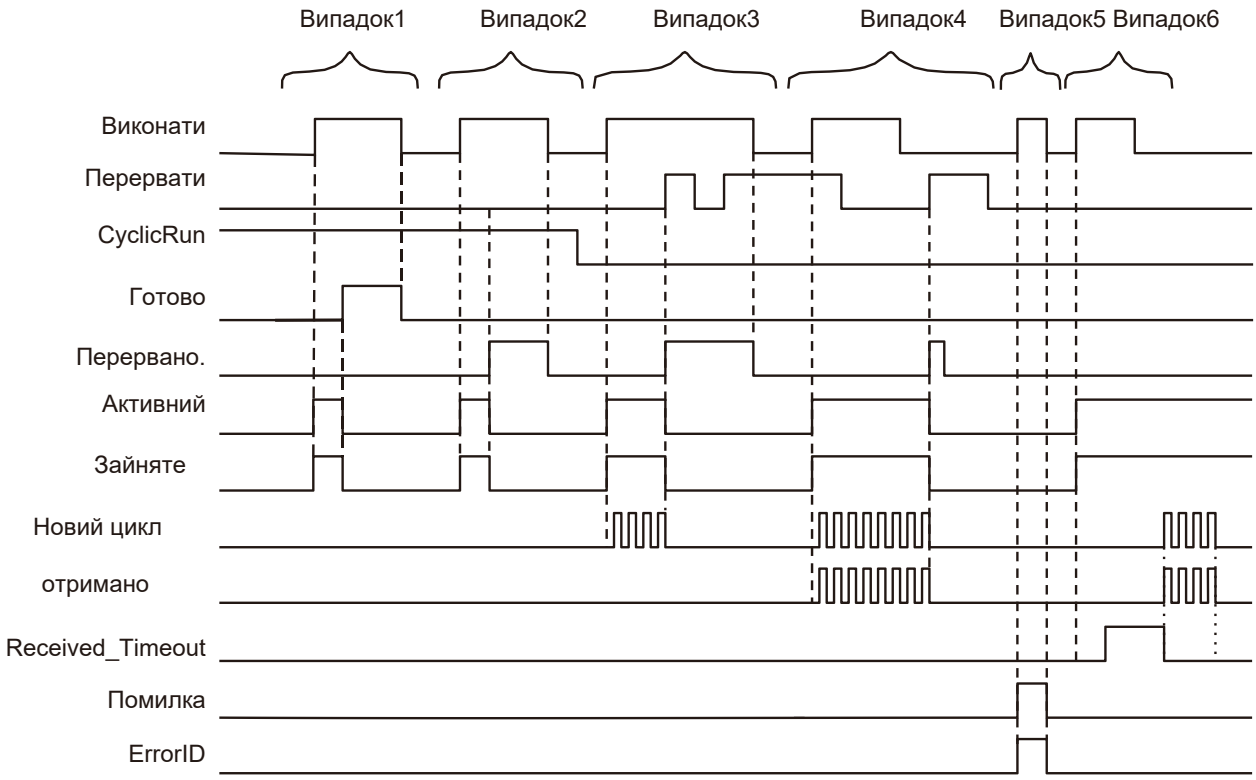
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли спілкування триває.	BOOL	ІСТИНА / НЕПРАВДА
Перервано	TRUE, коли поточне виконання інструкції перервано.	BOOL	ІСТИНА / НЕПРАВДА
Активний	ІСТИНА, якщо портом керує інструкція.	BOOL	ІСТИНА / НЕПРАВДА
Новий цикл	ІСТИНА, коли виконання інструкції завершується один раз (у багатоцикловому стані)	BOOL	ІСТИНА / НЕПРАВДА
Отримано	ІСТИНА, коли прийом успішний.	BOOL	ІСТИНА / НЕПРАВДА
Receive_TimeOut	ІСТИНА, коли настає час очікування отримання повідомлення.	BOOL	ІСТИНА / НЕПРАВДА
Receive_LengthOverMAX	TRUE, коли довжина отриманих даних перевищує макс. дозволена довжина.	BOOL	ІСТИНА / НЕПРАВДА
Receive_ActulLength	Довжина фактично отриманих даних	UINT	0-200
Помилка	TRUE, коли виникає помилка конфігурації інструкції.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок при виникненні помилки під час виконання інструкції. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після того, як *Execute* змінюється з FALSE на TRUE, а *Busy* змінюється на TURE у робочому стані одного циклу	◆ Коли *Execute* змінюється з TRUE на FALSE у робочому стані одного циклу.
Зайняте	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Abort* змінюється на TRUE у багатоцикловому стані. ◆ Коли *Done* змінюється на TRUE у стані одного циклу.
Перервано	◆ Коли *Abort* змінюється з FALSE на TRUE вперше.	◆ Коли *Abort* змінюється з TRUE на FALSE. ◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли *Execute* має значення FALSE і *Abort* змінюється з FALSE на TRUE, *Aborted* змінюється на TRUE, а через один цикл змінюється на FALSE.
Активний	◆ Коли *Execute* змінюється з FALSE на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ Коли поточна інструкція переривається іншою інструкцією в багатоцикловому стані. ◆ Коли *Done* змінюється на TRUE у стані одного циклу.
Новий цикл	◆ Коли ПЛК завершує один цикл роботи в багатоцикловому стані.	◆ Коли інструкція переходить до наступного циклу отримання та відправлення даних.
Отримано	◆ Коли інструкція отримує відповідне повідомлення.	◆ Коли вводиться наступний цикл після інструкції, отримує відповідне повідомлення.
Receive_TimeOut	◆ Після завершення налаштування параметрів для отримання даних і відповідне повідомлення не отримано протягом встановленого часу очікування.	◆ Коли отримано відповідне повідомлення.
Receive_LengthOverMAX	◆ Коли довжина фактично отриманих даних перевищує макс. дозволена довжина.	◆ Якщо довжина отриманих даних менша за макс. дозволена довжина.
Помилка	◆ Коли виникає помилка в конфігурації параметрів інструкції.	◆ Коли дані конфігурації інструкції правильні.

● Діаграма часу оновлення вихідних даних



Випадок 1 : у стані роботи з одним циклом *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Коли виконання інструкції завершено, *Busy* змінюється на FALSE, а *Done* змінюється на TRUE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

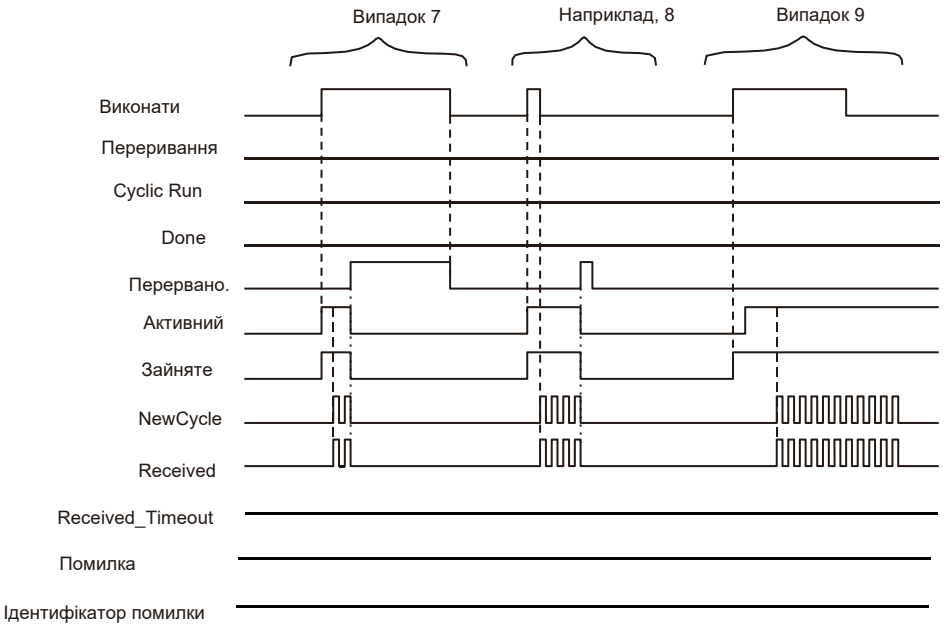
Випадок 2 : у робочому стані з одним циклом *значення Busy* і *Active* змінюються на FALSE, а *Abort* змінюється на TRUE, коли інструкція переривається іншою інструкцією. Тоді *Aborted* змінюється на FALSE, коли *Execute* змінюється на FALSE. Якщо *Execute* змінюється на FALSE до того, як інструкцію буде перервано, *Aborted* змінюється на TRUE протягом одного циклу.

Випадок 3 : у багатоцикловому стані, де налаштовано лише параметри надсилання даних, коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через деякий час *Newcycle* по черзі змінює значення TRUE і FALSE. Коли *Abort* змінюється на TRUE, *Busy* змінюється на FALSE, а *Aborted* на TRUE. Коли *Aborted* змінюється на FALSE, *Aborted* залишається TRUE. *Aborted* залишається TRUE, коли *Aborted* знову змінюється на TRUE. *Перервано* зміни на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : у стані з кількома циклами, коли параметри для отримання та надсилання даних налаштовано, а *параметр Abort* має значення TRUE, *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через деякий час як *Newcycle , так* і *Received* починають по черзі *змінюватись* між TRUE та FALSE. Коли після повторного запуску *Abort після Execute* значення TRUE змінюється на FALSE, *Aborted* змінюється на TRUE, через один цикл змінюється на FALSE, а інші змінюються на FALSE.

Випадок 5 : якщо є помилка в налаштованих параметрах інструкції, *Error* змінюється на TRUE як *Виконайте* зміни з FALSE на TRUE. *Помилка* змінюється на FALSE, коли *Execute* змінюється на FALSE.

Випадок 6 : у разі закінчення часу очікування отримання даних, коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через деякий час *Receive _ Timeout* змінюється на TRUE. Коли зв'язок відновлюється до нормального стану, *NewCycle* та *Received* починають по черзі змінювати значення TRUE та FALSE.



Випадок 1 : у стані з кількома циклами, коли *Execute* має значення TRUE, коли інструкція переривається іншою інструкцією, *Aborted* змінюється на TRUE, *Execute* змінюється на FALSE, а *Aborted* змінюється на FALSE.

Випадок 2 : у багатоцикловому стані, де *Execute* має значення FALSE, коли інструкція переривається іншою інструкцією, *Aborted* змінюється на TRUE, а через один цикл змінюється на FALSE.

Випадок 3 : коли інструкція переривається іншою інструкцією, *значення Busy* змінюється на TRUE. Через деякий час *активний*
зміни на TRUE та інші виходи дають відповідний вихід відповідно до наведених вище випадків.

- **функція**

  RS232_RS використовується для налаштування параметрів зв'язку за вільним протоколом RS232 і спостереження за станом зв'язку. Прошивка V1.01 і вище підтримує цю функцію.

- **Запобіжні заходи**

1. RS232_RS не додає контрольну суму автоматично. У режимі ASCII надіслані дані мають бути повідомленням ASCII, у яке було перетворено.
2. Загальна довжина надісланих даних становить 200 байт. Довжина, встановлена в інструкції, не включає початок і кінець надісланих повідомлень.
3. Add_STX_ET встановлює, чи додаються коди верхнього та нижнього колонтитулів у надіслане повідомлення. Функція активна лише тоді, коли коди верхнього та нижнього колонтитулів надісланого повідомлення ідентичні кодам отриманого повідомлення. В іншому випадку функція не може бути ввімкнена, якщо коди верхнього та нижнього колонтитулів надісланого повідомлення відрізняються від кодів отриманого повідомлення, і в цьому випадку виникне помилка, якщо функцію ввімкнено.
   a. Код заголовка та код нижнього колонтитула, встановлені в інструкції, призначені для надісланих повідомлень. Якщо надіслане повідомлення правильне, але код заголовка та код нижнього колонтитула в отриманому повідомленні відрізняються від коду надісланого повідомлення, контролер не зможе отримати повідомлення, і час очікування відбудеться під час отримання.
   b. Код заголовка та код нижнього колонтитула, встановлені в інструкції, призначені для надісланого повідомлення. Тому підлеглий пристрій не дасть відповіді, якщо надіслане повідомлення не відповідає вимогам до підлеглого повідомлення.

   Таким чином, функція не може бути ввімкнена, якщо коди верхнього та нижнього колонтитулів надісланого повідомлення не ідентичні кодам отриманого повідомлення.
4. Коли RS232_RS і RS_232_Link_Config існують разом,
   a. Коли інструкція *Execute* of RS232_RS змінюється з FALSE на TRUE, функція 232Link ПЛК починає діяти.
   b. Коли інструкція *Abort* RS232_RS змінюється з FALSE на TRUE, функція 232Link, яка виконується, знову почне діяти.

c. Коли *Execute* інструкції RS232_RS змінюється з FALSE на TRUE, *Enable* RS232_Link_Manage змінюється на TRUE, *Open* інструкції може керувати функціями зв'язку 232 зв'язку та вільного протоколу.

d. Якщо в програмі є декілька інструкцій RS232_RS, діятиме лише одна з них, яка була запущена останнім разом.

e. Для повідомлення, яке надсилається через інструкцію RS232_RS, адреса, де зберігаються отримані дані, повинна бути встановлена в інструкції RS232_RS, якщо підлеглий пристрій надсилає відповідне повідомлення.

8.14.4.6 Приклад безкоштовного протоколу RS232

Приклад програмування 1

◆ **Включення функції початку і кінця повідомлень**

1. Безкоштовний протокол застосовується для надсилання стандартного повідомлення Modbus у режимі ASCII із початковою адресою %MB4000, де зберігаються надіслані дані, та %MB5000, де зберігаються отримані дані, і отримує дані відповіді від підлеглого пристрою.
   Зміст повідомлення: 01 10 15 00 00 01 02 00 08.
   Спочатку обчислюється контрольна сума CF. Потім повідомлення перетворюється на код ASCII.
   Вміст повідомлення після перетворення: 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46.

1. Якщо параметри інструкції RS232_RS налаштовані на основі таблиці змінних 1, дані в %MB4000~%MB4019 є такими: 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46.

   Дані повідомлення на шині: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A .

■ **Таблиця змінних 1**

Ім'я змінної	Адреса	Тип даних	Початкове значення
RS		RS232_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#30
RS_SL		UINT	20
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	1
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Abt		BOOL	
RS_Act		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

◆ **Відключення функції початку і кінця повідомлень**

a. Безкоштовний протокол застосовується для надсилання стандартного повідомлення Modbus у режимі ASCII із початковою адресою %MB4000, де зберігаються надіслані дані, та %MB5000, де зберігаються отримані дані, і отримує дані відповіді від підлеглого пристрою.

Зміст повідомлення: 01 10 15 00 00 01 02 00 08.
Спочатку обчислюється контрольна сума CF. Потім повідомлення перетворюється на код ASCII.
Вміст повідомлення після перетворення: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

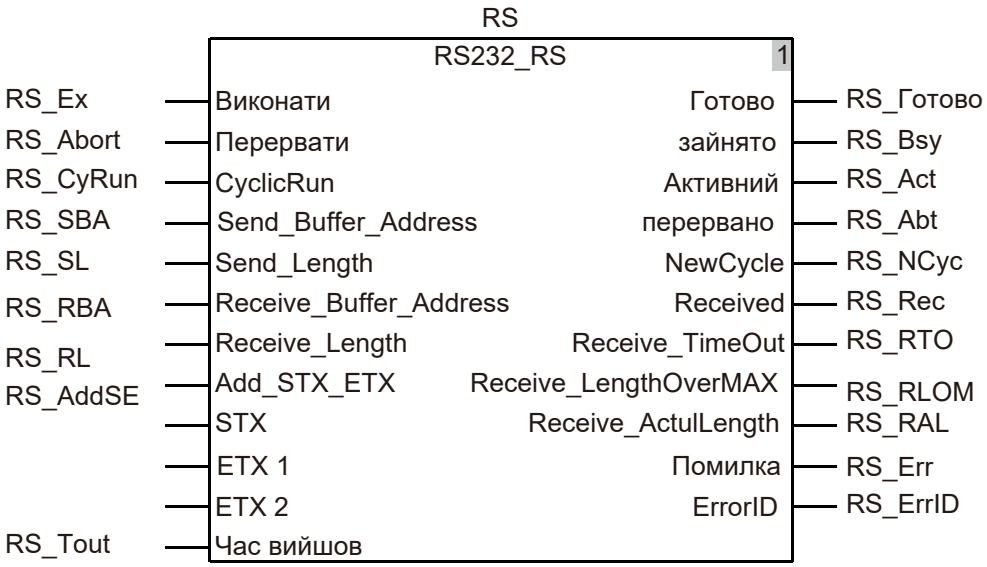
b. Якщо параметри інструкції RS232_RS налаштовано на основі таблиці змінних 2, дані в %MB4000~%MB4022 є 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A

Дані повідомлення на шині: 3A 30 31 31 30 31 35 30 30 30 30 30 31 30 32 30 30 30 38 43 46 0D 0A.

- **Таблиця змінних 2**

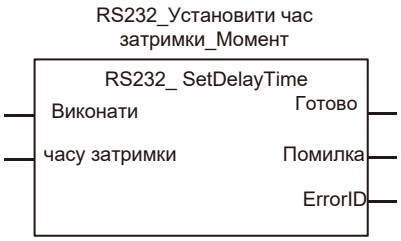
Ім'я змінної	Адреса	Тип даних	Початкове значення
RS		RS232_RS	
RS_Ex		BOOL	FALSE
RS_Abort		BOOL	FALSE
RS_CyRun		BOOL	TRUE
RS_SBA	%MB4000	USINT	16#3A
RS_SL		UINT	23
RS_RBA	%MB5000	USINT	
RS_RL		UINT	23
RS_AddSE		BOOL	0
RS_Tout		UINT	500
RS_Done		BOOL	
RS_Bsy		BOOL	
RS_Abt		BOOL	
RS_Act		BOOL	
RS_NCyc		BOOL	
RS_Rec		BOOL	
RS_RTO		BOOL	
RS_RLOM		BOOL	
RS_RAL		UINT	
RS_Err		BOOL	
RS_ErrID		WORD	

➢ **програма**


RS
RS232_RS
1
RS_Ex
RS_Abort
RS_CyRun
RS_SBA
RS_SL
RS_RBA
RS_RL
RS_AddSE
RS_Tout
Виконати
Перервати
CyclicRun
Send_Buffer_Address
Send_Length
Receive_Buffer_Address
Receive_Length
Add_STX_ETX
STX
ETX 1
ETX 2
Час вийшов
Готово
зайнято
Активний
перервано
NewCycle
Received
Receive_TimeOut
Receive_LengthOverMAX
Receive_ActulLength
Помилка
ErrorID
RS_Готово
RS_Bsy
RS_Act
RS_Abt
RS_NCyc
RS_Rec
RS_RTO
RS_RLOM
RS_RAL
RS_Err
RS_ErrID

8.14.4.7 RS232_SetDelayTime

FB/FC	Пояснення	Застосовна модель
FB	RS232_SetDelayTime встановлює час затримки відповіді на комунікаційному порту RS485 контролера.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
Час затримки	Час затримки відповіді на комунікаційному порту RS232 контролера. Він дійсний незалежно від того, чи контролер служить головним чи підлеглим. (Одиниця: мс)	UINT	0~65535 (0)	Коли *Execute* змінюється на TRUE

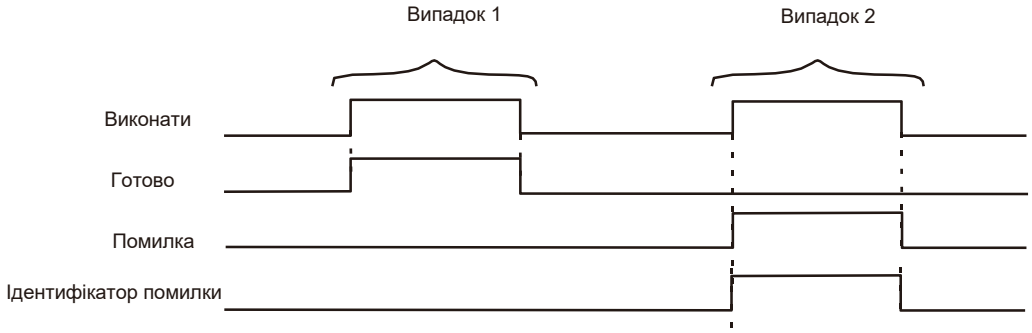
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли є помилка.	BOOL	ІСТИНА / НЕПРАВДА
Ідентифікатор помилки	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 3 : коли *Execute* змінюється з FALSE на TRUE, *DONE* змінюється з FALSE на TRUE.

Випадок 4 : коли виникає помилка, коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок. *Помилка* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

- **функція**

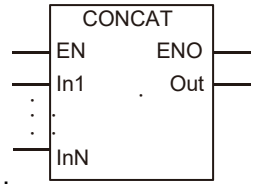
Інструкція RS232_SetDelayTime використовується для встановлення часу затримки відповіді комунікаційного порту RS232 контролера. Час налаштування дійсний, коли комунікаційний порт RS232 контролера працює як головний або підлеглий.

Коли комунікаційний порт RS232 працює як головний, він не надсилатиме наступне повідомлення, доки не мине встановлений час затримки після отримання даних відповіді від підлеглого. Коли комунікаційний порт RS232 працює як ведений, він не відповідатиме головному, доки не мине встановлений час затримки після отримання даних від головного.

8.15 Інструкції з обробки рядка

8.15.1 CONCAT

FB/FC	Пояснення	Застосовна модель
FC	CONCAT об’єднує дві або більше рядкових змінних або констант, щоб сформувати новий рядок.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1 до InN	Рядки для з'єднання	Вхід	Приєднаний параметр можна додати або видалити під час написання програми. Максимальна кількість об’єднаних параметрів – 8, а мінімальна – 2. N=2~8.	Залежить від типу даних змінної, з якою пов’язаний вхідний параметр.
Out	Результат приєднання	Вихід	Рядок отримано в результаті об’єднання	Залежить від типу даних змінної, до якої підключено вихідний параметр.

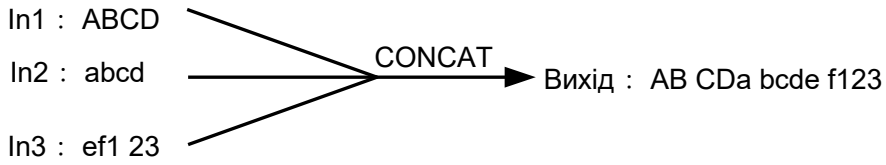
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1 до InN																				●
Out																				●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція CONCAT об’єднує два або більше рядків, щоб сформувати новий рядок, і новий рядок виводиться в *Out* Параметри від In1 до InN об’єднуються в такому порядку, як показано на наступному малюнку.



● **Застереження щодо правильного використання**

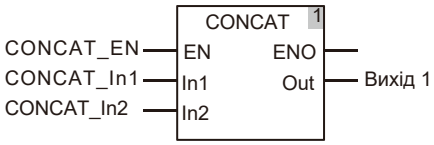
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Типи даних CONCAT_In1, CONCAT_In2 і Out1 є рядками, а значення CONCAT_In1 і CONCAT_In2 є "Asasz" і "B1255" відповідно. Коли CONCAT_EN має значення TRUE, значення Out1 дорівнює AsaszB1255.

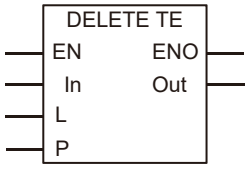
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
CONCAT_EN	BOOL	TRUE
CONCAT_In1	STRING	"Asasz"
CONCAT_In2	STRING	"B1255"
Out1	STRING	"AsaszB1255"



8.15.2 DELETE

FB/FC	Пояснення	Застосовна модель
FC	DELETE видаляє рядок заданої довжини з указаної позиції рядкової змінної або константи.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Рядок для видалення	Вхід	Рядок для видалення	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
L	Кількість символів для видалення	Вхід	Кількість символів для видалення	0~ максимальна довжина рядка
P	Початкова позиція видалення	Вхід	Початкова позиція видалення	1~ максимальна довжина рядка
Out	Результат видалення	Вихід	Рядок після видалення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

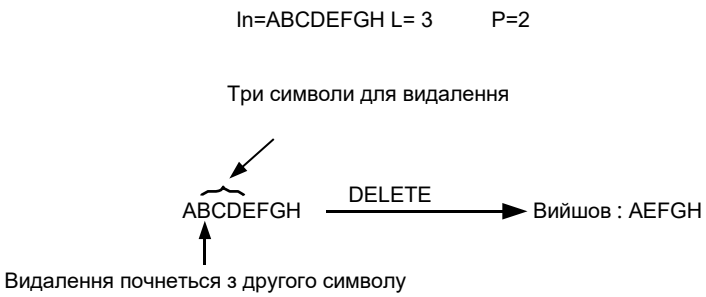
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
в																				●
L							●													
P							●													
Out																				●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція DELETE видаляє L символів, починаючи з позиції, визначеної *P* рядка *In* , і символи після видалення будуть виведені в *Out* . Спосіб видалення показано нижче.



● **Застереження щодо правильного використання**

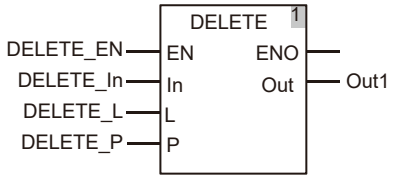
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ DELETE_In — це «AaBbCcDd», DELETE_L= 2 і DELETE_P = 3. Коли DELETE_EN має значення TRUE, Out1 — «AaCcDd».

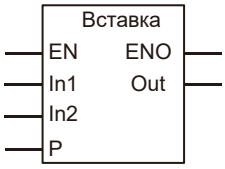
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
DELETE_EN	BOOL	TRUE
DELETE_In	STRING	"AaBbCcDd"
DELETE_L	UINT	2
DELETE_P	UINT	3
Out1	STRING	"AaCcDd"



8.15.3 INSERT

FB/FC	Пояснення	Застосовна модель
FC	INSERT вставляє рядок у вказану позицію в рядковій змінній або константі.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Оригінальний рядок	Вхід	Оригінальний рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Рядок для вставки	Вхід	Рядок для вставки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
P	Початкове положення введення	Вхід	Початкове положення введення	0~ максимальна довжина рядка
Out	Результат вставки	Вихід	Дивиться після вставки	Залежить від типу даних змінної, до якої підключено вихідний параметр.

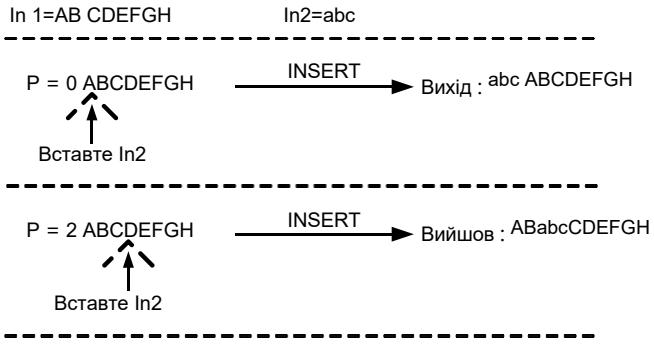
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
P							●													
Out																				●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● Пояснення функції

■ Інструкція INSERT вставляє рядок *In2* у рядок *In1* , а новий рядок виводиться до *Out.* Позиція вставки знаходиться між позицією, визначеною *P* , і позицією, визначеною *P+1* символів у *In1* . Якщо P =0, рядок *In2* вставляється на початку рядка *In1* . Спосіб вставки показано нижче.



● Застереження щодо правильного використання

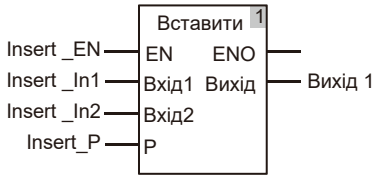
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Insert_In1 — «AaBbCcDd», Insert_In2 — «Ee», а Insert_P=2. Коли Insert_EN дорівнює TRUE, Out1 – це «AaEeBbCcDd».

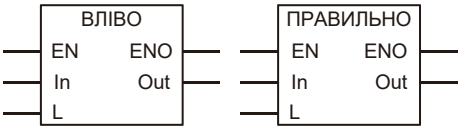
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
Insert_EN	BOOL	FALSE
Вставити_в1	STRING	"AaBbCcDd"
Вставити_в2	STRING	"Ee"
Вставити_P	UINT	2
Out1	STRING	"AaEeBbCcDd"



8.15.4 LEFT / RIGHT

FB/FC	Пояснення	Застосовна модель
FC	LEFT/RIGHT витягує рядок заданої довжини з рядкової змінної або константи.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Оригінальний рядок	Вхід	Оригінальний рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
L	Кількість символів, які потрібно отримати	Вхід	Кількість символів, які потрібно отримати	0~максимальна кількість символів
Out	Результат вилучення	Вихід	Результат вилучення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

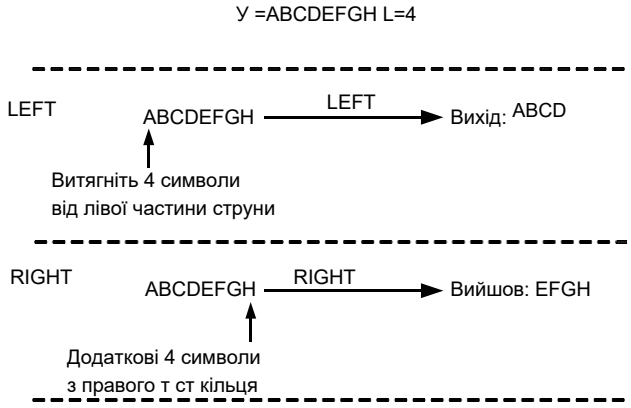
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In																				●
L							●													
Out																				●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція LEFT/RIGHT витягує рядок заданої довжини з рядка *In* , і витягнутий рядок виводиться до *Out* . Інструкція LEFT витягує символи зліва від рядка *In,* а інструкція RIGHT витягує символи з правого боку рядка. Спосіб вилучення символів показано нижче.



● **Застереження щодо правильного використання**

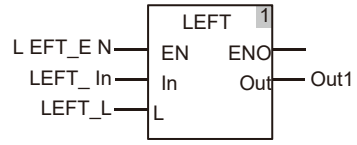
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Коли рядок LEFT_In дорівнює «AaBbCcDd», LEFT_L=2 і LEFT_EN дорівнює TRUE, Out1 дорівнює «Aa», як показано у наведеній нижче таблиці 1. Коли рядок RIGHT_In має значення «AaBbCcDd», RIGHT_L=2 і RIGHT_EN має значення TRUE, Out1 дорівнює «Dd», як показано в наступній таблиці 2.

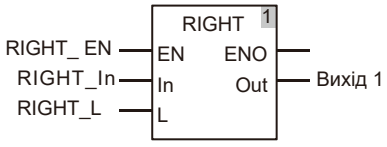
➢ **Таблиця змінних і програма 1**

Ім'я змінної	Тип даних	Поточне значення
LEFT_EN	BOOL	TRUE
LEFT_In	STRING	"AaBbCcDd"
LEFT_L	UINT	2
Out1	STRING	"Aa"



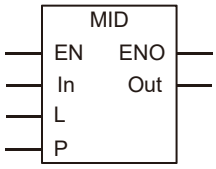
➢ **Таблиця змінних і програма 2**

Ім'я змінної	Тип даних	Поточне значення
RIGHT_EN	BOOL	TRUE
RIGHT_In	STRING	"AaBbCcDd"
RIGHT_L	UINT	2
Out1	STRING	"Дд"



8.15.5 MID

FB/FC	Пояснення	Застосовна модель
FC	MID витягує рядок заданої довжини з указаної позиції символу рядкової змінної або константи.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
в	Оригінальний рядок	Вхід	Оригінальний рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
L	Довжина символів для вилучення	Вхід	Кількість символів для вилучення	0~ максимальна кількість символів
P	Початкова позиція екстракції	Вхід	Початкова позиція екстракції	1~ максимальна кількість символів
Out	Результат вилучення	Вихід	Результат вилучення	Залежить від типу даних змінної, до якої підключено вихідний параметр.

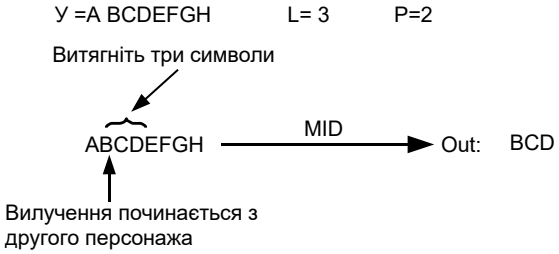
	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
в																				●
L							●													
P							●													
Out																				●

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● Пояснення функції

■ Інструкція MID витягує *L* символів, починаючи з символу числа P рядка *In* . Витягнутий рядок виводиться на *вихід* . Спосіб вилучення показаний нижче.



● Застереження щодо правильного використання

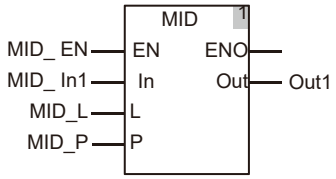
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Рядок MID_In — «AaBbCcDd», MID_L=2 і MID_LP=3. Коли MID_EN дорівнює TRUE, Out1 дорівнює «Bb».

➢ **Таблиця змінних і програма**

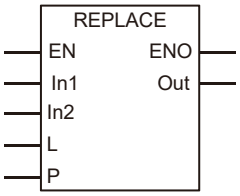
Ім'я змінної	Тип даних	Поточне значення
MID_EN	BOOL	TRUE
MID_In	STRING	"AaBbCcDd"
MID_L	UINT	2
MID_P	UINT	3
Out1	STRING	"Bb"



8

8.15.6 REPLACE

FB/FC	Пояснення	Застосовна модель
FC	Інструкція REPLACE замінює рядок заданої довжини, починаючи з указаної позиції, іншим рядком.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Оригінальний рядок	Вхід	Оригінальний рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Вставте рядок	Вхід	Рядок для вставки	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
L	Кількість символів	Вхід	Кількість символів для видалення	0~ максимальна кількість символів
P	Початкова позиція заміни	Вхід	Початкова позиція заміни	1~ максимальна кількість символів
Out	Результат заміни	Вихід	Результат заміни	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
L							●													
P							●													
Out																				●

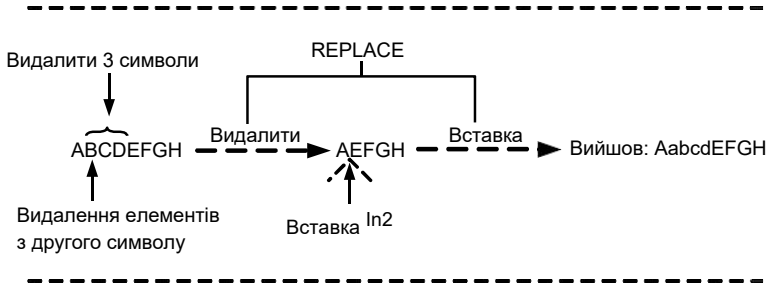
Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція REPLACE замінює L символів, починаючи з символу числа P рядка *In1* , вставляючи інший рядок *In2* . А результат заміни виводиться в *Out* . Процес заміни показано нижче.

In1=ABCDEFGH In2=abcd L=3 P=2



● **Застереження щодо правильного використання**

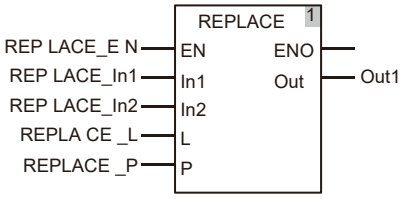
■ Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Рядок REPLACE_In1 — «AaBbCcDd», рядок REPLACE_In2 — «DELTA», REPLACE_L=2 і REPLACE_LP=3. Коли REPLACE_EN дорівнює TRUE, Out1 є «AaDELTACcDd».

➢ **Таблиця змінних і програма**

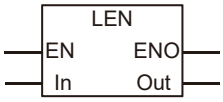
Ім'я змінної	Тип даних	Поточне значення
REPLACE_EN	BOOL	TRUE
REPLACE_In1	STRING	"AaBbCcDd"
REPLACE_In2	STRING	"ДЕЛЬТА"
REPLACE_L	UINT	2
REPLACE_P	UINT	3
Out1	STRING	"AaDELTACcDd"



8

8.15.7 LEN

FB/FC	Пояснення	Застосовна модель
FC	LEN обчислює кількість символів у рядку.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Рядок	Вхід	Рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Кількість символів	Вихід	Кількість символів	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
в																				●
Out							●													

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція LEN знаходить кількість символів у рядку, а результат виводиться в *Out* . Наприклад, якщо рядок ABCDEFGH, значення *Out* дорівнює 8.

● **Застереження щодо правильного використання**

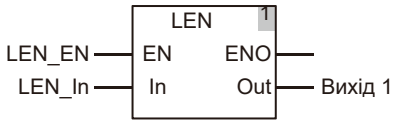
Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Рядок LEN_In — «AaBbCcDd». Оскільки LEN_EN має значення TRUE, значення Out1 дорівнює 8.

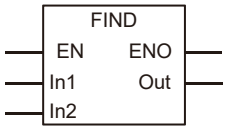
➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
LEN_EN	BOOL	TRUE
LEN_In	STRING	AaBbCcDd
Out1	UINT	8



8.15.8 FIND

FB/FC	Пояснення	Застосовна модель
FC	FIND шукає позицію зазначеного рядка в іншому рядку.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In1	Рядок	Вхід	Рядок	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
In2	Ключові символи для пошуку	Вхід	Ключові символи для пошуку	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Кількість символів	Вихід	Кількість символів	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
In1																				●
In2																				●
Out							●													

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

- Інструкція Find приймає символи в *In2* як ключові символи та шукає позицію ключових символів у рядку *In1* . Наприклад, оскільки *In1* дорівнює ABCDEFGH, а *In2* – DE, значення *Out* дорівнює 4.
- Пошук починається з першого символу в рядку *In1.*
- *In1* існує декілька рядків *In2* , значення *Out* є позицією першого *In2* від початку *In1* .

● **Застереження щодо правильного використання**

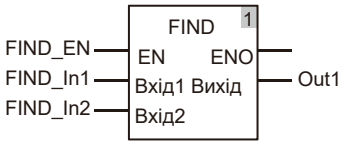
Вхідні змінні не можна опускати. Якщо вхідні змінні опущено, під час компіляції програмного забезпечення виникне помилка. Але вихідну змінну можна опускати.

Приклад програмування

■ Рядок FIND_In1 — «AaBbCcDd», а рядок FIND_In2 — «Cc». Оскільки FIND_EN має значення TRUE, значення Out1 дорівнює 5.

➢ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
FIND_EN	BOOL	TRUE
FIND_In1	STRING	"AaBbCcDd"
FIND_In2	STRING	"Cc"
Out1	UINT	5

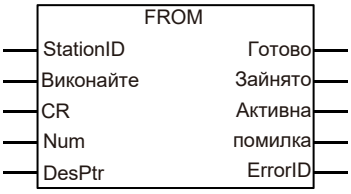


8.16 Інструкції щодо негайного оновлення

8.16.1 FROM

FB/FC	Пояснення	Застосовна модель
FB	Інструкція FROM читає значення в регістрах CR правого модуля розширення.	AS516E-B AS532EST-B AS564EST-B

FROM_у випадку



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
StationID	Розташування модуля розширення, підключеного до правої сторони контролера руху	USINT	Діапазон положення правого модуля розширення: 1~32 Номер позиції першого модуля з правого боку контролера руху - 1. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (Потрібно встановити значення змінної)	-
CR	Номер першого CR (контрольованого регістру), який потрібно зчитати	UINT	0~макс. номер CR (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
кількість	Кількість регістрів CR, які потрібно прочитати	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
DesPtr	Значення CR, зчитані інструкцією	INT або DINT	Діапазон типу даних прочитаного значення CR (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

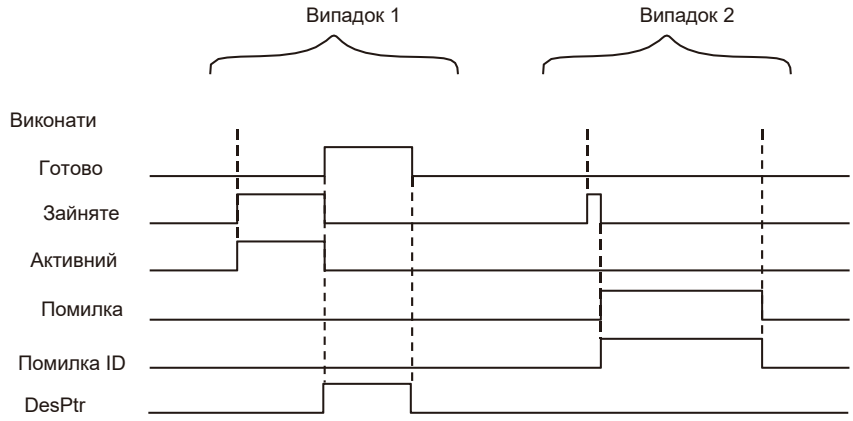
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Активний	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після завершення зчитування значень параметрів.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Зайняте	◆ Коли *Execute* змінюється на TRUE	◆ Коли *Done* змінюється з FALSE на TRUE ◆ Коли *помилка* змінюється на TRUE.
Активний	◆ Коли починається виконання інструкції	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *«Виконати»* змінюється з FALSE на TRUE, *«Зайнятий* » і *«Активний»* змінюється на TRUE, а через один період « *Готово»* змінюється на TRUE. Тим часом *Busy* та *Active* змінюються на FALSE і *DesPtr* показують відповідні дані в регістрах CR модуля розширення. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE, а значення *DesPtr* скидається на *0* .

Випадок 2 : коли виникає помилка, коли *Execute* має значення TRUE, *Error* змінюється з FALSE на TRUE, а *ErrorID* показує відповідні коди помилок. *Помилка* змінюється з TRUE на FALSE, а значення ErrorID очищається до 0 після того, як *Execute* змінюється з TRUE на FALSE.

● **Пояснення функції**

Інструкцію FROM можна застосувати для читання значень у регістрах модулів розширення правої сторони. Положення правого модуля визначається *StationID* . Діапазон ідентифікаторів станції правого модуля розширення становить 1~32. 1 позначає перший аналоговий модуль розширення праворуч, а 32 означає тридцять

другий модуль розширення з правого боку. Якщо діапазон стандартного ідентифікатора перевищує вказаний діапазон правого модуля, виникне помилка при виконанні інструкції.
Якщо інструкцією потрібно прочитати більше одного регістра CR, параметр *DesPtr* потрібно визначити як N-й елемент масиву. Дані в першому регістрі CR будуть зчитані в N-й елемент масиву, дані в другому регістрі CR будуть зчитані в N+1-й елемент і так далі. Таким чином, усі дані в кількох регістрах CR будуть зчитані в масив. Зверніться до прикладу програмування 2 для отримання детальної інформації.

⚠ Запобіжні заходи

До правої сторони контролера руху можна підключити до 32 модулів розширення. Наприклад, якщо AS04AD-A, AS16AP11T і AS04DA-A підключені до правої сторони контролера руху один за одним, значення *StationID* для AS04AD-A дорівнює 1, AS16AP11T дорівнює 2 і AS04DA-A дорівнює 3.

Приклад програмування 1

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
FRM	FROM	
FRM_ID	USINT	1
FRM _Ex	BOOL	FALSE
FRM _CR	UINT	0
FRM _Num	USINT	1
FRM _DP	INT	
FRM _Done	BOOL	
FRM _Bsy	BOOL	
FRM _Act	BOOL	
FRM _Err	BOOL	
FRM _ErrID	WORD	



AS04AD-A підключається до правої сторони контролера руху. Коли FRM _Ex змінюється з FALSE на TRUE, а FRM _Bsy і FRM _Act змінюються на TRUE одночасно, інструкція FROM починає виконуватися. Коли FRM _Done змінюється на TRUE, виконання інструкції завершено. FRM _DP показує, що значення в CR0, зчитане інструкцією, дорівнює 136, а отже, версія AS-04AD дорівнює 1.36.

Приклад програмування 2

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
FRM1	FROM	
FRM1_ID	USINT	1
FRM1_Ex1	BOOL	FALSE
FRM1_CR1	UINT	2
FRM1_Num1	USINT	4
FRM1_DP	Масив [1..4] INT	
FRM1_Done	BOOL	
FRM1_Bsy	BOOL	
FRM1_Act	BOOL	

Ім'я змінної	Тип даних	Поточне значення
FRM1_Err	BOOL	
FRM1_ErrID	WORD	



AS04AD-A підключається до правої сторони контролера руху. Коли FRM1_Ex змінюється з FALSE на TRUE, а FRM1_Bsy і FRM1_Act змінюються на TRUE одночасно, інструкція FROM починає виконуватися. Коли FRM1_Done змінюється на TRUE, виконання інструкції завершено. Значення, зчитані з CR2, CR3, CR4 і CR5, зберігаються в чотирьох елементах FRM1_DP[1], FRM1_DP[2], FRM1_DP[3] і FRM1_DP[4] масиву FRM1_DP.

8.16.2 TO

FB/FC	Пояснення	Застосовна модель
FB	Інструкція TO записує дані до вказаних регістрів CR правого модуля.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
StationID	Розташування модуля розширення, підключеного до правої сторони контролера руху	USINT	Діапазон положення правого модуля розширення: 1~32 Номер позиції першого модуля з правого боку контролера руху - 1. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Виконати	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	
CR	Номер першого регістра CR, у який мають бути записані дані	UINT	0~ макс. номер CR (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
кількість	Кількість регістрів CR, які потрібно прочитати	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
DesPtr	Значення CR, записане інструкцією	INT або DINT	Діапазон типу даних записаного значення CR (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

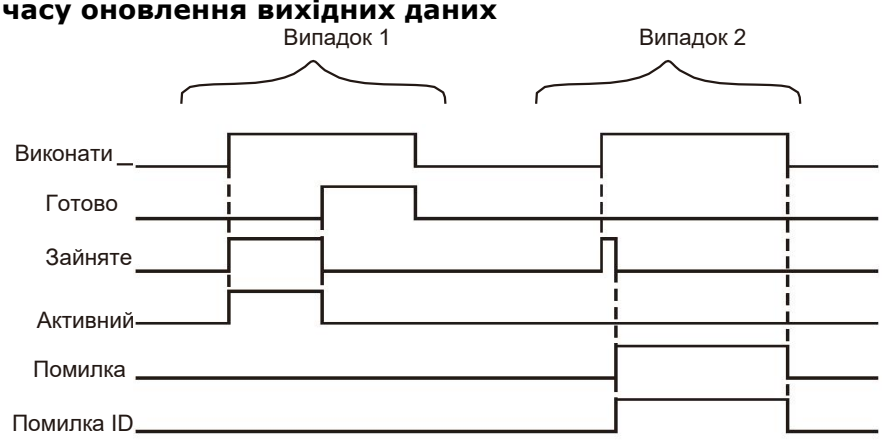
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли запис значення параметра завершено.	BOOL	ІСТИНА / НЕПРАВДА
Зайняте	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Активний	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	-

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після завершення запису значень параметрів.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції
Зайняте	◆ Коли *Execute* змінюється на TRUE	◆ Коли *Done* змінюється з FALSE на TRUE ◆ Коли *помилка* змінюється на TRUE.
Активний	◆ Коли починається виконання інструкції	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *«Виконати»* змінюється з FALSE на TRUE, *«Зайнятий»* і *«Активний»* змінюється на TRUE. Через крапку *«Готово»* змінюється на «ІСТИНА». Тим часом *«Зайнятий»* і *«Активний»* змінюється з TRUE на FALSE. Після того, як *«Виконати»* змінюється з «ІСТИНА» на «ХИБЬ», *«Готово»* змінюється з «ІСТИНА» на «ХИБНЯ».

Випадок 2 : коли виникає помилка, коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється з FALSE на TRUE, а *ErrorID* показує відповідні коди помилок. *Помилка* змінюється з TRUE на FALSE, а значення ErrorID очищається до 0 після того, як *Execute* змінюється з TRUE на FALSE.

● **Пояснення функції**

Інструкція TO використовується для запису даних у вказані регістри CR правого модуля.

Позиції модулів розширення правої сторони визначаються *StationID* . *Діапазон StationID* правого модуля становить 1~32. 1 являє собою перший модуль розширення з правого боку. 32 — це тридцять другий модуль розширення з правого боку. Якщо значення *StationID* виходить за межі зазначеного діапазону для модулів правої сторони, під час виконання інструкції виникне помилка.

Якщо інструкція використовується для запису значень у декілька регістрів CR, *DesPtr* потрібно визначити як N -й елемент масиву. Потім кілька значень будуть записані в кілька регістрів CR шляхом запису значення N -го елемента в перший CR, значення N+1 -го елемента в другий CR і так далі після виконання інструкції.

Зверніться до наступних прикладів програм, щоб дізнатися більше про використання.

⚠ **Запобіжні заходи**

До правої сторони контролера руху можна підключити до 32 модулів розширення. Наприклад, якщо AS04AD-A, AS16AP11T і AS04DA-A підключені до правої сторони контролера руху один за одним, значення *StationID* для AS04AD-A дорівнює 1, AS16AP11T дорівнює 2 і AS04DA-A дорівнює 3.

Приклад програмування 1

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Поточне значення
TO1	TO	
TO1_ID	USINT	1
TO1_Ex	BOOL	FALSE
TO1_CR	UINT	2
TO1_Num	USINT	1
TO1_DP	INT	10
TO1_Done	BOOL	
TO1_Bsy	BOOL	
TO1_Act	BOOL	
TO1_Err	BOOL	
TO1_ErrID	WORD	



AS04AD-A підключається до правої сторони контролера руху. Коли TO1_Ex змінюється з FALSE на TRUE, TO1_Bsy і TO1_Act одночасно змінюються на TRUE і починається виконання інструкції TO. Коли TO1_Done змінюється на TRUE, виконання інструкції завершено, і значення, яке записується в CR2 в AS04AD-A, дорівнює 10.

Приклад програмування 2

■ **Таблиця змінних і програма**

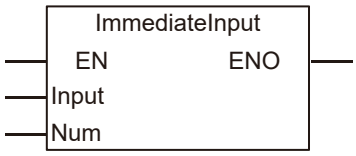
Ім'я змінної	Тип даних	Поточне значення
TO2	TO	
TO2_ID	USINT	1
TO2_Ex	BOOL	FALSE
TO2_CR	UINT	2
TO2_Num	USINT	4
TO2_DP	Array[1..4] INT	
TO2_Done	BOOL	
TO2_Bsy	BOOL	
TO2_Act	BOOL	
TO2_Err	BOOL	
TO2_ErrID	WORD	



AS04AD-A підключається до правої сторони контролера руху. Коли TO2_Ex змінюється з FALSE на TRUE, TO2_Bsy і TO2_Act одночасно змінюються на TRUE і починається виконання інструкції TO. Коли TO2_Done змінюється на TRUE, виконання інструкції завершено, а значення, записані в CR2, CR3, CR4 і CR5 в AS04AD-A, є значеннями, записаними в чотирьох елементах TO2_DP[1], TO2_DP[2], TO2_DP[3] і TO2_DP[4] масиву TO2_DP відповідно.

8.16.3 ImmediateInput

FB/FC	Пояснення	Застосовна модель
FC	ImediateInput використовується для негайного оновлення точок введення.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Input	Початкова точка введення	Вхід	Початкова точка введення	0~15
Num	Номер	Вхід	Кількість точок введення для негайного оновлення	1~16

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Input											●									
Num						●														

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

■ Інструкція ImmediateInput використовується для оновлення зовнішнього стану точки введення до %IX0.0~%IX0.7 і %IX1.0~%IX1.7. Якщо інструкція ImmediateInput не існує, контролер оновлює стан зовнішньої точки введення до %IX0.0~%IX0.7 і %IX1.0~%IX1.7 лише один раз під час кожного запуску сканування програми.

■ Значення *вхідного* параметра 0~7 і 8~15 відповідає %IX0.0~%IX0.7 і %IX1.0~%IX1.7. *Num* представляє кількість послідовних пристроїв, починаючи з одного, зазначеного в *Input* . Наприклад, якщо *вхідне* значення дорівнює 0, а Num дорівнює 2, це вказує на те, що стан зовнішньої точки введення оновлено до %IX0.0 і %IX0.1.

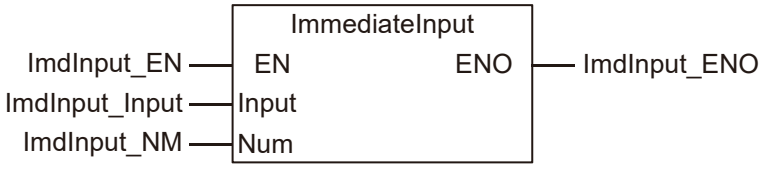
● **Застереження щодо правильного використання**

■ Ця інструкція використовується лише для негайного оновлення локальних точок введення замість точок введення розширення.

Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
ImdInput_EN	BOOL	FALSE
ImdInput_Input	INT	2
ImdInput_NM	USINT	2
ImdInput_ENO	BOOL	

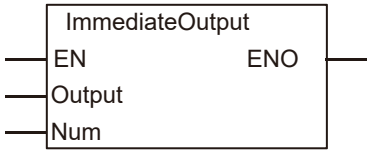


■ Пояснення програми

Коли вхідна змінна ImdInput_EN має значення TRUE, стан зовнішніх апаратних точок введення буде оновлено до %IX0.2 і %IX0.3.

8.16.4 ImmediateOutput

FB/FC	Пояснення	Застосовна модель
FC	ImediateOutput використовується для негайного оновлення вихідних точок.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Output	Початкова точка виведення	Вхід	Початкова точка Вихід	0~7
Num	Номер	Вхід	Кількість вихідних точок для негайного оновлення	1~8

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Output											●									
Num						●														

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних. .

● **Пояснення функції**

- Інструкція ImmediateOutput використовується для оновлення поточного стану внутрішньої вихідної точки %QX0.0~%QX0.7 до зовнішньої апаратної вихідної точки. Якщо інструкція ImmediateOutput не існує, контролер оновлює стан внутрішньої точки Вихід на зовнішню апаратну точку виводу. Статус %QX0.0~%QX0.7 визначається іншими інструкціями. Інструкція ImmediateOutput використовується лише для оновлення статусу %QX0.0~%QX0.7 для зовнішніх апаратних точок виводу. Інструкція ImmediateOutput не контролює TRUE або FALSE %QX0.0~%QX0.7.
- вихідного *параметра* 0~7 інструкції ImmediateOutput відповідає %QX0.0~%QX0.7. *Num* представляє кількість послідовних пристроїв, починаючи з одного, зазначеного в *Output* . Наприклад, коли *вихідне* значення дорівнює 0, а *Num* дорівнює 2, це вказує на те, що стан %QX0.0 і %QX0.1 оновлено до точки зовнішнього апаратного Вихід.

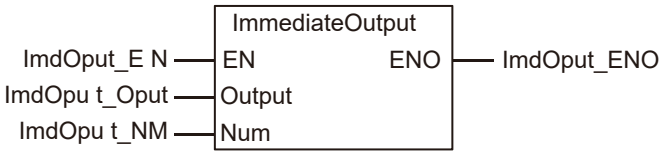
● **Застереження щодо правильного використання**

Ця інструкція використовується лише для негайного оновлення локальних точок виводу замість точок Вихід розширення.

Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Тип даних	Поточне значення
ImdOput_EN	BOOL	FALSE
ImdOput_Oput	INT	2
ImdOput_NM	USINT	2
ImdOput_ENO	BOOL	



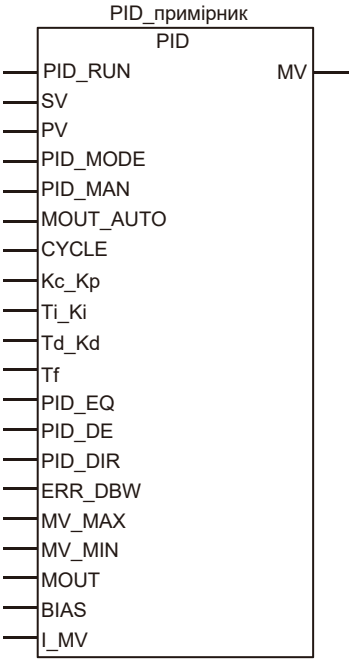
■ Пояснення програми

Коли вхідна змінна ImdOput_EN має значення TRUE, стан %QX0.2 і %QX0.3 буде оновлено до точки зовнішнього апаратного Вихід.

8.17 Інструкції щодо PID

8.17.1 PID

FB/FC	Пояснення	Застосовна модель
FB	Інструкція PID застосовна для операції PID.	AS516E-B AS532EST-B AS564EST-B



- **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
PID_RUN	Увімкнути роботу PID	Вхід	Увімкнути роботу PID	Правда чи брехня
SV	Цільове значення	Вхід	Цільове значення	−3,402823e+38 ~ −1.175495e-38 · 0 , +1,175495e-38 ~ +3,402823e+38
PV	Поточне значення	Вхід	Поточне значення	
PID_MODE	Режим ПІД-регулювання	Вхід	0: Автоматичне керування, вихідне значення (MV) бере участь у автоматичній роботі. 1: функція автоматичного налаштування параметрів; після завершення налаштування параметрів автоматично входить режим автоматичного керування (PID_MODE встановлено на 0) і заповнюються відповідні параметри Kc_Kp, Ti_Ki, Td_Kd і Tf.	0 , 1
PID_MAN	Режим PID A/M	Вхід	ІСТИНА: Ручний режим FALSE: Автоматичний режим	Правда чи брехня
MOUT_AUTO	Зарезервований	Вхід	-	-

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
CYCLE	Час вибірки （ T_S ）	Вхід	Час вибірки （ T_S ）	1~40 000 (Одиниця: мс)
Kc_Kp	Пропорційний коефіцієнт	Вхід	Розрахований пропорційний коефіцієнт P менше 0, Kc_Kp буде 0.	0 , +1,175495e-38 ~ +3,402823e+38
Ti_Ki	Інтегральний коефіцієнт	Вхід	Якщо розрахований коефіцієнт I менше 0, Ti_Ki буде 0.	0 , +1,175495e-38 ~ +3,402823e+38 (Одиниця: Ti = сек; Ki = 1/c)
Td_Kd	Коефіцієнт похідної	Вхід	Якщо обчислений коефіцієнт D менше 0, Td_Kd буде 0.	0 , +1,175495e-38 ~ +3,402823e+38 (Одиниця вимірювання: секунди)
Tf	Похідна-постійна часу дії	Вхід	Якщо постійна часу дії похідної менше 0, Tf дорівнюватиме 0	
PID_EQ	Зарезервований	Вхід	-	-
PID_DE	Зарезервований	Вхід	-	-
PID_DIR	PID напрямок вперед/назад	Вхід	TRUE: Позитивний напрямок (E=SV-PV) НЕПРАВИЛЬНО: Негативний напрямок (E=PV-SV)	Правда чи брехня
ERR_DBW	Діапазон, у межах якого значення помилки вважається 0	Вхід	Діапазон, у межах якого значення помилки вважається 0	−3,402823e+38 ~ −1.175495e-38 , 0 , +1,175495e-38 ~ +3,402823e+38
MV_MAX	Верхня межа вихідного значення MV	Вхід	Верхня межа вихідного значення MV	
MV_MIN	Нижня межа вихідного значення MV	Вхід	Нижня межа вихідного значення MV	
MVT	Значення ручного Вихід	Вхід	Значення ручного Вихід	
BIAS	Вихідне значення прямого зв'язку	Вхід	Вихідне значення прямого зв'язку	
I_MV	Зарезервований	Вхід	системний параметр; Не використовуйте його.	
MV	Вихідне значення	Вихід	Значення MV знаходиться між MV_MAX і MV_MIN.	−3,402823e+38 ~ −1.175495e-38 , 0 , +1,175495e-38 ~ +3,402823e+38

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
PID_RUN	●																			
SV														●						
PV														●						
PID_MODE												●								
PID_MAN	●																			
MOUT_AUTO	●																			
CYCLE												●								
Kc_Kp														●						
Ti_Ki														●						
Td_Kd														●						
Tf														●						
PID_EQ	●																			
PID_DE	●																			
PID_DIR	●																			
ERR_DBW														●						
MV_MAX														●						
MV_MIN														●						
MOUT														●						
BIAS														●						
I_MV														●						
MV														●						

Примітка:
Інструкція використовується для реалізації операції PID. Операція PID виконується лише тоді, коли інструкція PID виконується ПЛК. PID означає пропорцію, інтеграл і похідну. ПІД-регулювання широко застосовується до механічного обладнання, пневматичного обладнання та електронного обладнання.

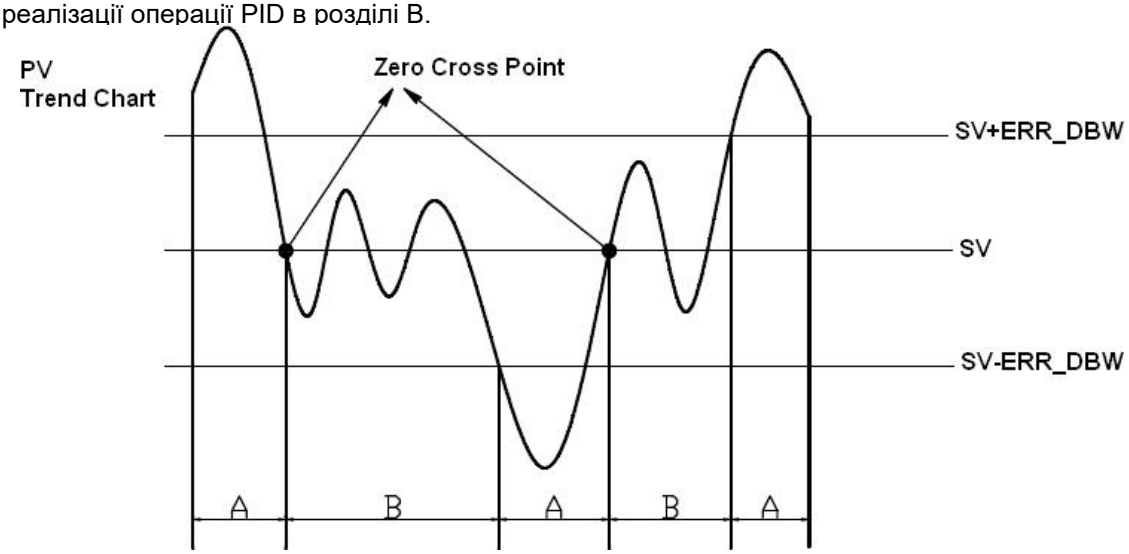
● **Пояснення функції**

- Немає обмежень щодо того, скільки разів можна використовувати інструкцію. Однак змінна, визначена I_MV, не може повторно використовуватися іншою програмою.
- Інструкцію PID можна використовувати лише в циклічному завданні.
- Поки PID сканується, відповідно до часу вибірки, визначеного *CYCLE,* виконується операція PID

реалізовано, і значення MV виводиться безпосередньо. ПЛК не розраховуватиме автоматично, чи досягне час сканування часу вибірки, визначеного *CYCLE* , для Вихід.

- Поточне значення (PV) ПІД-регулятора має бути постійним перед виконанням операції ПІД-регулятора. Якщо вхідні значення спеціальних модулів повинні бути захоплені для роботи ПІД, користувачі повинні звернути увагу на час аналого-цифрового перетворення модулів.

- Коли значення PV знаходиться в діапазоні **ERR_DBW** , на початку поточна помилка буде передана в операцію PID відповідно до нормальної обробки, а потім модуль ПЛК перевірить, чи поточна помилка відповідає умові перехресного статусу: PV (процес значення) проходить повз SV (цільове значення). Після того, як умова буде виконана, поточна помилка буде зарахована як 0 для операції PID. І після того, як значення PV виходить за межі діапазону **ERR_DBW** , поточна помилка знову буде введена в операцію PID.
  Якщо PID_DE дорівнює TRUE, це означає використання значення PV для обчислення контрольного значення похідної, і після виконання умови перехресного статусу ПЛК розглядатиме Δ **PV** як 0 для реалізації операції PID. (Δ **PV** = поточний **PV** – попередній **PV**). У прикладі, показаному нижче, поточна помилка буде введена в операцію PID відповідно до нормальної обробки в розділі A, а поточна помилка або Δ **PV** буде зарахована як 0 для реалізації операції PID в розділі B.



➢ **Алгоритм PID:**

Коли **PID_MODE** встановлено на 0, режим ПІД-регулювання є автоматичним

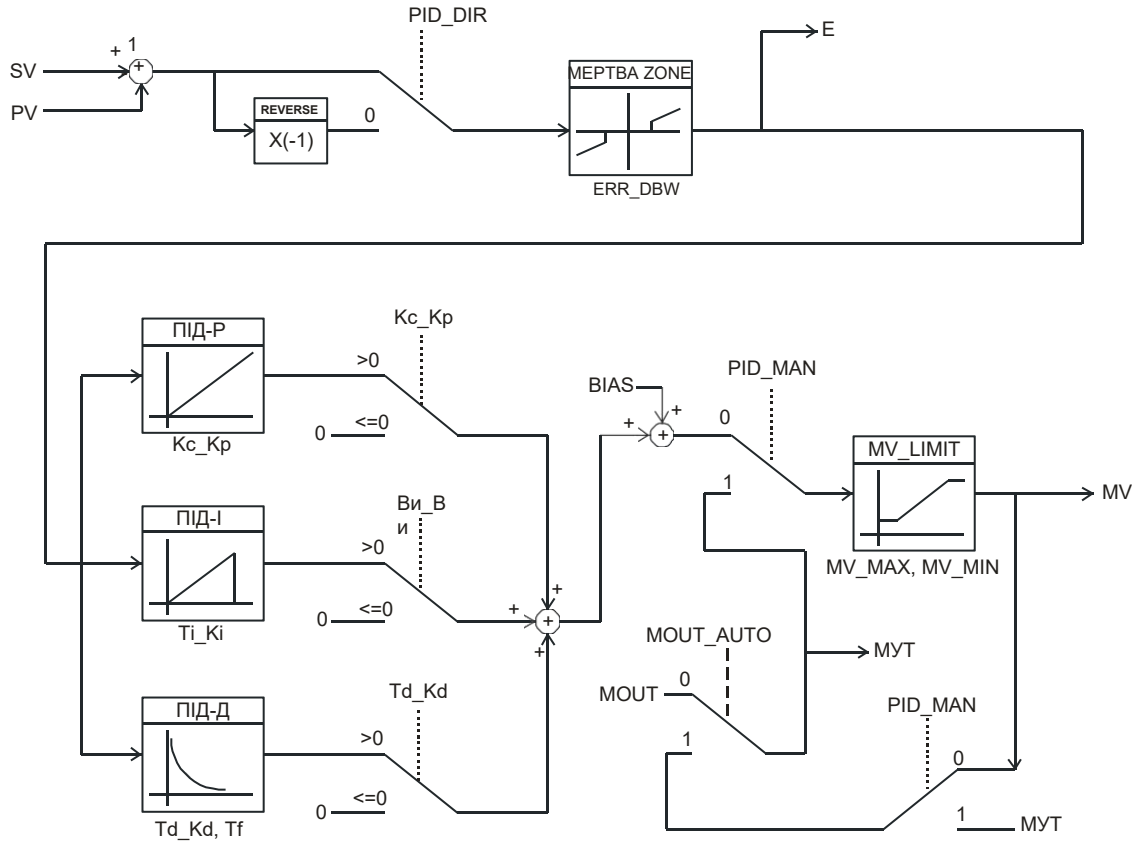
Незалежна формула та похідна E (**PID_EQ** =False & **PID_DE** =Fals e)

$$MV = K_P E + Ki \int_0^t \mathrm{Edt} + K_d * \frac{dE}{dt} + BIAS$$ *E = SV – PV* або *E = PV – SV*

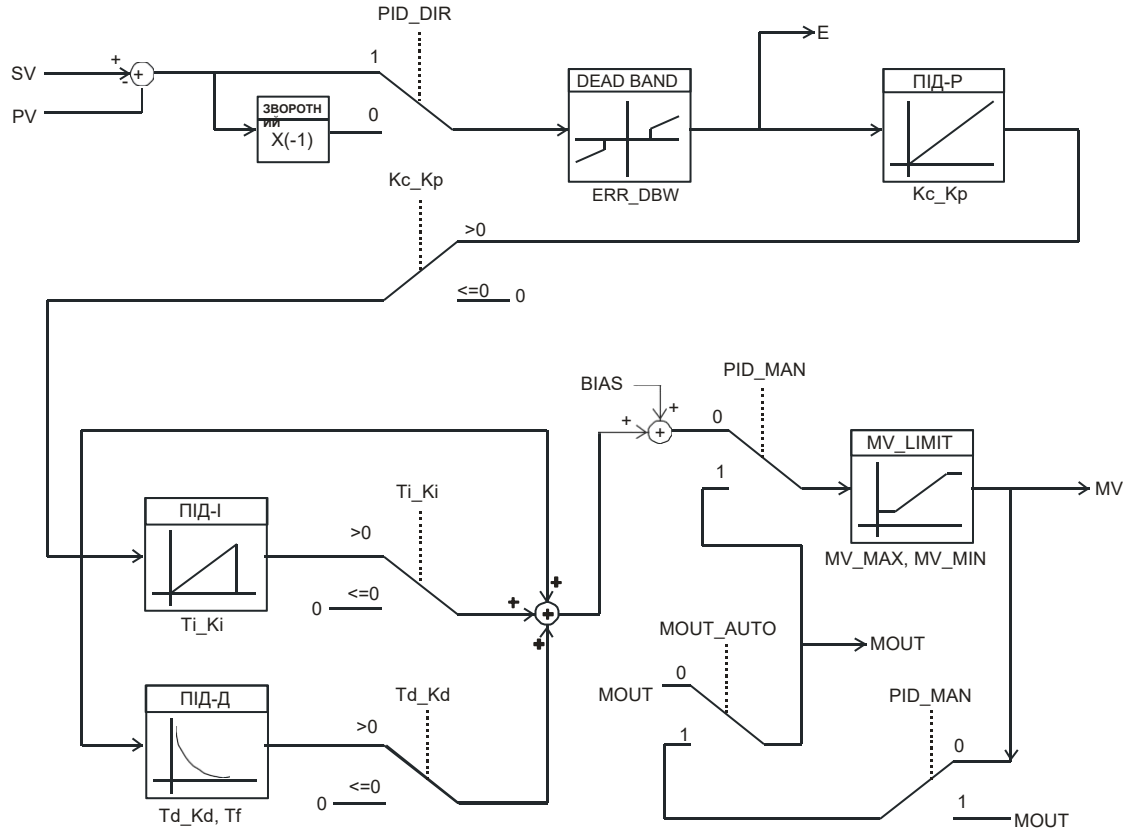
Коли **PID_MODE** встановлено на 1, режимом ПІД-регулювання є режим автоматичного налаштування. Після завершення налаштування параметра **PID_MODE** автоматично стає 0, а режим ПІД-регулювання стає автоматичним режимом керування

➢ **Блок-схема PID:**

Блок-схема PID (незалежна)



Блок-схема PID (залежно)



➢ **Застереження та рекомендації:**

Через те, що інструкцію PID можна використовувати в багатьох контрольованих середовищах, користувачі повинні правильно вибрати функцію керування. Наприклад, щоб запобігти виникненню неправильного керування, **PID_MODE** не можна використовувати в середовищі, керованому двигуном, оскільки він встановлений на 1, а значення MV перемикається між MAX і MIN.

Коли користувачі налаштовують параметри **Kc_Kp** , **Ti_Ki** і **Td_Kd** (PID_MODE встановлено на 0), вони повинні спочатку налаштувати **Kc_Kp** (відповідно до досвіду), а потім встановити **Ti_Ki** і **Td_Kd** на 0. Коли користувачі зможуть контролювати, вони можуть збільшити **Ti_Ki** і **Td_Kd** . Коли **Kc_Kp** дорівнює 1, це означає, що пропорційне посилення становить 100%. Тобто значення похибки збільшується в одиницю. Коли пропорційне посилення менше 100%, значення помилки зменшується. Коли пропорційне посилення перевищує 100%, значення помилки збільшується.

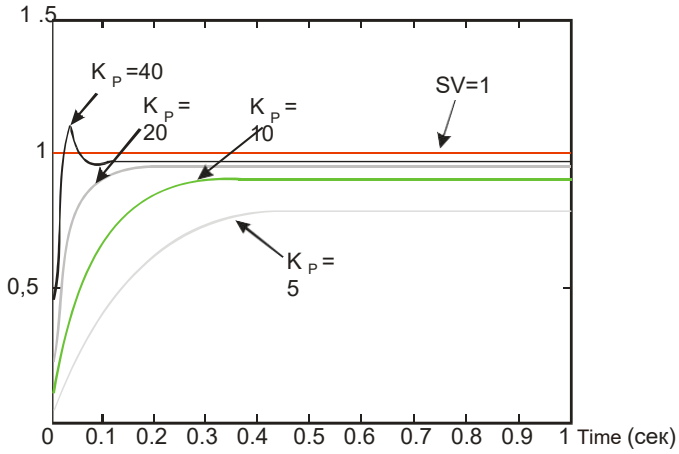
Щоб запобігти зникненню автоматично налаштованих параметрів після відключення електроенергії, ми пропонуємо користувачам зберігати параметри в зафіксованих змінних, коли вибрано **PID_MODE =1.** Параметри, які були налаштовані автоматично, не обов'язково підходять для кожного контрольованого середовища. Таким чином, користувачі можуть змінювати параметри, які були налаштовані автоматично. Однак пропонується, щоб користувачі змінювали лише **Ti_Ki** та **Td_Kd.**

Інструкцію слід використовувати з багатьма параметрами. Щоб запобігти неправильному контролю, не встановлюйте параметри випадково.

➢ **Етапи налаштування параметрів, що використовуються з інструкцією PID**

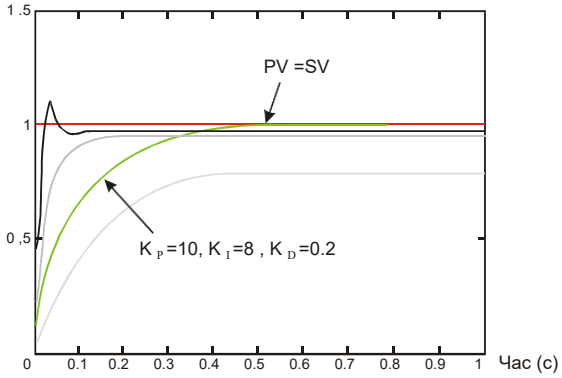
Припустимо, що функція передачі G (s) заводу є функцією першого порядку $G(s)=\frac{b}{s+a}$, *SV* є 1, час вибірки *CYCLE* становить 10 мілісекунд. Пропонується, щоб кроки налаштування параметрів були такими.

Крок 1: спочатку встановіть K_i і K_d на 0. Потім послідовно встановіть K_P на 5, 10, 20 і 40 і запишіть цільові значення та значення процесу. Результати показані на діаграмі нижче.



Крок 2: Коли K_P становить 40, спостерігається надмірна реакція. Таким чином, K_P не вибрано. Коли K_P дорівнює 20, крива реакції ***PV*** близька до ***SV***, і надмірної реакції немає. Однак через швидкий запуск перехідне вихідне значення (MV) дуже велике. K $_P$ ***також*** не обрано. Коли K_P дорівнює 10, крива реакції PV ***плавно*** наближається до ***SV .*** Тому вибрано K_P. Коли K_P дорівнює 5, реакція йде надто повільно. Таким чином, K_P не вибрано.

Крок 3: Після того як K_P буде встановлено на 10, збільште K_I. Наприклад, K_I послідовно встановлюється на 1, 2, 4 і 8. K ***I*** не має бути більшим за K_P. Потім збільште K_D. Наприклад, K_D встановлюється на 0,01, 0,05, 0,1 і 0,2 послідовно. K ***D*** не має перевищувати десяти відсотків *від* K_P. Нарешті, співвідношення між ***PV*** і ***SV*** представлено на наступній діаграмі.



Примітка:

Приклад лише для ознайомлення. Користувачі повинні правильно налаштувати параметри відповідно до практичного стану системи керування

Приклад програмування

- **Використання функції автоматичного налаштування для контролю температури**
- **Призначення: використання функції автоматичного налаштування для розрахунку найбільш відповідних параметрів для ПІД-контролю температури**
- **Пояснення:**

  Через те, що користувачі можуть бути не знайомі з характеристиками температурного середовища, яке контролюється вперше, вони можуть використовувати функцію автоматичного налаштування для початкового налаштування (**PID_MODE** встановлено на 1). Після завершення налаштування параметра **PID_MODE** встановлюється на 0. Контрольоване середовище в цьому прикладі – піч. Приклад програми наведено нижче

- **Таблиця змінних і програма**

1. Глобальна таблиця змінних

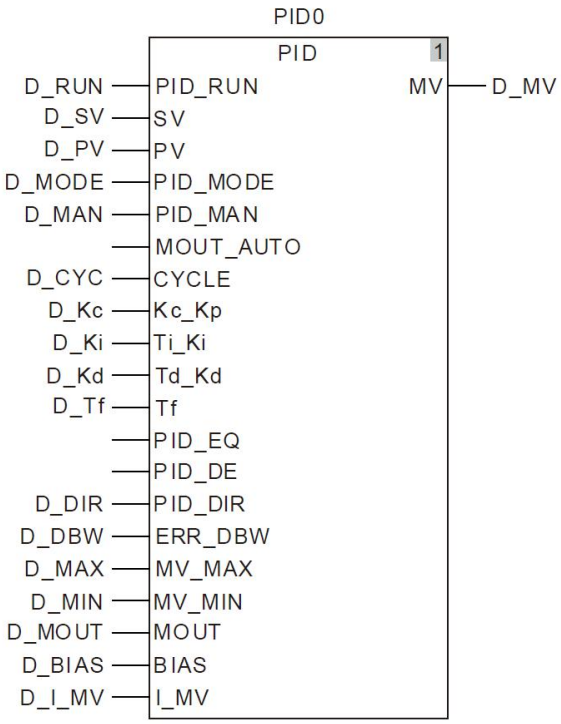
Ім'я змінної	Тип даних	Поточне значення
M1	BOOL	FALSE
D_RUN	BOOL	
G_In1	INT	

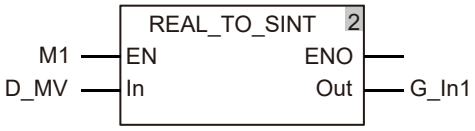
2. Таблиця локальних змінних

Ім'я змінної	Тип даних	Поточне значення
PID0	PID	
D_SV	REAL	
D_PV	REAL	
D_MODE	DINT	
D_MAN	BOOL	
D_CYC	DINT	
D_Kc	REAL	
D_Ki	REAL	
D_Kd	REAL	
D_Tf	REAL	
D_DIR	BOOL	
D_DBW	REAL	
D_MAX	REAL	
D_MIN	REAL	
D_MOUT	REAL	
D_BIAS	REAL	
D_I_MV	REAL	
D_MV	REAL	

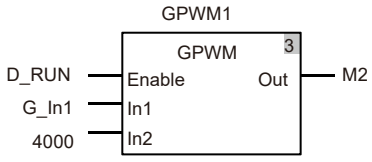
3. Програма

a. POU1 (Циклічне завдання) :

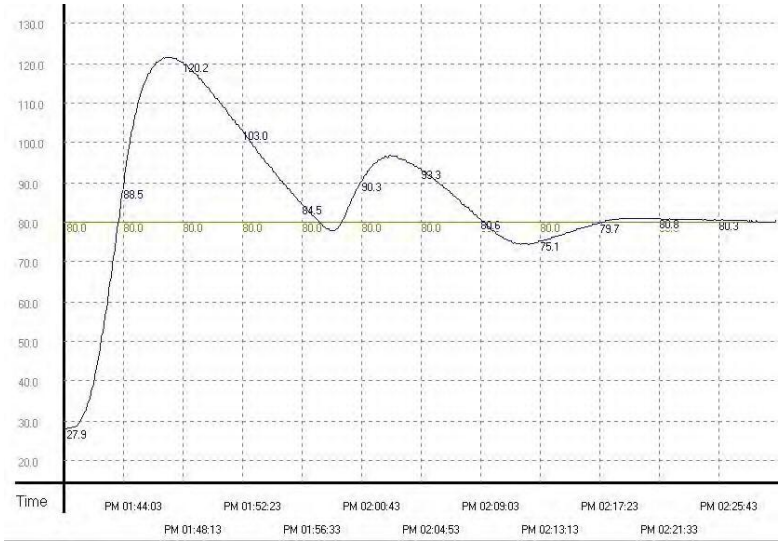


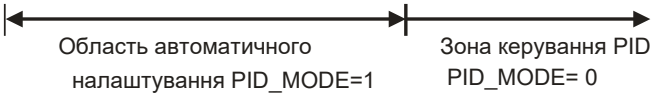


b. POU2 (Завдання вільного ходу) :

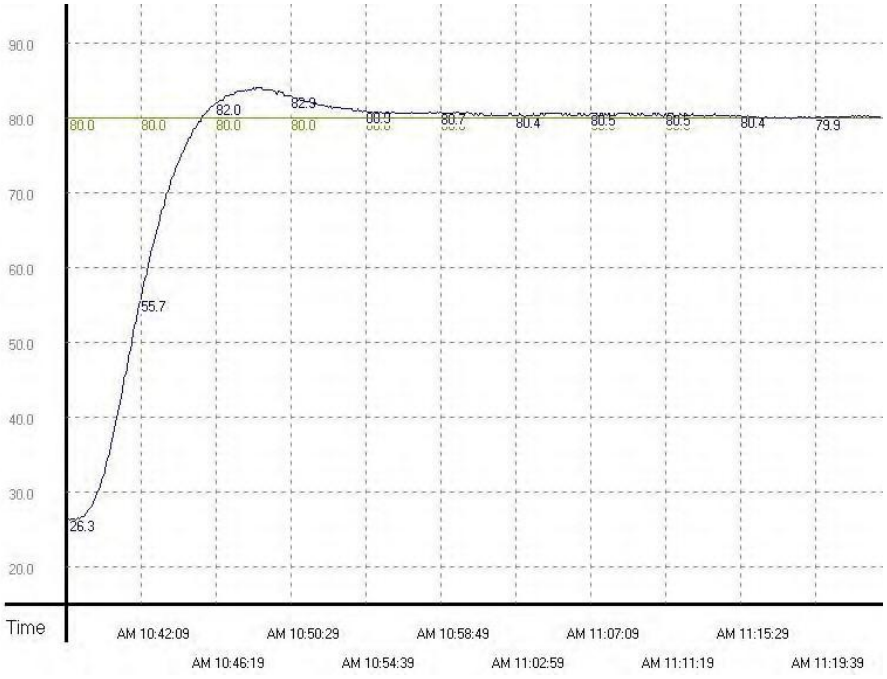


в. Експериментальний результат функції автоматичного налаштування наведено нижче.

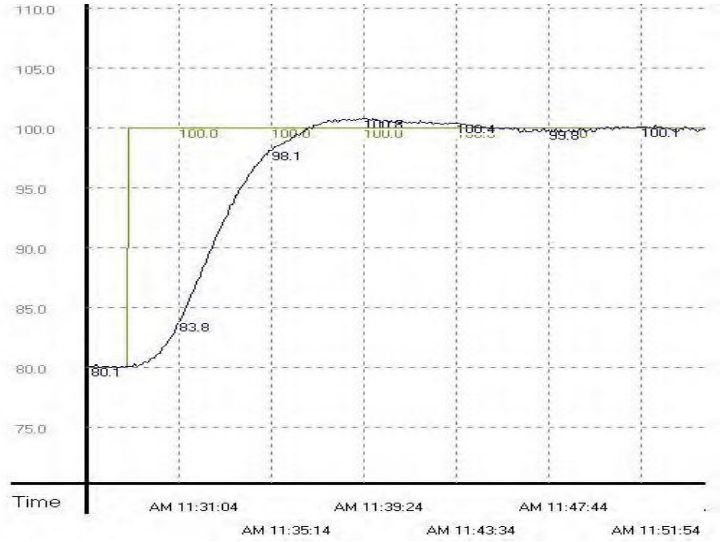




d. Експериментальний результат використання параметрів, налаштованих для контролю температури, наведено нижче.



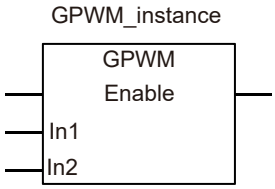
e. Як показано на діаграмі вище, після автоматичного налаштування параметрів користувачі можуть отримати хороший результат контролю температури. Щоб контролювати температуру, потрібно лише близько двадцяти хвилин. Коли цільова температура змінюється від 80°C до 100°C, результат такий, як показано нижче



f. Як показано на діаграмі вище, коли цільова температура змінюється від 80°C до 100°C, попередньо налаштовані параметри все ще можна використовувати для контролю температури. Крім того, для контролю температури не потрібно багато часу.

8.17.2 GPWM

FB/FC	Пояснення	Застосовна модель
FB	Інструкція GPWM використовується в імпульсному виході.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Enable	Увімкнути	Вхід	Виконання інструкції починається, коли *Увімкнути* зміни з FALSE на TRUE.	Правда чи брехня
In1	Ширина вихідного імпульсу	Вхід	Після виконання інструкції встановіть тривалість вихідного імпульсу (мс).	0~32767
In2	Вихідний цикл імпульсу	Вхід	Коли почнеться виконання інструкції, встановіть цикл вихідного імпульсу (мс).	1~32767
Out	Реєстр для Вихід імпульсу	Вихід	Вихід є TRUE в межах ширини вихідного імпульсу.	Правда чи брехня

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING
Enable	●																			
In1											●									
In2											●									
Out	●																			

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

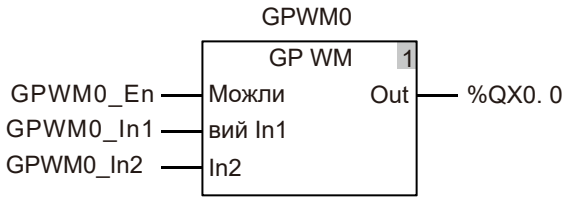
● **Пояснення функції**

- Інструкція GPWM використовується в імпульсному виході.
- Будь ласка, використовуйте інструкцію GPWM у завданні вільного ходу. В іншому випадку вихід інструкції GPWM може бути неточним.
- Значення In1 і In2 можна змінювати під час виконання інструкції GPWM.
- Коли In1 ≤ 0, вихідний регістр імпульсу не має виходу. Коли In1 ≥ in2, вихідний регістр імпульсів завжди увімкнено.
- Вихід інструкції GPWM може використовувати змінну або будь-який бітовий регістр. Для отримання додаткової інформації зверніться до «Розділ 3.1.2 Регістри та типи даних».

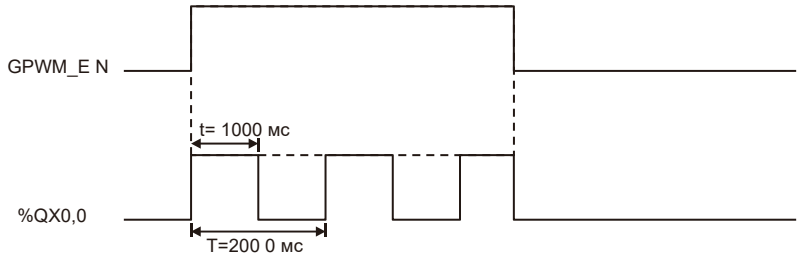
Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
GPWM0	GPWM	
GPWM0_En	BOOL	FALSE
GPWM0_In1	INT	1000
GPWM0_In2	INT	2000



■ **Графік часу**



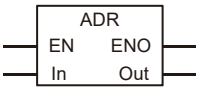
■ **Додаткове пояснення:**

Коли GPWM_EN змінюється на TRUE, інструкція працює в звичайному режимі.
Коли GPWM_EN змінюється на FALSE, вихід інструкції змінюється на FALSE.

8.18 Адресна інструкція

8.18.1 ADR

FB/FC	Пояснення	Застосовна модель
FC	ADR використовується для отримання адреси змінної.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
In	Введіть значення	Вхід	Змінна, адреса якої має бути виведена	Залежить від типу даних змінної, з якою пов'язаний вхідний параметр.
Out	Результат операції	Вихід	Адреса вихідної змінної IN	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				Рядок	покажчик
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	STRING	Покажчик на xx
In	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Out																					●

Примітка:

1. Символ ● вказує на те, що параметру дозволено підключатися до змінної типу даних.
2. Тип даних xx ' **Pointer To** xx' має бути таким самим, як і *IN* .

● **Пояснення функції**

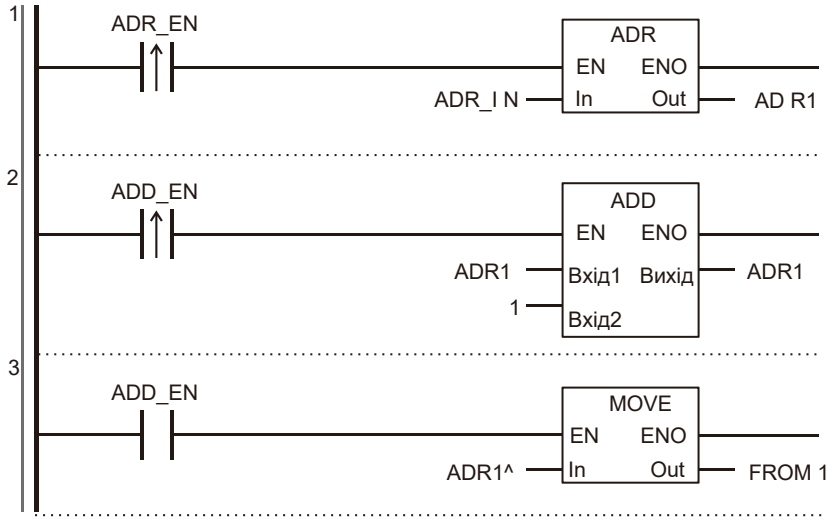
■ Інструкція ADR використовується для отримання адреси змінної. Отримайте адресу змінної, а потім знайдіть значення, що зберігається в адресі змінної, використовуючи символ ^.

Приклад програмування

■ Інструкція ADR використовується для отримання адреси вхідної змінної ADR_In і розміщення її у вихідній змінній ADR1. Адреса ADR_In – %MW0. Після виконання інструкції ADR ADR1 вказує на %MW0. Після виконання ADD ADR1 вказує на %MW1. ADR1^ означає отримати значення з адреси ADR1. (Значення ADR1 дорівнює %MW1, а ADR1^ означає отримати значення в %MW1.) Значення в %MW1 переміщується до змінної OUT1 за допомогою виконання інструкції MOVE.

➢ **Таблиця змінних і програма**

Ім'я змінної	Адреса	Тип даних
ADR_EN		BOOL
ADR_In	%MW0	INT
ADR1		POINTER TO INT
OUT1		INT



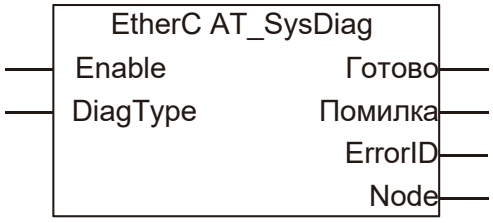
8.19 Діагностика мережі

8.19.1 EtherCAT Діагностика

8.19.1.1 EtherCAT_SysDiag

FB/FC	Пояснення	Застосовна модель
FB	EtherCAT_SysDiag використовується для діагностики системи EtherCAT лише за допомогою порту EtherCAT, вбудованого в контролер руху.	AS516E-B AS532EST-B AS564EST-B

EtherC AT_SysDiag_instance



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	Коли *Увімкнути* змінюється на TRUE
DiagType	1: Чи була вісь налаштована в програмному забезпеченні; 2: Чи встановила вісь з'єднання з портом EtherCAT; 3: Чи надіслала вісь екстрене повідомлення.	USINT	1, 2, 3	Коли *Увімкнути* змінюється на TRUE

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-
Вузол	Виводить відповідні стани всіх осей на основі вхідного значення DiagType.	Array[1..32] BOOL	-

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ TRUE, коли виконання інструкції завершено.	◆ Коли *Enable* змінюється на FALSE ◆ Коли помилка змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли нестандартна ситуація усунена.

● **Функції**

EtherCAT_SysDiag використовується для діагностики стану ведених пристроїв, підключених до порту EtherCAT. Цю інструкцію може використовувати лише порт EtherCAT, вбудований у контролер руху.

*Я*кщо значення *Node* є масивом BOOL, можна вивести стани 1~32 осей. *Я*кщо значення *DiagType* відрізняється, значення *Node* представляє інше значення.

Коли значення *DiagType* дорівнює 1, стан Node означає, чи вісь налаштовано в програмному забезпеченні.

*Я*кщо *Node* має значення TRUE, це означає, що вісь налаштовано в програмному забезпеченні. *Я*кщо *Node* має значення FALSE, це означає, що вісь не було налаштовано в програмному забезпеченні.

Коли значення *DiagType* дорівнює 2, стан ***вузла*** вказує, чи встановила вісь з'єднання з портом EtherCAT.

*Я*кщо *Node* має значення TRUE, вісь встановила з'єднання з портом EtherCAT. *Я*кщо *Node* має значення FALSE, вісь не встановила з'єднання з портом EtherCAT.

*Я*кщо значення *DiagType* дорівнює 3, значення *Node* показує, чи надсилала вісь екстрене повідомлення.

*Я*кщо *Node* має значення TRUE, вісь надіслала сигнал тривоги. *Я*кщо *Node* має значення FALSE, аварійний сигнал осі стерто.

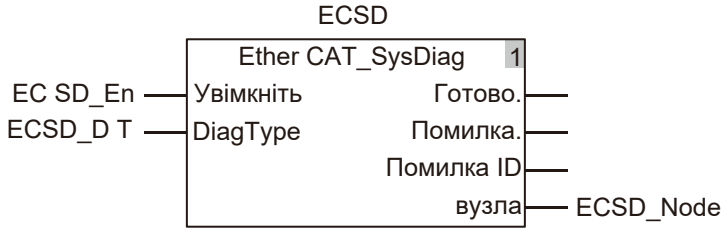
*Я*кщо значення *DiagType* дорівнює 2, а *Node* відповідає змінній a, a[1] має значення TRUE, коли вісь № 1 з'єднується з портом EtherCAT. a[1] має значення FALSE, коли кабель EtherCAT, під'єднаний до осі1, видалено.

Приклад програмування

1. Таблиця змінних

Ім'я змінної	Тип даних	Початкове значення
ECSD	EtherCAT_SysDiag	
ECSD _EN	BOOL	
ECSD _DT	USINT	2
ECSD _Node	Array[1..32] BOOL	

2. Програма



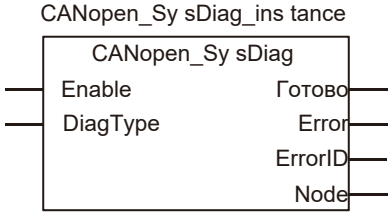
Ви можете дізнатися, чи встановила відповідна вісь з'єднання з портом EtherCAT, за допомогою значень елементів масиву ECSD_Node.
Якщо ECSD_Node[1]=1, це означає, що вісь №1 з'єдналася з портом EtherCAT.

Якщо ECSD_Node[1]=0, це означає, що вісь №1 не з'єдналася з портом EtherCAT. Це може бути тому, що кабель зв'язку не під'єднано або від'єднано після під'єднання. Якщо ви хочете визначити, чи вісь № 2 з'єдналася з портом EtherCAT, використовуйте значення ECSD_Node[2]. Для інших осей таким же способом можна оцінити стан з'єднання з портом EtherCAT.

8.19.2 Діагностика CANopen

8.19.2.1 CANopen_SysDiag

FB/FC	Пояснення	Застосовна модель
FB	CANopen_SysDiag використовується для діагностики стану всіх підлеглих пристроїв, підключених до порту CANopen.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Увімкнути* змінюється на TRUE.
DiagType	1: чи налаштовано підлеглі пристрої в програмному забезпеченні. 2: Чи підключено підлеглі пристрої та порт CANopen. 3: Чи видано підлеглі екстрене повідомлення.	USINT	1,2,3 (Необхідно встановити значення змінної)	Коли *Увімкнути* змінюється на TRUE.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Вузол	Виводить відповідні стани всіх підлеглих пристроїв на основі різних значень *DiagType.*	Array[1..32] BOOL	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ TRUE, коли виконання інструкції вдалось.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється на FALSE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли помилка очищена.

- **Функції**

  CANopen_SysDiag використовується для діагностики стану всіх підлеглих пристроїв, підключених до порту CANopen.

(Лише порт CANopen, вбудований у контролер руху, може використовувати цю інструкцію.)

Вузол Вихід — це масив BOOL для Вихід станів 1~32 осей. Значення *Node* має інше значення, коли значення вхідного *DiagType* змінюється.

Коли значення вхідного *DiagType* дорівнює 1, значення *Node* означає, чи налаштовано підлеглі пристрої в програмному забезпеченні. TRUE означає, що підлеглі налаштовані в програмному забезпеченні. FALSE означає, що підлеглі пристрої не налаштовані в програмному забезпеченні.

Коли значення вхідного *DiagType* дорівнює 2, значення *Node* означає, чи підлеглі пристрої встановлюють з'єднання з портом CANopen. TRUE означає, що підключення встановлено. FALSE означає, що підключення не встановлено.

Коли значення вхідного *DiagType* дорівнює 3, значення *Node* означає, чи підпорядковані пристрої випускають екстрене повідомлення. TRUE означає, що екстрене повідомлення надіслано. FALSE означає, що екстрене повідомлення не надіслано.

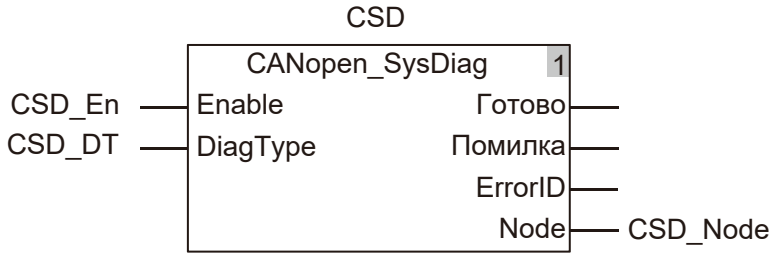
Наприклад, якщо значення *DiagType* дорівнює 2, а змінна вихідного *вузла* — a, значення a[1] є TRUE, коли підлеглий пристрій 1 і порт CANopen встановлюють з'єднання, а значення a[1] — FALSE, коли кабель CANopen підлеглого 1 видалено.

Приклад програмування

1. Таблиця змінних

Ім'я змінної	Тип даних	Початкове значення
CSD	CANOpen_SysDiag	
CSD_EN	BOOL	
CSD_DT	USINT	2
CSD_Node	ARRAY[1..32] BOOL	

2. Програма



Зі значень членів масиву CSD_Node можна визначити, чи підключено відповідний підлеглий пристрій до порту зв'язку CANopen контролера чи ні.

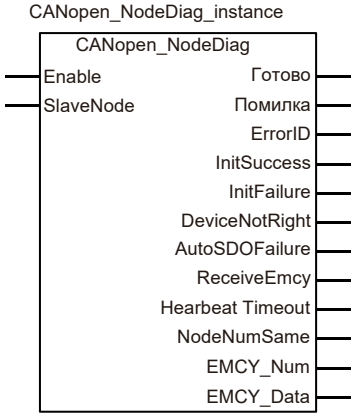
Якщо CSD_Node[1]=1, це вказує на те, що вісь 1 і комунікаційний порт CANopen підключено, якщо CSD_Node[1]=0, це означає, що вісь 1 і комунікаційний порт CANopen не підключено. Можлива причина полягає в тому, що кабель зв'язку між портом зв'язку CANopen і віссю 1 не під'єднано належним чином або кабель зв'язку від'єднано після під'єднання.

Для підключення осі 2 до комунікаційного порту CANopen ви можете використовувати значення CSD_Node[2], щоб судити про успішне підключення чи ні.

Для підключення інших осей до комунікаційного порту CANopen використовуйте той самий спосіб, щоб визначити, чи підключення успішне чи ні.

8.19.2.2 CANopen_NodeDiag

FB/FC	Пояснення	Застосовна модель
FB	CANopen_NodeDiag використовується для діагностики стану зазначеного підлеглого пристрою, підключеного до порту CANopen.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Увімкнути* змінюється на TRUE.
SlaveNode		USINT	1~32 (Необхідно встановити значення змінної)	Коли *Увімкнути* змінюється на TRUE.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
InitSuccess	ІСТИНА, коли ініціалізація підлеглого пристрою вдається.	BOOL	ІСТИНА / НЕПРАВДА
InitFailure	ІСТИНА, коли ініціалізація підлеглого не вдається.	BOOL	ІСТИНА / НЕПРАВДА
DeviceNotRight	ІСТИНА, коли підпорядкований реєстр неправильний.	BOOL	ІСТИНА / НЕПРАВДА
Помилка AutoSDOF	ІСТИНА, коли налаштування для автоматичного SDO підлеглого не вдається.	BOOL	ІСТИНА / НЕПРАВДА
ReceiveEmcy	TRUE, коли підлеглий пристрій отримує екстрене повідомлення.	BOOL	ІСТИНА / НЕПРАВДА
HeartbeatTimeout	ІСТИНА, коли настає тайм-аут повідомлення серцевого ритму.	BOOL	ІСТИНА / НЕПРАВДА
NodeNumSame	ІСТИНА, коли адреси станції головного та підлеглих дублюються.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
EMCY_Num	Записує номер екстреного повідомлення, яке отримує контролер.	USINT	0~5
EMCY_Data	Записує екстрене повідомлення, яке контролер отримує від підлеглого.	ARRAY[1..5] З CANopen_EMCY_Type	

- **Час оновлення**

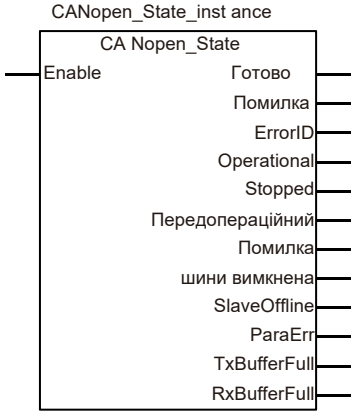
Ім'я	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли вказівка виконана успішно.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється на FALSE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли помилка очищена.

- **Функції**

  CANopen_NodeDiag використовується для діагностики стану зазначеного підлеглого пристрою, підключеного до порту CANopen. (Лише порт CANopen, вбудований у контролер руху, може використовувати цю інструкцію.) Якщо підлеглий пристрій, налаштований у програмному забезпеченні, не може працювати нормально, інструкцію можна використати для діагностики причини помилки та запису кількості разів, коли підлеглий пристрій надсилає екстрене повідомлення. повідомлення та дані екстреного повідомлення.

8.19.2.3 CANopen_State

FB/FC	Пояснення	Застосовна модель
FB	CANopen_State використовується для діагностики стану порту CANopen.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Увімкнути* змінюється на TRUE.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли виконання інструкції завершено.	BOOL	ІСТИНА/НЕПРАВДА
Помилка	TRUE, коли є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Оперативний	ІСТИНА, коли головний перебуває в робочому стані.	BOOL	ІСТИНА / НЕПРАВДА
Зупинився	ІСТИНА, коли головний пристрій перебуває в стані зупинки.	BOOL	ІСТИНА / НЕПРАВДА
Передопераційний	ІСТИНА, коли головний пристрій знаходиться в передопераційному стані.	BOOL	ІСТИНА / НЕПРАВДА
BusOff	ІСТИНА, коли перешкоди шини надто сильні або в мережі існують продукти різних швидкостей передачі даних.	BOOL	ІСТИНА / НЕПРАВДА
BusError	ІСТИНА, коли виникає помилка шини.	BOOL	ІСТИНА / НЕПРАВДА
SlaveOffline	ІСТИНА, коли деякий підлеглий не в мережі.	BOOL	ІСТИНА / НЕПРАВДА
ParaError	ІСТИНА, коли виникає помилка параметрів конфігурації головного та підлеглого.	BOOL	ІСТИНА / НЕПРАВДА
TxBufferFull	ІСТИНА, коли область буфера для головного надсилання даних заповнена.	BOOL	ІСТИНА / НЕПРАВДА
RxBufferFull	TRUE, коли область буфера для головного пристрою для отримання даних заповнена.	BOOL	ІСТИНА / НЕПРАВДА

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли вказівка виконана успішно.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється на FALSE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли помилка очищена.

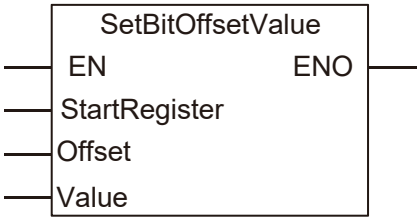
● **Функції**

CANopen_State використовується для діагностики стану порту CANopen (тільки порт CANopen, вбудований у контролер руху, може використовувати цю інструкцію.) Він може повідомляти про стан порту CANopen і про те, чи будь-який підлеглий пристрій перебуває в автономному режимі.

8.20 Значення біта зсуву читання та запису

8.20.1 SetBitOffsetValue

FB/FC	Пояснення	Застосовна модель
FC	SetBitOffsetValue використовується для встановлення значення певного біта вказаної адреси на TRUE або FALSE.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Почніть реєстрацію	Початкова адреса	Вхід	Вкажіть початкову адресу	
Зсув	Скільки бітів зміщено	Вхід	Укажіть значення зсуву, розглядаючи біт0 змінної, на яку вказує *StartRegister* , як на посилання.	0~1023
Значення	Вхідне значення	Вхід	Введіть вказане бітове значення	ІСТИНА / НЕПРАВДА

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				покажчик
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	Покажчик на XX
StartRegister																				●
Зсув			●				●													
Значення	●																			

Примітка:

1. Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.
2. xx типу даних «Покажчик на xx» включає типи даних Bit string і Integer.

● **Пояснення функції**

Інструкція SetBitOffsetValue використовується для встановлення значення певного біта вказаної адреси на TRUE або FALSE. Параметр *StartRegister* є змінною типу Pointer. Параметр *Offset* визначає значення зміщення, яке розглядає біт0 змінної, на яку вказує параметр *StartRegister* , як посилання. Параметр *Value* показує значення параметра зазначеного біта, яке є TRUE або FALSE. Біт, який разом визначено *StartRegister* і *Offset,* можна встановити на TRUE або FALSE за допомогою цієї інструкції.

Якщо вказане значення зсуву перевищує діапазон початкової адреси *StartRegister* , операція буде виконана відповідно до відповідної адреси зсуву, яка розглядає StartRegister як початкову адресу.

Дивіться більше в наступному прикладі.

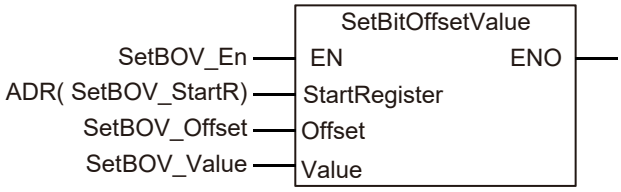
● **Застереження щодо правильного використання**

Параметр *StartRegister* інструкції SetBitOffsetValue є змінною типу Pointer і для отримання адреси потрібно використовувати інструкцію ADR.

Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Адреса пристрою	Тип даних	Поточна вартість
SetBOV_EN		BOOL	TRUE
SetBOV _StartR	%MW0	UINT	
SetBOV _Offset		UINT	
SetBOV _Value		BOOL	



■ **Program Explanation**

Коли змінна SetBOV_EN має значення TRUE, установіть значення зазначеного біта на значення SetBOV_Value, змінивши значення змінних SetBOV_Offset і SetBOV_Value.

Дивіться пояснення нижче.

1. Якщо SetBOV _Value встановлено на FALSE, а значення SetBOV _Offset дорівнює 2, значення Bit2 SetBOV _StartR （ %MW0 ） є FALSE.

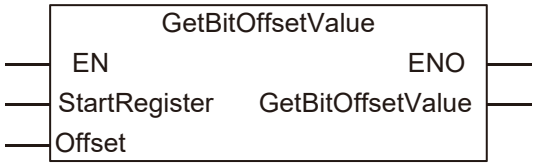
2. Коли SetBOV _Value встановлено на TRUE, а значення SetBOV _Offset дорівнює 2, значення Bit2 SetBOV _StartR （ %MW0 ） є TRUE.

3. Коли для SetBOV _Value встановлено значення FALSE, а значення SetBOV _Offset дорівнює 16 (що розглядає біт0 змінної, на яку вказує StartRegister, як еталонний біт), значення біт1
%MW1 є ЛОЖНІЮ.

4. Коли для SetBOV _Value встановлено значення TRUE, а значення SetBOV _Offset дорівнює 16 (що розглядає біт0 змінної, на яку вказує StartRegister, як еталонний біт), значення біт1
%MW1 є ІСТИНОЮ.

8.20.2 GetBitOffsetValue

FB/FC	Пояснення	Застосовна модель
FC	GetBitOffsetValue використовується для читання значення певного біта вказаної адреси та відображення його у вихідному параметрі.	AS516E-B AS532EST-B AS564EST-B



● **Параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
StartRegister	Початкова адреса	Вхід	Вкажіть початкову адресу	
Offset	Скільки бітів зміщено	Вхід	Укажіть значення зсуву, розглядаючи біт0 змінної, на яку вказує *StartRegister* , як на посилання	0~1023
GetBitOffsetValue	Значення біта читання	Вихід	Значення біта читання	ІСТИНА / НЕПРАВДА

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				покажчик
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	Покажчик на XX
StartRegister																				●
Offset			●				●													
GetBitOffsetValue	●																			

Примітка:

1. Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.
2. xx типу даних «Покажчик на xx» включає типи даних Bit string і Integer.

● **Пояснення функції**

Інструкція GetBitOffsetValue використовується для читання значення зазначеного біта, а вихід *GetBitOffsetValue* показує значення стану читання. Параметр *StartRegister* є змінною типу Pointer. Параметр *Offset* визначає значення зміщення, яке розглядає біт0 змінної, на яку вказує параметр *StartRegister* , як посилання. Параметр *GetBitOffsetValue* показує значення стану зазначеного біта, яке є TRUE або FALSE.

Якщо вказане значення зсуву перевищує діапазон початкової адреси *StartRegister* , операція буде виконана відповідно до відповідної адреси зсуву, яка розглядає StartRegister як початкову адресу. Дивіться більше в наступному прикладі.

● **Застереження щодо правильного використання**

Параметр *StartRegister* інструкції GetBitOffsetValue є змінною типу Pointer і для отримання адреси потрібно використовувати інструкцію ADR.

Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Адреса пристрою	Тип даних	Поточна вартість
SetBOV_EN		BOOL	
SetBOV _StartR	%MW5	UINT	
SetBOV _Offset		USINT	
GBOValue		BOOL	



■ Пояснення програми

Коли змінна SetBOV_EN має значення TRUE, зчитувати стан зазначеного біта шляхом зміни значення параметра *Offset* . Дивіться пояснення нижче.

1. Коли значення SetBOV _StartR (%MW5) дорівнює 8 (2#0000,1000), а SetBOV _Offset дорівнює 3, значення GBOValue дорівнює TRUE, що є значенням біта 3 %MW5.

1. Коли значення SetBOV _StartR дорівнює 0, а SetBOV _Offset дорівнює 3, значення GBOValue дорівнює FALSE, що є значенням Біта 3 %MW5.

1. Коли значення %MW6 дорівнює 8 (2#0000,1000), а SetBOV _Offset дорівнює 19 (що розглядає біт0 змінної, на яку вказує *StartRegister* , як посилання), значення GBOValue дорівнює TRUE, що є значенням біт3 %MW6.

4. Коли значення SetBOV _StartR дорівнює 0, а SetBOV _Offset — 19 (що розглядає біт0 змінної, на яку вказує *StartRegister* , як посилання), значенням GBOValue є FALSE, що є значенням біт3 %MW6.

8.21 Інструкція FCS

8.21.1 CRC16

FB/FC	Пояснення	Застосовна модель
FC	Інструкція CRC16 використовується для обчислення значення CRC зазначених даних.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Дані	Початкова адреса даних для розрахунку значення CRC	Вхід	Початкова адреса даних для розрахунку значення CRC. Початкову адресу можна отримати за інструкцією ADR	——
Довжина	Довжина даних для розрахунку значення CRC	Вхід	Скільки байтів даних для обчислення значення CRC шляхом підрахунку від початкової адреси даних з одиницею: байт.	1~255
OutOrder	Встановіть розташування байтів вихідного значення CRC	Вхід	FALSE означає, що для вихідного значення CRC його молодший байт знаходиться зліва від старшого байта. TRUE означає, що для вихідного значення CRC молодший байт знаходиться праворуч від старшого байта.	Правда чи брехня
CRC16	Значення CRC	Вихід	Значення CRC отримано через обчислення на основі параметра *Data*	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				ПОКАЗЧИК
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	ПОКАЖЧИК НА BYTE
Дані																				●
Довжина							●													
OutOrder	●																			
CRC16			●																	

Примітка:

Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція CRC16 використовується для обчислення значення CRC зазначених даних.

Інструкція виконує розрахунок значення CRC відповідно до початкової адреси, заданої *Дані* та кількість байтів, визначених *Length* .

Результат обчислення поміщається у вихідний параметр *CRC16* . Розташування байтів значення CRC визначається *OutOrder* .

● **Застереження щодо правильного використання**

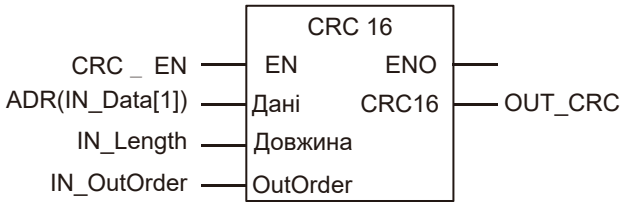
Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена.

Приклад програмування

■ **Таблиця змінних і програма**

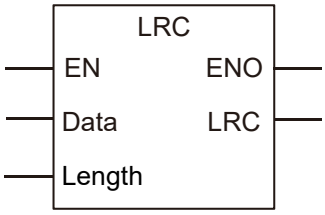
Ім'я змінної	Адреса	Тип даних	Поточне значення
CRC_EN		BOOL	TRUE
IN_Data	%MB100	МАСИВ [1..6] БАЙТІВ	[16#01,16#03,16#10,16#01,16#00,16#02]
IN_Length		UINT	6
IN_OutOrder		BOOL	FALSE
OUT_CRC		WORD	16#910B

■ Коли CRC_EN змінюється з FALSE на TRUE, інструкція CRC16 перевіряє 6 байтів [16#01,16#03,16#10,16#01,16#00,16#02] в IN_Data.
Результатом обчислення є 16#910B, який поміщається в OUT_CRC.



8.21.2 LRC

FB/FC	Пояснення	Застосовна модель
FC	Інструкція LRC використовується для знаходження значення LRC зазначених даних.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	Значення	Вхід / Вихід	опис	Допустимий діапазон
Дані	Початкова адреса даних для розрахунку значення LRC	Вхід	Початкова адреса даних для розрахунку значення LRC. Початкову адресу можна отримати за інструкцією ADR	——
Довжина	Довжина даних для розрахунку значення LRC	Вхід	Скільки байтів даних для обчислення значення LRC шляхом підрахунку від початкової адреси даних з одиницею: байт.	1~255
LRC	Значення LRC	Вихід	Значення LRC зазначених даних	Залежить від типу даних змінної, до якої підключено вихідний параметр.

	Логічний	Бітовий рядок				Ціле число								Реальне число		Час, дата				ПОКАЗЧИК
	BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL	TIME	DATE	TOD	DT	ПОКАЖЧИК НА BYTE
Дані																				●
Довжина							●													
LRC		●																		

Примітка:
Символ ● вказує, що параметру дозволено підключатися до змінної або константи типу даних.

● **Пояснення функції**

Інструкція LRC використовується для знаходження значення LRC зазначених даних.

Інструкція виконує розрахунок значення LRC відповідно до початкової адреси, визначеної *Дані* та кількість байтів, визначених *довжиною.*
Результат розрахунку поміщається у вихідний параметр *LRC* .

● **Застереження щодо правильного використання**

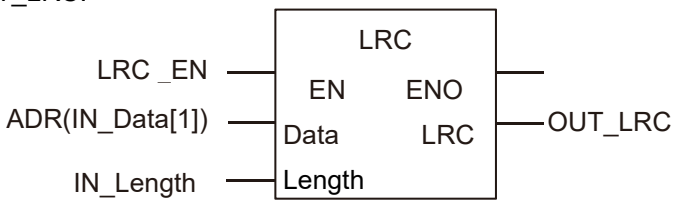
Вхідні змінні не можна опускати. Під час компіляції програмного забезпечення виникне помилка, якщо будь-яка вхідна змінна пропущена.

Приклад програмування

■ Таблиця змінних і програма

Ім'я змінної	Адреса	Тип даних	Поточне значення
LRC_EN		BOOL	TRUE
IN_Data	%MB100	МАСИВ [1..6] БАЙТІВ	[16#01,16#03,16#10,16#01,16#00,16#02]
IN_Length		UINT	6
OUT_LRC		WORD	16#E9

■ Коли LRC_EN змінюється з FALSE на TRUE, Інструкція LRC перевіряє 6 байтів [16#01,16#03,16#10,16#01,16#00,16#02] в IN_Data. Результатом обчислення є 16#E9, який поміщається в OUT_LRC.



8.22 Інструкції для правого модуля розширення

8.22.1 Інструкції щодо AS02PU та AS04PU

8.22.1.1 Умови застосування відповідних інструкцій AS02PU та AS04PU

- **Примітки:**

1. Інструкції, пов'язані з AS02PU та AS04PU, доступні лише для модуля PU з правого боку ПЛК серії AS500. Але вони не застосовуються до модулів PU, підключених до правого боку віддалених модулів.
2. Контролер серії AS500 із мікропрограмою версії 1.03 або новішої підтримує інструкції, пов'язані з AS02PU та AS04PU.
3. Лише AS02PU з мікропрограмою версії 1.02.00 або новішої підтримує інструкцію PUCNT.
4. Додаткову інформацію про модулі AS02PU та AS04PU див. у ***посібнику з модулів серії AS*** .

8.22.1.2 Інструкції щодо AS02PU та AS04PU

8.22.1.2.1. PUCONF

FB/FC	Пояснення	Застосовна модель
FB	PUCONF використовується для зміни параметрів керування для вказаного модуля PU.	AS516E-B AS532EST-B AS564EST-B

PUCONF_instance



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, *вихід помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Вісь	Вкажіть номер осі для зазначеного модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Режим	Вкажіть режим Вихід вказаної осі	USINT	0~255 (1)	Коли *Enable* має значення TRUE
Швидкість	Початкова/кінцева швидкість Одиниця: Гц	UINT	0~10000 (100)	Коли Enable має значення TRUE
Час	Вкажіть час прискорення Одиниця: мс	UINT	0~10000 (100)	Коли *Enable* має значення TRUE
Dчас	Вкажіть час уповільнення Одиниця: мс	UINT	0~10000 (100)	Коли *Enable* має значення TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Мшвидкість	Вкажіть максимальну вихідну швидкість. Одиниця: Гц	UDINT	AS02PU: 100~200000 (200000) AS04PU: 100~100000 (100000）	Коли *Enable* має значення TRUE
З_но	Укажіть кількість сигналів фази Z для пошуку після повернення до початку координат.	INT	-100~100 (0)	Коли *Enable* має значення TRUE
Зсув	Укажіть положення вихідного зсуву після завершення наведення та пошуку сигналів фази Z.	INT	-10000~10000 (0)	Коли *Enable* має значення TRUE

Примітка:

1. Контролер серії AS500 з мікропрограмою версії 1.03 або новішої підтримує інструкцію PUCONF.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0/ Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0/ Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *Режим* встановлює режим Вихід вихідної осі, а значення налаштувань пояснюються в наступній таблиці.

Значення режиму Вихід	опис	Зауваження
0	Одноточковий імпульсний вихід (парна точка лише для виходу)	Наприклад, Y0.0 або Y0.2 для Вихід
1	Пульс (точка з парним числом) + напрямок (точка з непарним числом)	Наприклад, Y0.0 для імпульсу, а Y0.1 для напрямку. Y0.1: ON для негативного напрямку; Y0.1: ВИМК для позитивного напрямку
2	CW (точка парного числа) + CCW (точка з непарним числом)	Наприклад, Y0.0 для CW (позитивний напрямок), а Y0.1 для CCW (негативний напрямок)
3	Фаза A (Точка з парним номером) + Фаза B (Точка з непарним номером)	Наприклад, Y0.0 для фази A, а Y0.1 для фази B. Коли фаза A є провідною, фаза B: позитивний напрямок; коли фаза B є провідною, фаза A: негативний напрямок
Інше значення	Автоматичний перехід до режиму 1 (значення за замовчуванням)	

- **Вихідні параметри**

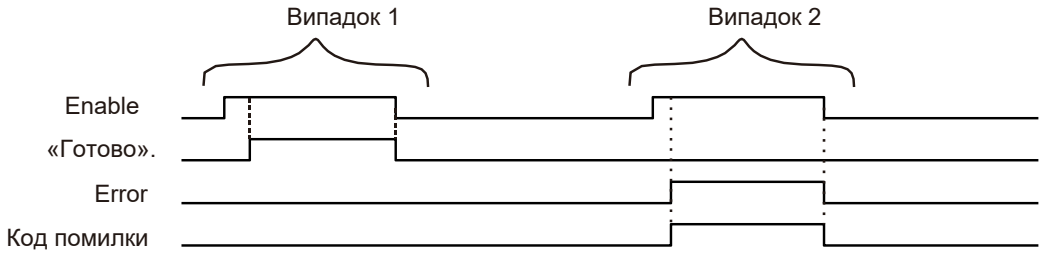
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	TRUE, коли налаштування параметрів для вихідної осі модуля PU завершено.	BOOL	ІСТИНА / НЕПРАВДА

Назва параметра	функція	Тип даних	Допустимий діапазон
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Після завершення налаштування параметрів для вказаної осі	◆ Коли вісь відключена ◆ Коли *помилка* змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими. Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1: *Готово* змінюється на ІСТИНА, коли налаштування параметрів осі для вказаної вихідної осі завершено після того, як *Увімкнути* зміниться з ФАЛШ на ІСТИНА. *Значення Done* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE і *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

● **функція**

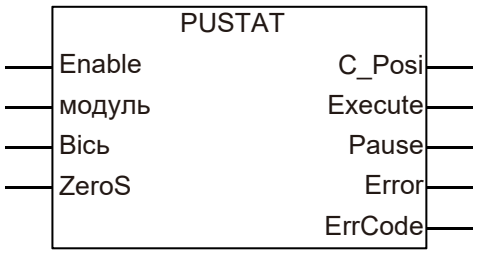
PUCONF використовується для встановлення параметрів керування для модуля PU.

1. Усі параметри керування, встановлені в інструкції, можна встановити з програмного забезпечення. Інструкцію пропонується виконувати у випадку, якщо значення відповідних параметрів необхідно змінити кілька разів під час виконання програми.
2. Інструкція PUCONF є імпульсною інструкцією. Параметри модуля PU встановлюються при включеній інструкції. Таким чином, якщо значення параметра потрібно оновити, повторно активуйте *Enable* інструкції для скидання параметра.
3. Оскільки встановлені параметри доставляються через команду зв’язку модуля, підтвердьте стан Вихід *«Готово»* або *«Помилка»* , перш ніж значення параметра буде змінено, а потім продовжуйте відповідні операції.
4. Виконання інструкції PUCONF дійсне, коли немає вихідного імпульсу з вихідних точок модуля PU. Однак вхідні параметри не вступлять в силу після виконання інструкції PUCONF, поки вихідні точки модуля PU виводять імпульси.

8.22.1.2.2. PUSTAT

FB/FC	Пояснення	Застосовна модель
FB	PUSTAT читає вихідний стан модуля PU.	AS516E-B AS532EST-B AS564EST-B

PUSTAT_примірник



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Вісь	Вкажіть номер осі модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
ZeroS	Очистити поточну вихідну позицію до 0	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE

8

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUSTAT.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

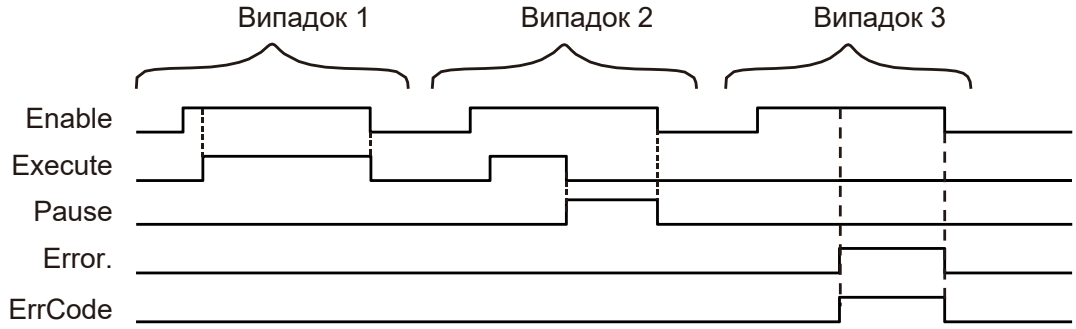
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
C_Posi	Виводить поточне положення вихідної осі для вказаного модуля PU.	DINT	-
Виконати	TRUE, коли інструкція виконується.	BOOL	ІСТИНА / НЕПРАВДА
Пауза	TRUE, коли виконання інструкції призупинено.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Виконати	◆ Коли інструкція виконується.	◆ Коли вказана вісь зупиняється ◆ Коли зазначена вісь призупиняє роботу ◆ Коли *помилка* змінюється на TRUE
Пауза	◆ Коли виконання інструкції призупинено.	◆ Коли зазначена вісь зупиняється. ◆ Поки зазначена вісь працює ◆ Коли *помилка* змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1: коли *Enable* змінюється з FALSE на TRUE, а вихідна швидкість імпульсів у вихідних точках модуля PU не дорівнює 0, *Execute* змінюється з FALSE на TRUE. Після того, *як Enable* змінює значення з TRUE на FALSE, *Execute* змінює значення з TRUE на FALSE.

Випадок 2: коли *Enable* змінюється з FALSE на TRUE, а вихідна швидкість імпульсів у вихідних точках модуля PU дорівнює 0, *Pause* змінюється з FALSE на TRUE. Після того, як *«Увімкнути»* змінюється з «ІСТИНА» на «ХИБНЯ», *«Пауза»* змінюється з «ІСТИНА» на «ХИБНЯ».

Випадок 3: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE і *ErrCode* показує відповідні коди помилок і тим часом змінює *Done* на FALSE. Коли *Увімкнути* змінюється на TRUE і помилка скидається, *Error* змінюється на FALSE.

● **функція**

Інструкція PUSTAT зчитує стан модуля PU в реальному часі.

1. *C_Posi* відображає поточне положення вихідної осі для вказаного модуля PU. Значення параметра є зафіксованим значенням і зберігається в модулі PU. Якщо значення має бути очищено, встановіть для *ZeroS* значення TRUE під час запуску інструкції.
2. *Execute* — це вихід лише для читання, який показує, чи виводить вісь Вихід зазначеного модуля PU чи ні. Якщо *Execute* має значення True, це означає, що вивід виконується. Коли *Виконати*

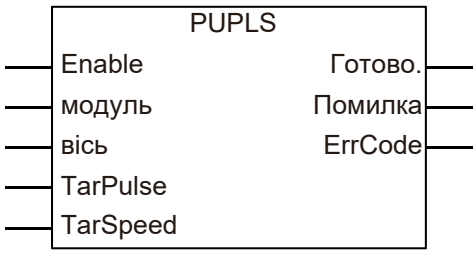
FALSE, це означає, що вихідна вісь не використовується та може прийняти наступну команду Вихід.

3. *Пауза* — вихідний параметр лише для читання для керування вихідною віссю вказаного модуля PU для призупинення його Вихід. Якщо *Pause* має значення True, це означає, що вихідний сигнал призупинено, поточна швидкість дорівнює 0, а поточний вихідний сигнал не досяг вказаного цільового вихідного положення. Якщо відновити вихід, вихідне значення буде очищено автоматично. Якщо *пауза* має значення TRUE, наступна команда Вихід не буде прийнята.

8.22.1.2.3. PUPLS

FB/FC	Пояснення	Застосовна модель
FB	PUPLS контролює вказану вісь для Вихід встановленої кількості імпульсів із цільовою швидкістю з поточною позицією як контрольною точкою через модуль PU.	AS516E-B AS532EST-B AS564EST-B

PUPLS_примірник



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Вісь	Вкажіть номер осі для модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
TarPulse	Задайте кількість вихідних імпульсів осі	DINT	0 або позитивне число (0)	Коли *Enable* має значення TRUE
TarSpeed	Укажіть цільову вихідну швидкість осі (одиниця: Гц).	DINT	AS02PU: -200000~200000 (0) AS04PU: -100000~100000 (0)	Коли *Enable* має значення TRUE

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUPLS.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *TarPulse* встановлює кількість вихідних імпульсів. Номер імпульсу є 32-розрядним додатним числом. Коли значення дорівнює 0, це означає, що Вихід виконується завжди, кількість вихідних імпульсів не обмежена і Вихід не зупиняється, доки інструкцію не буде вимкнено. Коли значення менше 0, ПЛК автоматично використовує доповнення до 2, щоб перетворити значення в додатне ціле число як кількість вихідних імпульсів.
4. *TarSpeed* встановлює цільову вихідну швидкість (одиниця: Гц). Цільову швидкість можна змінити в будь-який час після ввімкнення інструкції, і модуль PU автоматично переключиться на щойно встановлену цільову швидкість після Вихід повного імпульсу.

   Примітка: перш ніж змінити цільову швидкість, будь ласка, зверніть увагу на те, чи збігаються змінена швидкість і час сканування ПЛК.

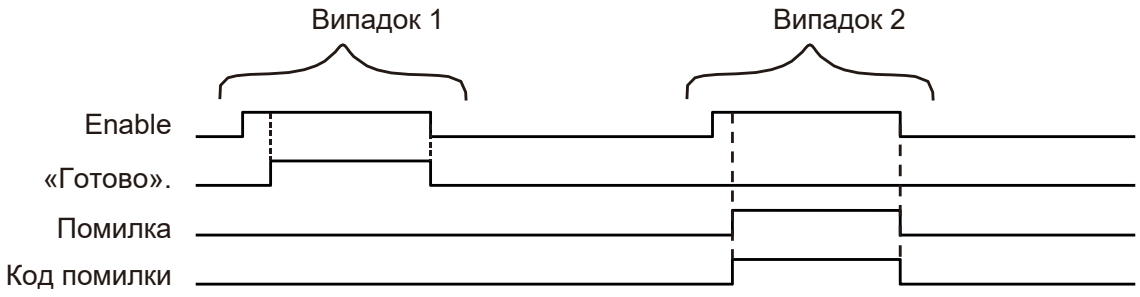
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	ІСТИНА, коли число цільових вихідних імпульсів визначається *TarPulse* досягнуто.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли цільове вихідне число імпульсів визначається *TarPulse* досягнуто.	◆ Коли вказана вісь вимкнена. ◆ Коли *помилка* змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1: *Done змінюється на TRUE, коли* досягається цільове число вихідних імпульсів, визначене *TarPulse після зміни Enable* з FALSE на TRUE. *Значення Done* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

● **функція**

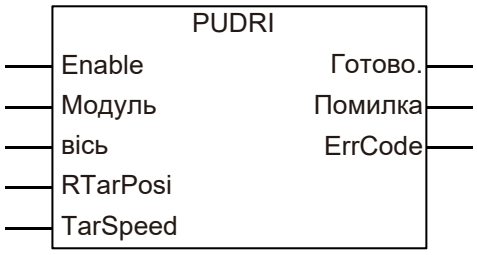
Інструкція PUPLS керує заданою віссю для Вихід заданої кількості імпульсів на цільовій швидкості з поточною позицією як контрольною точкою через модуль PU.

1. Коли *TarSpeed* є додатним числом (>0), це означає, що вихідна точка «позитивного напрямку» УВІМКНЕНА. Якщо TarSpeed є від'ємним числом (<0), це означає, що вихідна точка «негативного напрямку» увімкнена. Коли TarSpeed дорівнює 0, це означає, що Вихід буде призупинено після повного Вихід імпульсу, що виконується.
2. Інструкція PUPLS не підтримує функцію прискорення та уповільнення. Замість цього використовуйте інструкцію PUDRI, якщо вам потрібна функція прискорення та уповільнення.
3. Для зміни швидкості можна використовувати інструкцію PUPLS. Під час виконання інструкції ви можете змінити значення *TarSpeed* , щоб змінити вихідну швидкість.

8.22.1.2.4. PUDRI

FB/FC	Пояснення	Застосовна модель
FB	PUDRI контролює вказану вісь для Вихід заданої кількості імпульсів із цільовою швидкістю, швидкістю прискорення та швидкістю уповільнення з поточним положенням як контрольною точкою через модуль PU.	AS516E-B AS532EST-B AS564EST-B

PU DRI_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Вісь	Вкажіть номер осі для вказаного модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
RTarPosi	Вкажіть кількість вихідних імпульсів для відносного позиціонування.	DINT	0, позитивне чи негативне число (0)	Коли *Enable* має значення TRUE
TarSpeed	Вкажіть цільову вихідну швидкість осі. Одиниця: Гц	DINT	AS02PU: -200000~200000 (0) AS04PU: -100000~100000 (0)	Коли *Enable* має значення TRUE

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUDRI.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення параметрів 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *RTarPosi* встановлює кількість імпульсів для відносного позиціонування, вважаючи поточне положення початковим. Значення може бути 32-розрядним додатним числом, 32-розрядним від'ємним числом або 0. Якщо значення більше за 0, вихід буде здійснюватися в додатному напрямку (а точка Вихід напряму вимкнена). Якщо значення менше 0, вихід буде здійснюватися у негативному напрямку (і точка Вихід напряму буде увімкнена). Коли значення дорівнює 0, результат *Done* змінюється на TRUE.

4. *TarSpeed* встановлює цільову вихідну швидкість (одиниця: Гц). Це може бути 32-розрядне позитивне число. Якщо значення менше 0, інструкція автоматично використовуватиме доповнення до 2, щоб перетворити значення на додатне ціле число. Якщо значення дорівнює 0, інструкція сповістить модуль про перехід у режим паузи. Фактичний вихід сповільнюватиметься зі швидкістю уповільнення, поки вихідна швидкість не дорівнюватиме 0, а потім вихідна *пауза* зміниться на TRUE. Для отримання додаткової інформації зверніться до інструкції PUSTAT.

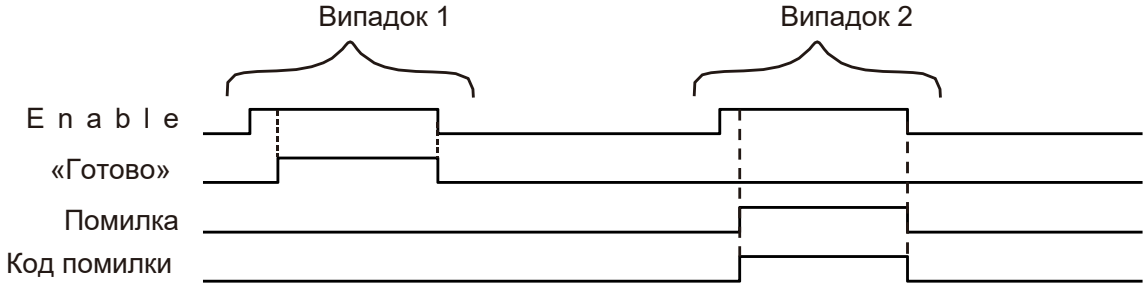
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	досягається кількість імпульсів для відносного позиціонування, визначене *RTarPosi .*	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ досягається кількість імпульсів для відносного позиціонування, визначена *RTarPosi .*	◆ Коли вказана вісь вимкнена. ◆ Коли помилка змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



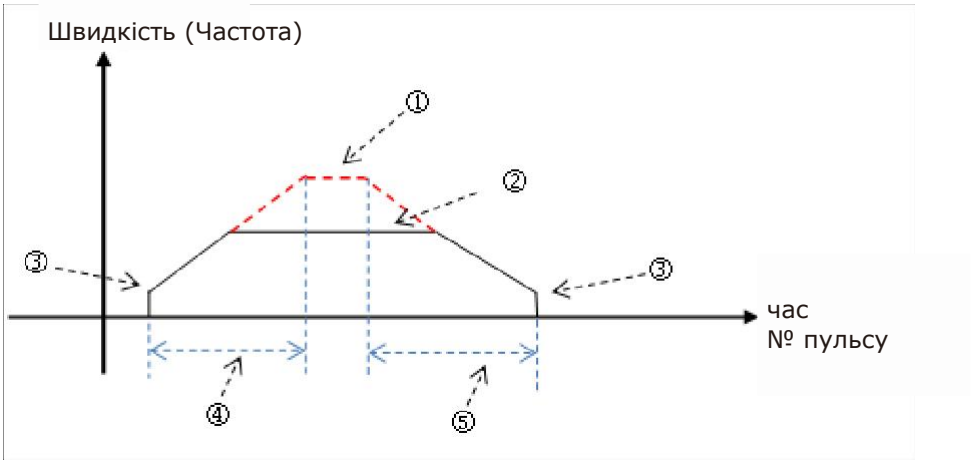
Випадок 1: *«Зроблено»* змінюється на «ІСТИНА», коли позиція, встановлена *RTarPosi* для відносного позиціонування, досягається після того, як *«Enable* » змінюється з «ХИБНІСТЬ» на «ІСТИННА». *Значення Done* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

● **функція**

PUDRI контролює вказану вісь для Вихід заданої кількості імпульсів із цільовою швидкістю, швидкістю прискорення та швидкістю уповільнення з поточним положенням як контрольною точкою через модуль PU.

1. Після початку Вихід цільову швидкість можна будь-коли змінити. Під час фактичної зміни швидкості ПЛК автоматично змінюватиме швидкість на основі встановленої швидкості прискорення та уповільнення в інструкції PUCONF.
2. Дивіться малюнок кривої прискорення та уповільнення інструкції DPUDRI нижче.



① : значення максимальної вихідної швидкості. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

② : цільова швидкість, визначена інструкцією Вихід модуля PU. Цільова вихідна швидкість не повинна перевищувати максимальну вихідну швидкість. Якщо максимальна вихідна швидкість перевищена, максимальна вихідна швидкість вважається вихідною швидкістю.

③ : Початкове/кінцеве значення вихідної швидкості. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметрів через область конфігурації для модуля PU у Конфігурації обладнання.

④ : значення часу прискорення. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

⑤ : Значення часу уповільнення. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

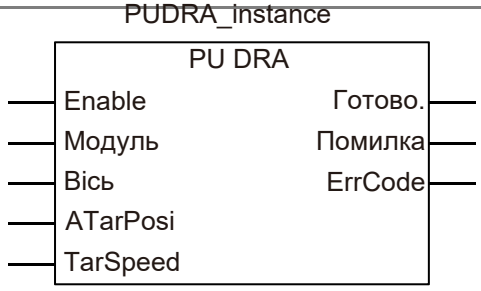
Прискорення та уповільнення, якими керує модуль PU, виконується відповідно до фіксованого нахилу. Отже, фактичний час прискорення та час уповільнення змінюються залежно від вихідної цільової швидкості. Дивіться відповідні формули для розрахунку швидкості прискорення та швидкості уповільнення, як показано нижче.

(Макс. вихідна швидкість - початкова швидкість)/час прискорення,

(Макс. вихідна швидкість - кінцева швидкість)/час уповільнення

8.22.1.2.5. PUDRA

FB/FC	Пояснення	Застосовна модель
FB	Через модуль PU PUDRA контролює вказану вісь для виведення встановленої кількості імпульсів для абсолютного позиціонування на цільовій швидкості, швидкості прискорення та швидкості уповільнення з нульовою точкою як контрольною точкою.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Увімкнути* правда
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Увімкнути* правда
Вісь	Вкажіть номер осі для вказаного модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Увімкнути* правда
ATarPosi	Вкажіть кількість вихідних імпульсів для абсолютного позиціонування.	DINT	0, позитивне чи негативне число (0)	Коли *Увімкнути* правда
TarSpeed	Вкажіть цільову вихідну швидкість вихідної осі. Одиниця: Гц	DINT	AS02PU: -200000~200000 (0) AS04PU: -100000~100000 (0)	Коли *Увімкнути* правда

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUDRA.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *ATarPosi* встановлює кількість вихідних імпульсів для абсолютного позиціонування, вважаючи вихідну точку початковою позицією. Вхідним значенням може бути 32-бітне додатне число, 32-бітне негативне число або 0. Модуль PU автоматично порівнює його з поточною позицією. Якщо результат порівняння більший за 0, вихід буде проведено в позитивному напрямку (а вихідна точка напрямку вимкнена). Якщо результат порівняння менший за 0, вихід буде проведено у негативному напрямку (і точка Вихід напряму ввімкнена). Коли значення дорівнює 0, результат *Done* змінюється на TRUE.

4. *TarSpeed* встановлює цільову вихідну швидкість (одиниця: Гц). Це може бути 32-розрядне позитивне число. Якщо значення менше 0, інструкція автоматично використовуватиме доповнення до 2, щоб перетворити значення на додатне ціле число. Якщо значення дорівнює 0, інструкція сповістить модуль про перехід у режим паузи. Фактичний вихід сповільнюватиметься зі швидкістю уповільнення, поки вихідна швидкість не дорівнюватиме 0, а потім вихідна *пауза* зміниться на TRUE. Для отримання додаткової інформації зверніться до інструкції PUSTAT.

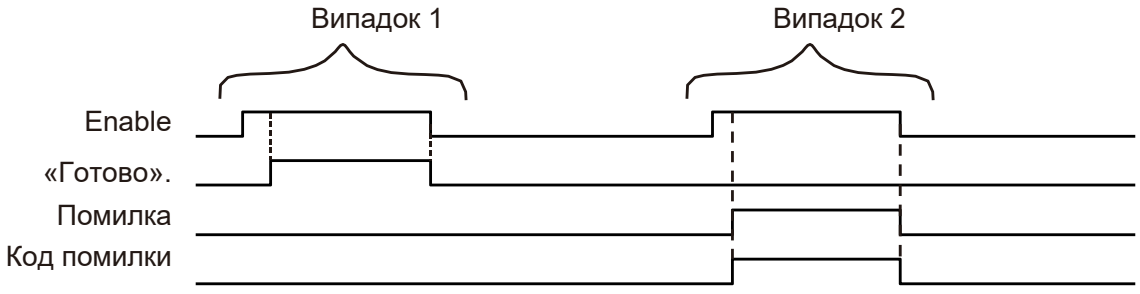
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	досягається кількість імпульсів для абсолютного позиціонування, визначене *ATarPosi .*	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ досягається кількість імпульсів для абсолютного позиціонування, визначена *ATarPosi .*	◆ Коли вказана вісь вимкнена. ◆ Коли помилка змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



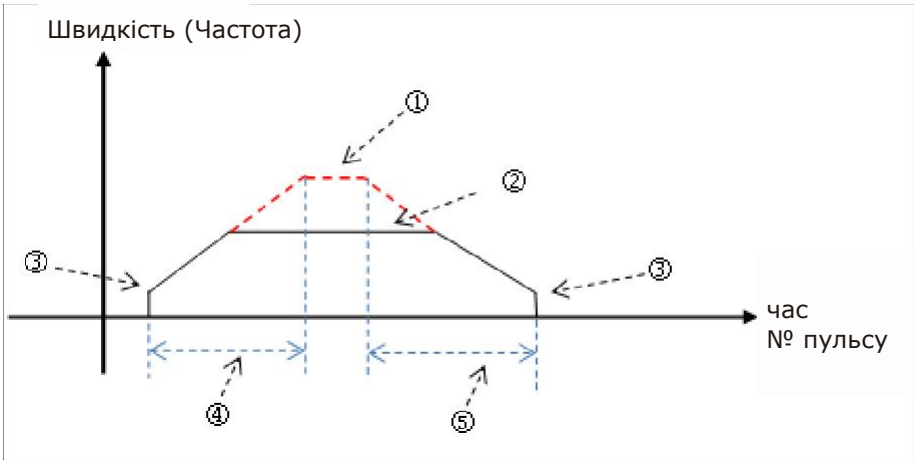
Випадок 1: *Done* змінюється на TRUE, коли кількість цільових вихідних імпульсів, визначена *ATarPosi ,* досягається після зміни *Enable* з FALSE на TRUE. *Значення Done* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE і *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

● **функція**

Інструкція PUDRA використовується для керування вказаною віссю для Вихід заданої кількості імпульсів для абсолютного позиціонування на цільовій швидкості, швидкості прискорення та швидкості уповільнення з нульовою точкою як контрольною точкою через модуль PU.

1. Після початку Вихід цільову швидкість можна будь-коли змінити. Під час фактичної зміни швидкості ПЛК автоматично змінюватиме швидкість на основі встановленої швидкості прискорення та уповільнення в інструкції PUCONF.
2. Дивіться малюнок кривої прискорення та уповільнення інструкції PUDRA нижче.



① : значення максимальної вихідної швидкості. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

② : цільова швидкість, визначена інструкцією Вихід модуля PU. Цільова вихідна швидкість не повинна перевищувати максимальну вихідну швидкість. Якщо максимальна вихідна швидкість перевищена, максимальна вихідна швидкість вважається вихідною швидкістю.

③ : Початкове/кінцеве значення вихідної швидкості. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметрів через область конфігурації для модуля PU у Конфігурації обладнання.

④ : значення часу прискорення. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

⑤ : Значення часу уповільнення. Зверніться до налаштування в інструкції PUCONF для встановлення параметрів. Або встановіть значення параметра в області конфігурації для модуля PU у Конфігурації обладнання.

Прискорення та уповільнення, якими керує модуль PU, виконується відповідно до фіксованого нахилу. Отже, фактичний час прискорення та час уповільнення змінюються залежно від вихідної цільової швидкості. Дивіться відповідні формули для розрахунку швидкості прискорення та швидкості уповільнення, як показано нижче.

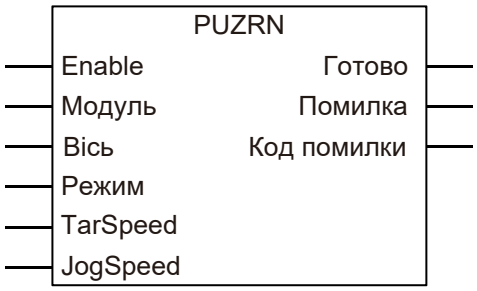
(Макс. вихідна швидкість - початкова швидкість)/час прискорення,

(Макс. вихідна швидкість - кінцева швидкість)/час уповільнення

8.22.1.2.6. PUZRN

FB/FC	Пояснення	Застосовна модель
FB	PUZRN керує вказаною віссю для виконання дії наведення на основі встановленого режиму наведення та швидкості через модуль PU.	AS516E-B AS532EST-B AS564EST-B

PUZRN_екземпляр



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Вісь	Номер осі для вказаного модуля PU.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Режим	Вкажіть режим початкового наведення для осі модуля ПУ	UINT	0~8 або 255 (0)	Коли *Enable* має значення TRUE
TarSpeed	Встановіть максимальну вихідну швидкість для наведення Одиниця: Гц	DINT	Для AS02PU, Режим 0~8 : -100~-200000 і 100~200000 (Значення змінної має бути встановлено); Режим 255 : 0, позитивне чи негативне число (0)	Коли *Enable* має значення TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
			Для AS04PU, Режим 0~8： -100 ~ -100000 і 100~100000 (Значення змінної має бути встановлено); Режим 255： 0, позитивне чи негативне число (0)	
JogSpeed	Встановіть швидкість поштовху для наведення Одиниця: Гц	INT	1~10000 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUZRN.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *Mode* встановлює режим наведення. Пояснення режимів наведено в наступній таблиці.

Режим	Опис функції	Точка введення використовується разом з іншими налаштуваннями	Зауваження
0	Безпосередньо очистити поточну позицію до 0.	Жодного	
1	Вкажіть точку торкання точки DOG як початкову точку; вісь починає рухатися в негативному напрямку, а потім зупиняється після виходу з положення точки DOG.	ПЕС	Використовуйте налаштування в налаштуваннях обладнання
2	Вкажіть точку торкання точки DOG як початкову точку; вісь починає рухатися в позитивному напрямку, а потім зупиняється після виходу з положення точки DOG.	ПЕС	Використовуйте налаштування в налаштуваннях обладнання
3	Після того, як режим Mode=1 завершиться, шукайте встановлену кількість фаз Z.	Вхід фази DOG і Z	Використовується разом із налаштуванням у конфігурації обладнання
4	Після завершення роботи Mode=2 шукайте встановлену кількість фаз Z.	Вхід фази DOG і Z	Використовується разом із налаштуванням у конфігурації обладнання

Режим	Опис функції	Точка введення використовується разом з іншими налаштуваннями	Зауваження
5	Після завершення роботи Mode=1 виведіть вказану кількість вихідних імпульсів.	ПЕС	Використовується разом із налаштуванням у конфігурації обладнання
6	Після завершення роботи Mode=2 виведіть вказану кількість вихідних імпульсів.	ПЕС	Використовується разом із налаштуванням у конфігурації обладнання
7	Після завершення роботи Mode=1 знайдіть кількість фаз Z, а потім виведіть значення зсуву.	Вхід фази DOG і Z	Використовується разом із налаштуванням у конфігурації обладнання
8	Після завершення роботи Mode=2 знайдіть кількість фаз Z, а потім виведіть значення зсуву.	Вхід фази DOG і Z	Використовується разом із налаштуванням у конфігурації обладнання
255	Змініть поточну вихідну позицію для осі.	Жодного	Використовується разом з Параметр *TarSpeed*
Інший	Зарезервований		

Примітка. Зазначена поведінка наведення може не реалізовуватися, якщо точки введення для вибраного режиму не використовуються разом із налаштуванням у конфігурації обладнання.

4. *TarSpeed* встановлює максимальну вихідну швидкість для самонаведення, коли значення режиму становить від 0 до 8. Значенням налаштування може бути 32-розрядне додатне число, 32-розрядне від'ємне число або 0. Плюс (+) і мінус (-) знаки позначають напрямок, у якому починається наведення. Наприклад, знак + означає початок наведення в позитивному напрямку. Якщо режим 255, значення в *TarSpeed* оновлюється як поточна вихідна позиція модуля PU.

5. *JogSpeed* — це швидкість бігу для досягнення вихідного положення. Значення налаштування — це 16-бітне позитивне число в діапазоні 1~10 000 (Гц).

● **Вихідні параметри**

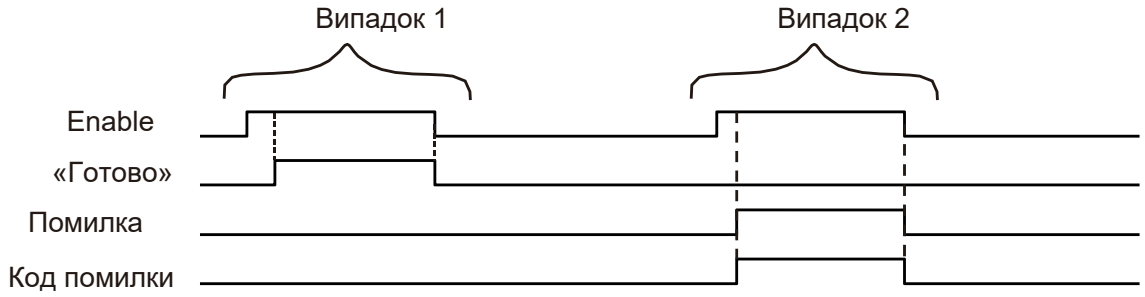
Назва параметра	функція	Тип даних	Допустимий діапазон
Готово	ІСТИНА, коли вихід модуля PU досягає початкового положення.	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

8

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Готово	◆ Коли вихід модуля PU досягає початкового положення.	◆ Коли вказана вісь вимкнена. ◆ Коли помилка змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1: *зроблено* зміни на TRUE, коли початкова позиція досягнута після зміни *Enable* з FALSE на TRUE. *Значення Done* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE і *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

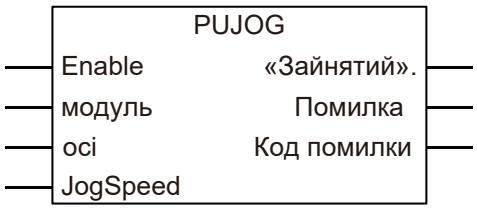
● **функція**

PUZRN керує вказаною віссю для виконання дії наведення на основі встановленого режиму наведення та швидкості через модуль PU.

8.22.1.2.7. PUJOG

FB/FC	Пояснення	Застосовна модель
FB	PUJOG контролює вихідний сигнал заданої осі через модуль PU.	AS516E-B AS532EST-B AS564EST-B

PUJO G_примірник



● **Вхідні**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція вмикається, коли *Enable* змінюється з FALSE на TRUE.	BOOL	ІСТИНА чи НЕПРАВДА (ЛОЖЬ)	Коли *Enable* має значення TRUE.
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість – 32. Якщо вказаний модуль не є модулем PU, вихід *помилки* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
Вісь	Вкажіть номер осі для вказаного модуля ПУ.	USINT	AS02PU: 1~2 (Потрібно встановити значення змінної) AS04PU: 1~4 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
JogSpeed	Вкажіть вихідну швидкість бігу. Одиниця: Гц	DINT	AS02PU: -200000~200000 (0) AS04PU: -100000~100000 (0)	Коли *Enable* має значення TRUE.

Примітка:

1. Контролер серії AS500 з мікропрограмою V1.03 або новішої підтримує інструкцію PUJOG.
2. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення налаштування 1~4 представляють вихідні дані axis1~axis4 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Номери осей і відповідні вихідні точки наведені в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA
AS04PU	Y0.0 / Y0.1	Y0,2 / Y0,3	Y0,4 / Y0,5	Y0,6 / Y0,7

3. *TarSpeed* встановлює вихідну швидкість поштовху. Значення налаштування може бути 32-бітним додатним числом, 32-бітним від'ємним числом або 0. Якщо значення *TarSpeed* є додатним числом (>0), це означає вихід у додатному напрямку (точка Вихід напряму ВИМКНЕНА). Якщо значення *TarSpeed* є від'ємним числом (<0), це означає вихід у від'ємному напрямку (точка Вихід напряму ввімкнена). Коли значення *TarSpeed* дорівнює 0, це означає, що вихід припиняється.

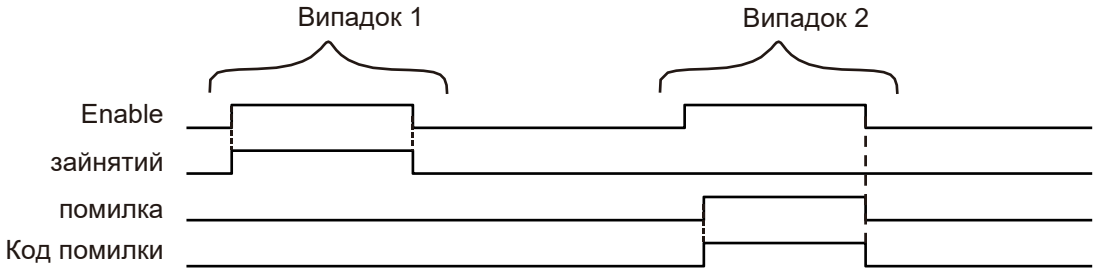
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Зайняте	TRUE під час виконання інструкції	BOOL	ІСТИНА / НЕПРАВДА
Помилка	ІСТИНА, поки є помилка у виконанні інструкції.	BOOL	ІСТИНА / НЕПРАВДА
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Зайняте	◆ Коли *Enable* змінюється з FALSE на TRUE.	◆ Коли вказана вісь вимкнена. ◆ Коли *помилка* змінюється на TRUE
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1: *Busy* змінюється з FALSE на TRUE, коли *Enable* змінюється з FALSE на TRUE. *Готово* змінюється з TRUE на FALSE, коли *Enable* змінюється з TRUE на FALSE.

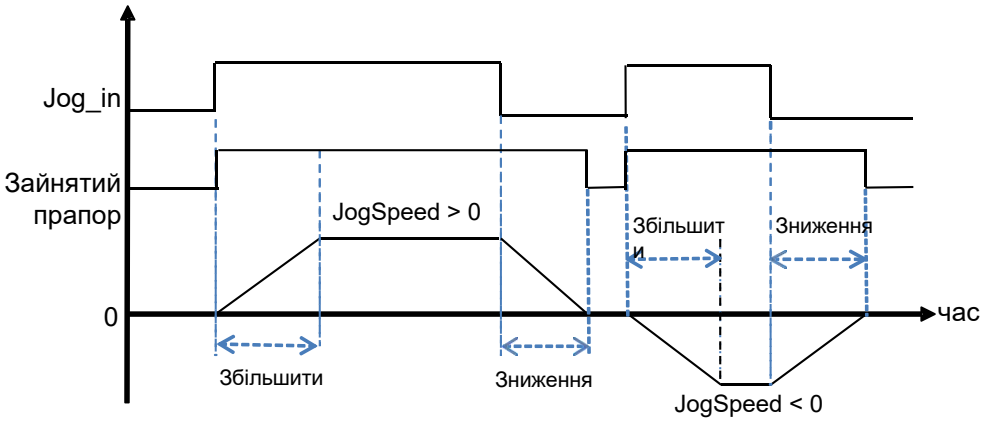
Випадок 2: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrCode* показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

8

● **функція**

Інструкція PUJOG керує вихідним сигналом заданої осі через модуль PU.

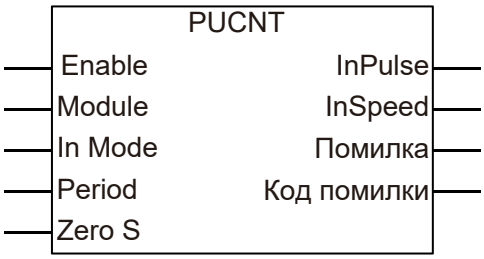
1. Дивіться часову діаграму Вихід інструкції PUJOG, наведену нижче. (**Jog_in** — це перемикач для запуску інструкції та **прапорець Busy** для вихідного сигналу *Busy* .


Jog_in
Зайнятий прапор
0
JogSpeed > 0
Збільшити
Зниження
Збільшити
Зниження
JogSpeed < 0
час

8.22.1.2.8. PUCNT

FB/FC	Пояснення	Застосовна модель
FB	PUCNT функціонує як високошвидкісний лічильник для модуля PU.	AS516E-B AS532EST-B AS564EST-B

PUCNT_примірник



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція вмикається, якщо *ввімкнути* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	Коли *Увімкнути* правда.
Модуль	Вкажіть серійний номер одного з модулів праворуч від ПЛК. Перший номер 1, другий номер 2 і так далі. Будь-який тип модуля праворуч від ПЛК пронумерований. Максимальна кількість становить 32. Якщо вказаний модуль не є модулем PU, вихід *Error* зміниться з FALSE на TRUE.	USINT	1~32 (Потрібно встановити значення змінної)	Коли *Увімкнути* правда.
InMode	Вкажіть режим введення кодера та множення частоти для підрахунку.	USINT	0~7 (0)	Коли *Увімкнути* правда.
Крапка	Вкажіть період часу для фіксації швидкості. Одиниця: мс	UINT	10~1000 (Мінімальне або максимальне значення автоматично встановлюється як значення параметра, якщо діапазон перевищено.)	Коли *Увімкнути* правда.
ZeroS	Очистіть значення лічильника до 0	BOOL	Правда чи брехня (FALSE)	Коли *Увімкнути* правда.

Примітка:

1. Контролер серії AS500 із мікропрограмою версії 1.03 або новішої підтримує інструкцію PUCNT.
2. Лише AS02PU-A з мікропрограмою V1.02.00 або пізнішою підтримує інструкцію PUCNT.
3. *Axis* встановлює номер вихідної осі для вказаного модуля PU. Значення параметрів 1~2 представляють вихід axe1~axis2 зазначеного модуля PU відповідно. Якщо модуль PU не має відповідного номера осі для Вихід, *помилка* зміниться з FALSE на TRUE.

   Осі та відповідні вихідні точки показані в наступній таблиці.

Назва модуля PU	Вісь 1	Вісь 2	Вісь 3	Вісь 4
AS02PU	Y0.0 / Y0.1	Y0,2 / Y0,3	NA	NA

4. *InMode* встановлює режим введення джерела кодера та множення частоти для підрахунку. Дивіться детальне пояснення значень налаштувань у наступній таблиці.

InMode	
Налаштування	Режим введення (модуль PU працюватиме на основі значення налаштування, як показано нижче.)
16#00	Чотирикратний частотний вхід фази A/B Фаза A веде до фази B, вказуючи на підрахунок у позитивному напрямку. Фаза B веде до фази A, вказуючи на підрахунок у негативному напрямку
16#01	Вхід фази A/B одноразової частоти Фаза A веде до фази B, вказуючи на підрахунок у позитивному напрямку. Фаза B веде до фази A, вказуючи на підрахунок у негативному напрямку
16#02	Подвійний частотний вхід фази A/B Фаза A веде до фази B, вказуючи на підрахунок у позитивному напрямку. Фаза B веде до фази A, вказуючи на підрахунок у негативному напрямку
16#05	Імпульсний + спрямований вхід (A+/A-: імпульсний вхід; B+/B-: спрямований вхід) Фаза B ВИМК.: відлік у позитивному напрямку; Фаза B ON: підрахунок у негативному напрямку. Фаза A: підрахунок, викликаний наростаючим фронтом.
16#06	Імпульсний + спрямований вхід (A+/A-: імпульсний вхід; B+/B-: спрямований вхід) Фаза B ON: підрахунок у позитивному напрямку. Фаза B OFF: підрахунок у негативному напрямку. Фаза A: підрахунок, викликаний наростаючим фронтом.
16#07	Однофазний імпульсний вхід (A+/A-: імпульсний вхід) Фаза A: підрахунок, викликаний наростаючим фронтом
інші	Зарезервований

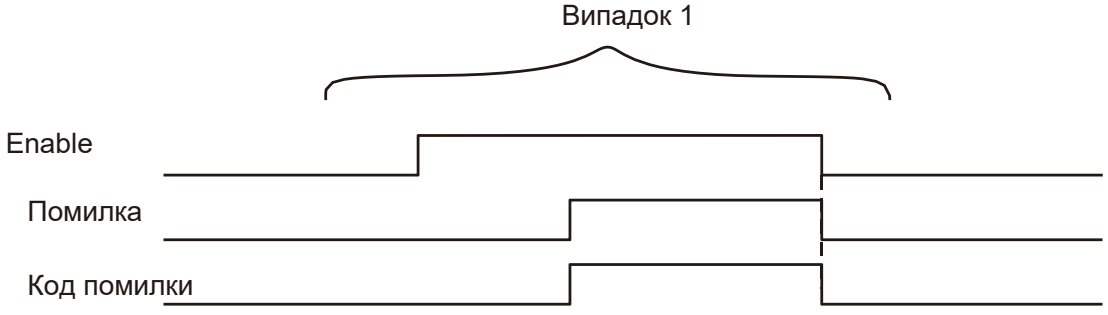
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
InPulse	Кількість імпульсів, які були введені.	DINT	-
InSpeed	Кількість підрахованих пульсів за період часу	DINT	-
ErrCode	Містить код помилки, коли виникає помилка. Дивіться більше в розділі 12.2.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Зайняте	◆ Коли *Enable* змінюється з FALSE на TRUE.	◆ Коли вказана вісь вимкнена. ◆ Коли *помилка* змінюється на TRUE.
Помилка	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли помилка виправлена.

● **Діаграма часу оновлення вихідних даних**



Випадок 1: коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE і *ErrCode*

показує відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *виникає помилка* змінюється на FALSE.

- **функція**

PUCNT функціонує як високошвидкісний лічильник для модуля PU.

1. *InPulse* – це кількість імпульсів, які були введені. Це зафіксоване значення. Якщо його потрібно очистити до 0, просто встановіть *ZeroS* з FALSE на TRUE під час виконання PUCNT.
2. *InSpeed* показує кількість імпульсів, підрахованих за час циклу, визначений *Period* . Користувачі можуть використовувати InSpeed/Period

   формула для перетворення в значення з одиницею Гц. Значення у формулі повинні бути в міжнародних стандартних одиницях.

Пам'ятка

9

Розділ 9 Вступ до параметрів осі

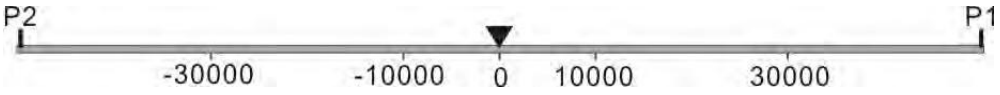
Зміст

9.1 Description of Axis Parameters 9-2

9.1 Опис параметрів осі

Серійний номер	Назва параметра	функція	Тип даних	Значення за замовчуванням
1	Ім'я	Назва осі	STRING	-
"Назва" - це слово-примітка, яке використовується лише для назви сервоприводу без фактичного значення.				
2	Ідентифікатор вузла	CANopen ідентифікатор вузла осі; діапазон: 1-32	USINT	-
«Ідентифікатор вузла» — це адреса станції CANopen сервоприводу.				
3	Тип і одиниця виміру осі	Тип осі: лінійна вісь / вісь обертання Одиниця: одиниця кроку (***UnitsPerRotation***). Наприклад, користувачі можуть вказати мм (міліметр) або ° (градус) як одиницю.	-	Лінійна вісь

Лінійна вісь:

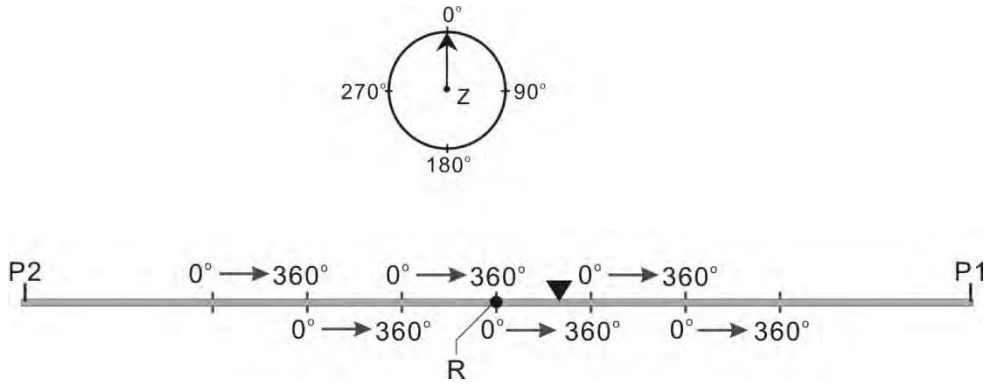


Модель лінійної осі

Примітка:

P1	Позитивний ліміт
P2	Негативний ліміт
▼	Положення сервоприводу

Вісь обертання:



Модель поворотної осі （ “Modulo”: 360 ）

Примітка:

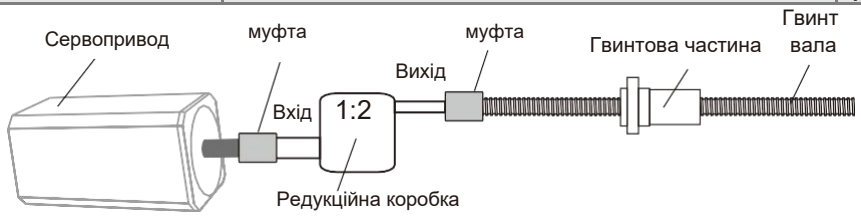
P1	Позитивний ліміт
P2	Негативний ліміт
▼	Положення сервоприводу
P	Головна позиція
3	Вісь серводвигуна

Різниця між лінійною віссю та віссю обертання:

Вісь обертання розглядає модуль як цикл, який є різницею між лінійною віссю та віссю обертання. Положення кінцевого приводу лінійної осі дорівнює 500, а відповідне положення поворотної осі дорівнює 140, що є залишком від 500, поділеного за модулем (360).

Серійний номер	Назва параметра	функція	Тип даних	Значення за замовчуванням
4	Модуло	Цикл, який використовується для рівного розподілу фактичного положення виконавчого механізму терміналу.	LREAL	360
5	Обмеження програмного забезпечення	Вмикає програмне обмеження; Якщо цей пункт не вибрано, максимальне/мінімальне положення осі, яке обмежує програмне забезпечення, недійсне. Якщо цей пункт вибрано, допустиме максимальне/мінімальне положення осі, обмежене програмним забезпеченням.	BOOL	0
6	Максимальна позиція	Максимальне положення осі обмежене програмним забезпеченням	LREAL	-
7	Мінімальна позиція	Мінімальне положення осі обмежене програмним забезпеченням	LREAL	-
8	Максимальна роздільна здатність	Максимальна роздільна здатність для кількості сервоімпульсів	UDINT	1280000
9	Чисельник одиниць	Щоб установити кількість імпульсів, необхідних, коли двигун робить один оберт, налаштувавши параметр і *одиницю знаменника* .	UINT	128
10	Одиниця Знаменник	Щоб установити кількість імпульсів, необхідних, коли двигун робить один оберт, налаштувавши *чисельник одиниць* і параметр.	UINT	1
11	Імпульси/обертання	Скільки імпульсів потрібно, коли серводвигун робить один оберт.	UINT	10000
Чисельник одиниць і *знаменник одиниць* спільно встановлюють електронне передавальне число сервоприводу. Електронне передавальне число використовується для встановлення кількості імпульсів, які сервопривод отримує за одне обертання серводвигуна. Роздільна здатність сервомотора становить 1 280 000 імпульсів/оберт. Припустімо, що значення параметра 11 (імпульси/обертання) дорівнює N. Отже, N*(чисельник одиниць / знаменник одиниць) = 1 280 000.				
12	InputRotation	Цей параметр і *OutputRotation* визначити механічне передавальне число.	UINT	1
13	OutputRotation	*InputRotation* і цей параметр визначають механічне передавальне число.	UINT	1
14	Крок (одиниць на обертання)	Кількість одиниць, на які переміщується виконавчий механізм терміналу, коли вихідний кінець шестерні обертається на одне коло.	UINT	10000
Як показано нижче, ***InputRotation*** =1, ***OutputRotation*** =2, це означає, що вхідний механізм коробки передач обертається на одне коло, а вихідний механізм коробки передач обертається на 2 кола. ***UnitsPerRotation*** представляє відповідну позицію (одиниці), в якій кульковий гвинт переміщується, коли вихідний механізм коробки передач обертається на одне коло. Наприклад, якщо вихідний механізм шестерні обертається на одне коло і кульковий гвинт переміщається на 1 мм, а ***UnitsPerRotation*** встановлено на 1, через інструкцію щодо руху відносного положення кульковий гвинт переміститься на 1 одиницю, тобто кульковий гвинт переміститься на 1 мм; Якщо ***UnitsPerRotation*** встановлено на 1000, кулькова гвинтова переміщення переміститься на 1 одиницю через інструкцію руху MC_MoveRelative, тобто фактично на 1/1000 мм. Одиницею позиції в інструкції керування рухом, G-кодах та електронному кулачку є одиниця.				

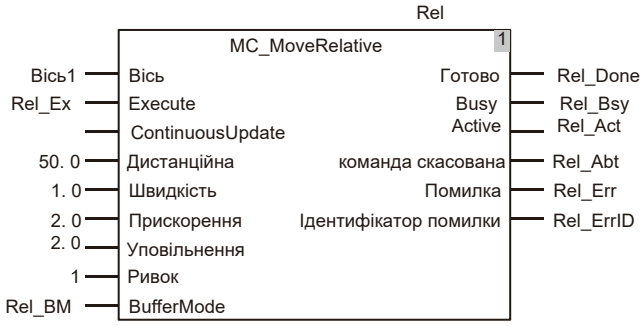
Серійний номер	Назва параметра	функція	Тип даних	Значення за замовчуванням



Як зазначено вище, ***UnitsPerRotation*** встановлено на 1, і кулькова гвинтова переміщатиметься на 50 мм зі швидкістю 1 мм/с, прискоренням 2 мм/с ^{2} і швидкістю зміни прискорення 1 мм/с ^{3} після виконання наступного MC_MoveRelative.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
відн	MC_MoveRelative	
Rel_Ex	BOOL	ПОМИЛКОВИЙ
Рел_БМ	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	СЛОВО	



15	Режим самонаведення	Встановити вихідний режим сервоприводу; діапазон: 1~35. Додаткову інформацію див. у додатку D.	UINT	1
16	Швидкість 1	Швидкість від початку повернення до пошуку домашнього вимикача; Одиниця: об/хв, діапазон налаштування: 1-2000 об/хв	UDINT	20
	Швидкість 2	Швидкість від знаходження домашнього вимикача до досягнення механічного будинку; Одиниця: об/хв, діапазон налаштування: 1-500 об/хв	UDINT	10

10

Розділ 10 Функція керування рухом

Зміст

10.1 EN and ENO 10-2
10.2 Relation among Velocity, Acceleration and Jerk 10-3
10.3 Introduction of BufferMode 10-6
10.4 The State Machine 10-32

Контролер руху серії AS500E розроблено відповідно до протоколу керування рухом CANopen DSP402, і для нього потрібні інструкції керування рухом, визначені як функціональні блоки.

Інструкції з керування рухом для модуля MC базуються на технічних специфікаціях функціональних блоків керування рухом у PLCopen.

Нижче наведено введення того, про що потрібно знати під час використання інструкцій руху.

10.1 EN та ENO

Якщо одна інструкція, яка використовується, має EN та ENO, а EN має значення FALSE (0), функція, визначена інструкцією, не виконуватиметься, а вихідні значення інструкції не оновлюватимуться. Навпаки, функція, визначена інструкцією, буде виконана, і вихідні значення будуть оновлені, якщо EN має значення TRUE (1).

Вихід ENO та вхід EN залишаються узгодженими один з одним. ENO змінюється на TRUE, а EN – на TRUE. ENO змінюється на FALSE, а EN на FALSE.

Для інструкції FB виконання інструкції продовжиться, оскільки її EN змінюється з TRUE на FALSE після виконання. Але вихідні значення інструкції FB не будуть оновлені.

10.2 Зв'язок між швидкістю, прискоренням і ривком

Контролер руху використовує метод прискорення та уповільнення квадратичної кривої. За допомогою цього методу генерована форма хвилі швидкості S-типу може ефективно зменшити механічний удар. Крім того, необхідно вказати принаймні швидкість (v), прискорення (Acc) або уповільнення (Dec) і швидкість зміни прискорення (Jerk), коли використовуються інструкції керування рухом.

Швидкість: вказує на максимальну швидкість руху осі з одиницею/секунда.

Прискорення: вказує на максимальне прискорення руху осі з одиницею/секунду ^{2}.

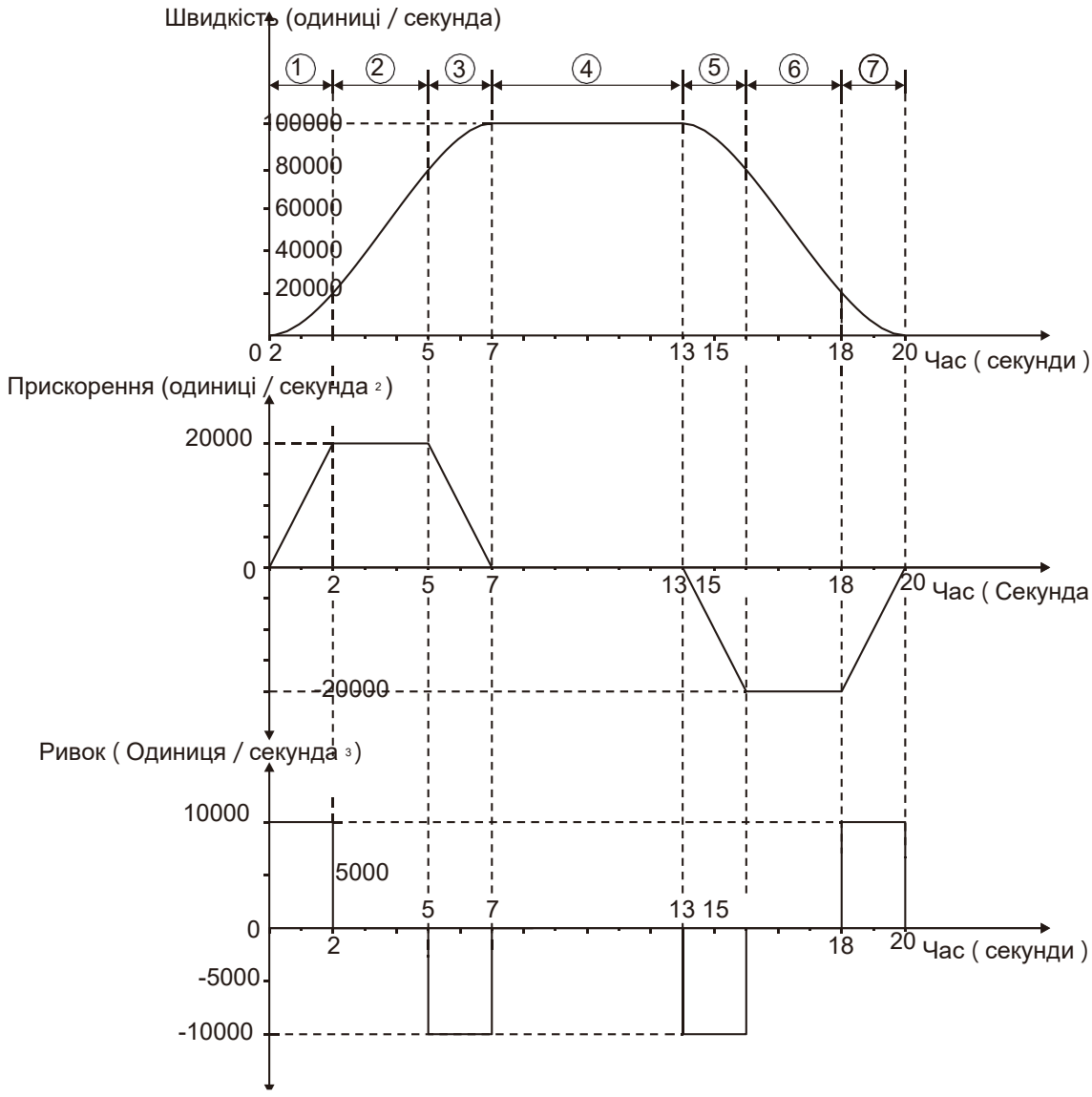
Ривок: вказує на максимальну швидкість зміни прискорення або уповільнення руху осі з одиницею/секунду ^{3}. Значення *Jerk* можна вказати в інструкції, і значення буде використовуватися для осі при прискоренні та уповільненні. Плавність швидкості можна покращити, змінивши значення *Jerk* .

- **Співвідношення між швидкістю, прискоренням (гальмуванням) і ривком:**

$$Acc(Dec) = \frac{dv}{dt}$$

$$Jerk = \frac{dAcc}{dt}$$

Прискорення (уповільнення) — це зміна швидкості за одиницю часу. Швидкість зміни прискорення - це зміна прискорення за одиницю часу. Наприклад, одна інструкція MC_MoveRelative використовується для виражати зв'язок між трьома елементами. Відстань 1300000 одиниць; 100000 швидкість одиниць/секунду; прискорення становить 20000 одиниць/секунду ^{2}, а ривок становить є 10000 одиниць/секунду ^{3}. Дивіться наступну діаграму, щоб дізнатися про зв’язок між цими елементами.



● **Зв'язки між швидкістю, прискоренням і ривком пояснюються в наступній таблиці.**

Етап №	Час (секунда)	Ривок (одиниці / секунда ^{3})	Прискорення / уповільнення (Одиниця / секунда ^{2})	Швидкість (одиниці / секунда)	Тип руху
1	0~2	10000 одиниць/секунду ^{3}	Прискорення збільшено до 20000 одиниць/с ^{2}	Збільшення	Прискорений рух зі зростаючим прискоренням
2	2~5	0	Прискорення залишається на рівні 20000 одиниць/с ^{2}	Збільшення	Прискорений рух з постійним прискоренням
3	5~7	-10000 одиниць/секунду ^{3}	Прискорення зменшено до 0.	Збільшується до 100000 одиниця/секунду	Прискорений рух зі спадаючим прискоренням

Етап №	Час (секунда)	Ривок (одиниці / секунда ^{3})	Прискорення / уповільнення (Одиниця/секунда ^{2})	Швидкість (одиниці / секунда)	Тип руху
4	7~13	0	Прискорення було зменшено до 0 одиниць/секунду ^{2,a} на цьому етапі було 0 одиниць/секунду ^{2}.	100000 одиниця/секунду	Рух з постійною швидкістю
5	13~15	-10000 одиниць/секунду ^{3}	Уповільнення збільшено до 20000 одиниць/секунду ^{2}.	Зменшується	Уповільнений рух зі збільшенням уповільнення
6	15~18	0	Уповільнення було збільшено до 20000 одиниць/секунду ^{2,a} на цьому етапі було 20000 одиниць/секунду ^{2}.	Зменшується	Уповільнений рух з постійним уповільненням
7	18~20	10000 одиниць/секунду ^{3}	Уповільнення зменшено до 0.	Зменшується до 0	Уповільнений рух із спадаючим уповільненням

10.3 Введення BufferMode

Для тієї самої осі можна запустити іншу інструкцію руху, коли одна інструкція руху керує рухом осі. Є 6 режимів буфера для вибору для перемикання з однієї інструкції руху, що виконується, на іншу інструкцію руху. Буферний режим можна вибрати за допомогою параметра *BufferMode* інструкції буферизованого руху.
Терміни *BufferMode* пояснюються нижче.

1. Поточна інструкція: інструкція руху, яка в даний момент керує віссю.
2. Буферизована інструкція: інструкція, яка очікує на виконання.
3. Швидкість проходження: швидкість, з якою вісь рухається в момент, коли інструкція, що виконується, перемикається на інструкцію з буфером.
4. Цільова швидкість: Параметр *швидкості* інструкції
5. Цільове положення: параметр *«Позиція»* або *«Дистанція»* інструкцій, пов'язаних із положенням

- **Шість режимів буфера для вибору**

Буферний режим	опис
0: mcAborting **(Переривання)**	Інструкція, що виконується в даний момент, негайно переривається.
1: mcBuffered **(Буферизовано)**	Буферизована інструкція просто починає керувати віссю після завершення виконання поточної інструкції.
2: mcBlendingLow **(Змішувати з низьким)**	Буферизована інструкція починає керувати віссю після досягнення цільової позиції поточної інструкції. Швидкість проходження є нижчою з цільових швидкостей поточної інструкції та буферизованої інструкції.
3: mcBlendingPrevious **(Змішати з попереднім)**	Буферизована інструкція починає керувати віссю після досягнення цільової позиції поточної інструкції. Швидкість проходження — це цільова швидкість поточної інструкції.
4: mcBlendingNext **(Змішати з наступним)**	Буферизована інструкція починає керувати віссю після досягнення цільової позиції поточної інструкції. Швидкість проходження — це цільова швидкість буферизованої інструкції.
5: mcBlendingHigh **(Змішувати з високим)**	Буферизована інструкція починає керувати віссю після досягнення цільової позиції поточної інструкції. Швидкість проходження є вищою з цільових швидкостей поточної інструкції та буферизованої інструкції.

Примітки:

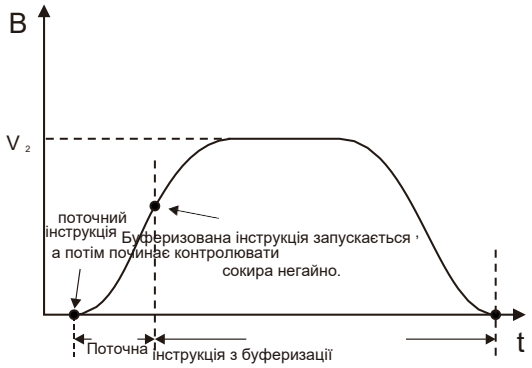
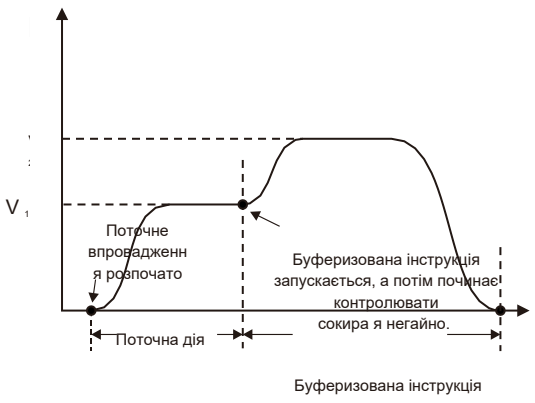
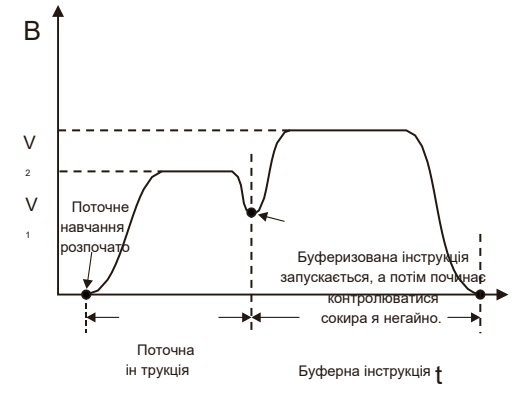
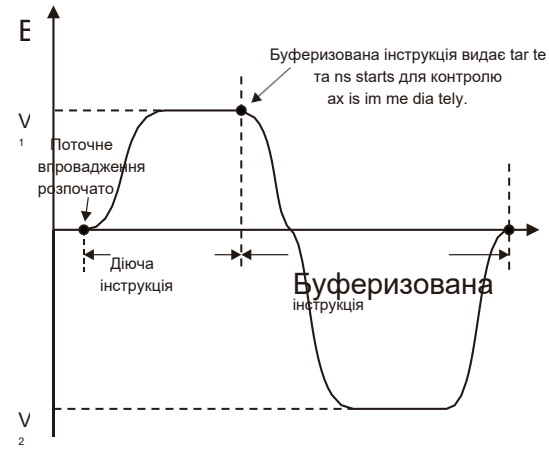
1. Одна вісь підтримує лише один режим буфера. Якщо для однієї осі буде виконано кілька режимів буфера, виникне помилка.
   Наприклад, параметри *BufferMode* інструкцій 2 і інструкцій 3 не є mcAborting. Інструкція 2 (буферизована інструкція) буде переключена з інструкції 1 (поточна інструкція). Інструкція 3 повідомить про помилку, якщо інструкція 3 перемикається з інструкції 2, коли виконання інструкції 1 не завершено. Якщо параметр *BufferMode* інструкції 3 має значення mcAborting, інструкцію 1 і інструкцію 2 буде негайно перервано, а інструкцію 3 буде виконано негайно.
2. Коли інструкція MC_MoveSuperimposed керує лише віссю, буферизована інструкція, за винятком MC_MoveAdditive, виконується, а інструкція MC_MoveSuperimposed переривається незалежно від значення параметра *BufferMode* .
   У той час як поточна інструкція та MC_MoveSuperimposed або MC_HaltSuperimposed спільно керують віссю, а потім виконується інша інструкція руху, усі інструкції, які виконувалися раніше, будуть перервані, якщо *BufferMode* =mcAborting; якщо *BufferMode* = mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext і mcBlendingHigh, поточна інструкція та буферизація

інструкція буде змішана відповідно до значення параметра *BufferMode* без жодного впливу на виконання MC_MoveSuperimposed або MC_HaltSuperimposed.

● **Приклад: використання двох інструкцій MC_MoveRelative для пояснення.**

Максимальна швидкість першої інструкції MC_MoveRelative становить V_1, а відстань – S_1. Максимальна швидкість другої інструкції MC_MoveRelative становить V_2, а відстань – S_2. Змінюючи значення *BufferMode* другої інструкції MC_MoveRelative, ви можете отримати різні процеси змішування двох інструкцій. Подробиці див. нижче.

■ **Переривання: Buffermode=mcAborting. Дивіться приклади чотирьох випадків, як показано нижче.**

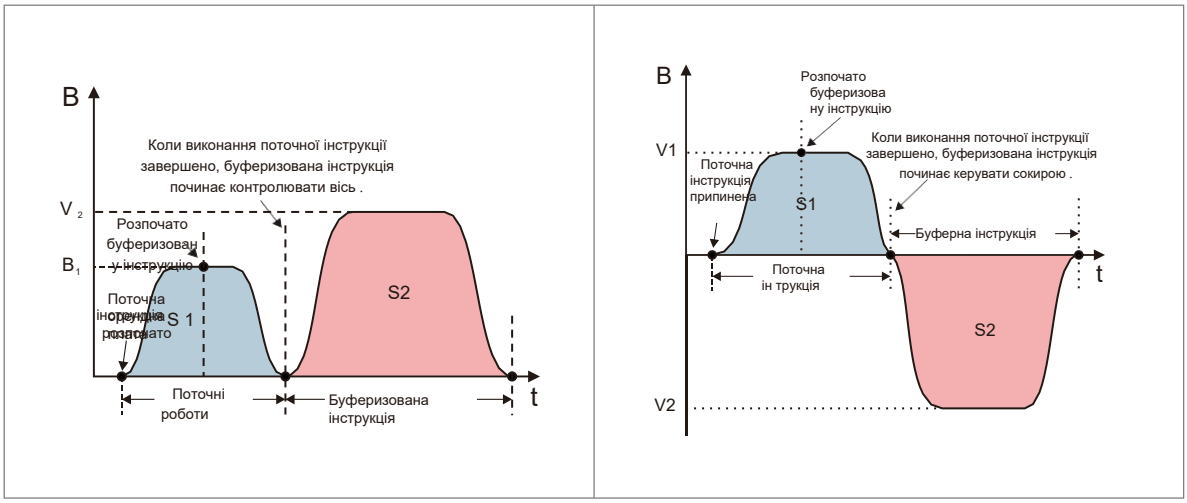
1. Поточна інструкція переривається, поки керована вісь прискорюється.	**2. Поточна інструкція переривається, поки керована вісь рухається постійно швидкість.**
	
3. Поточна інструкція переривається, поки керована вісь сповільнюється.	**4. Напрямок швидкості буфера інструкція протилежна поточній в конструкція.**
	

■

■ **Буферизовано: Buffermode=** mcBuffered **. Дивіться два**

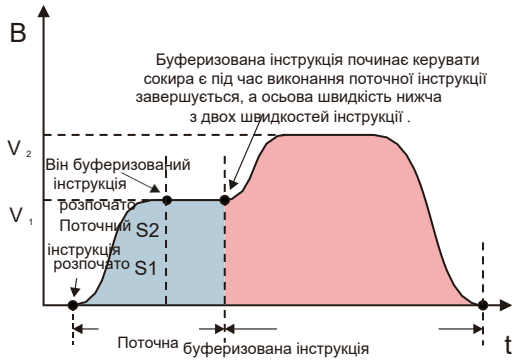
1. Коли напрямок буферизується інструкція така ж, як у поточна інструкція	**2. Коли напрямок буферизується інструкція протилежна інструкції поточна інструкція.**

8

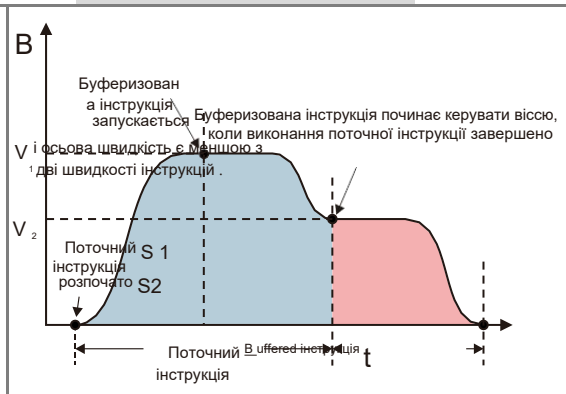


■ **Змішування з низькою швидкістю: Buffermode=** mcBlendingLow **. Дивіться три**

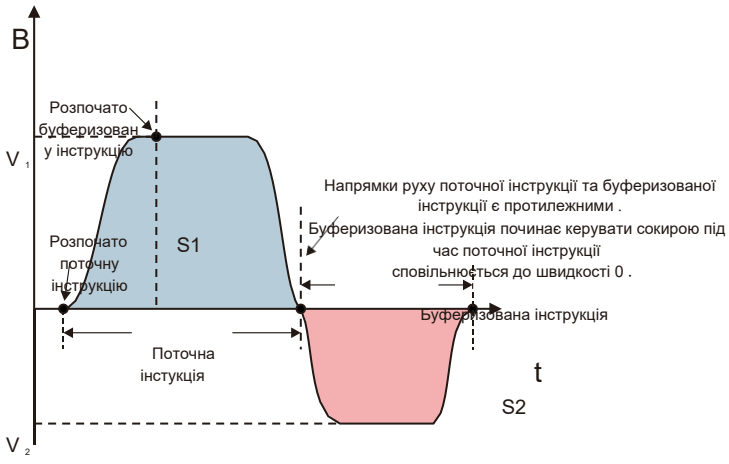
1. При швидкості течії інструкція менше, ніж у буферизована інструкція.



2. При швидкості течії інструкція більша, ніж у буферизована інструкція.

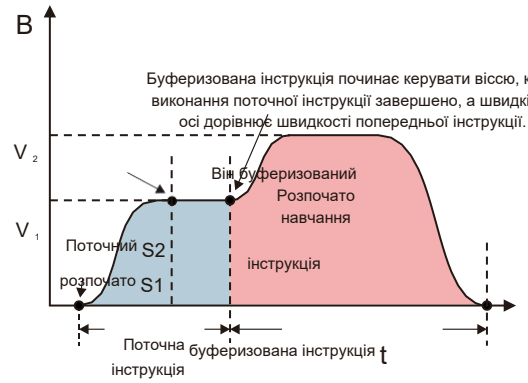


3. Коли напрямок швидкості поточної інструкції протилежний буферизованій інструкція.

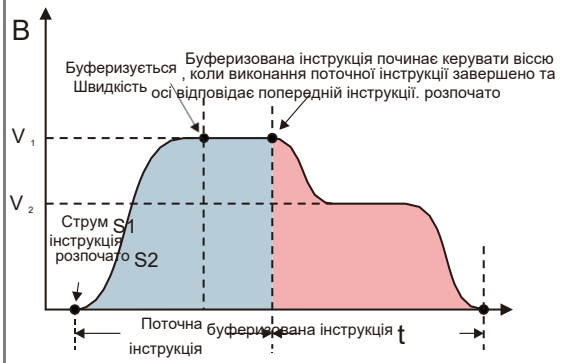


■ **Змішування з попередньою швидкістю: Buffermode=** mcBlendingPrevious **. Дивіться три випадки, як показано нижче.**

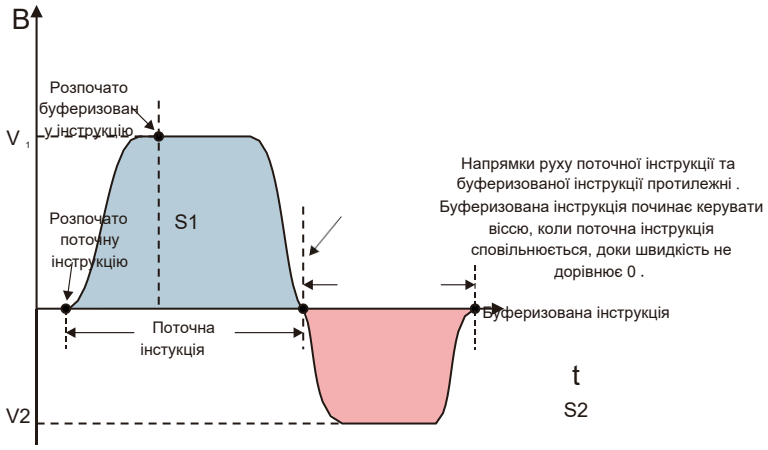
1. При цільовій швидкості течії інструкція менше, ніж у буферизована інструкція.



2. При цільовій швидкості течії інструкція більша, ніж у буферизована інструкція.

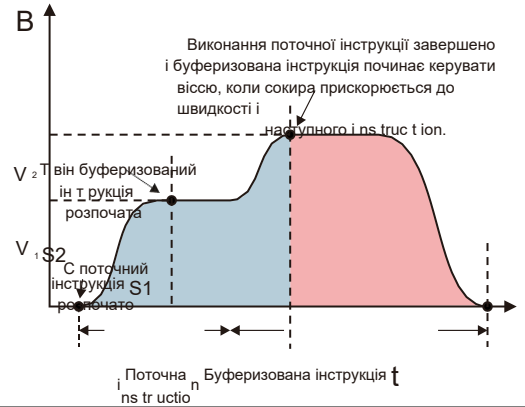


3. Коли напрямок швидкості поточної інструкції протилежний буферизованій інструкція.

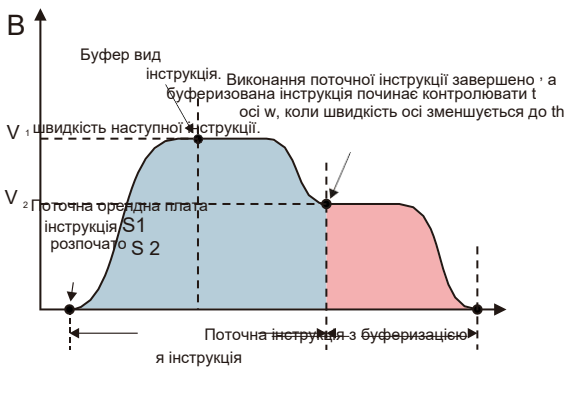


■ **Змішування з наступною швидкістю: Buffermode=** mcBlendingNext **. Дивіться три випадки, як показано нижче.**

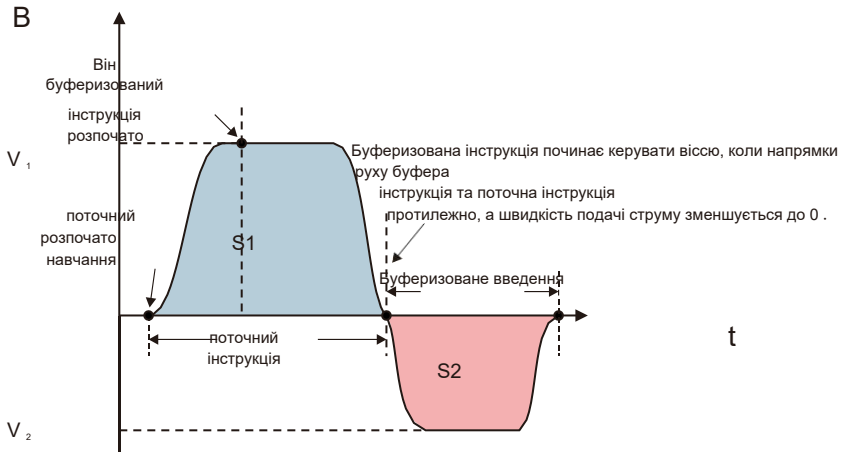
1. При цільовій швидкості течії інструкція менше, ніж у буферизована інструкція



2. При цільовій швидкості течії інструкція більша, ніж у буферизована інструкція

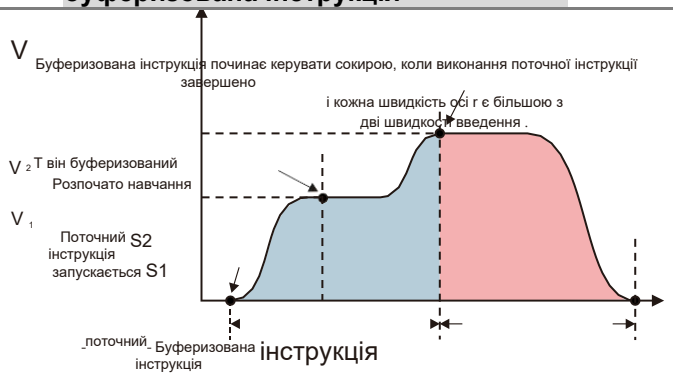


3. Коли напрямок швидкості поточної інструкції протилежний буферизованій інструкція

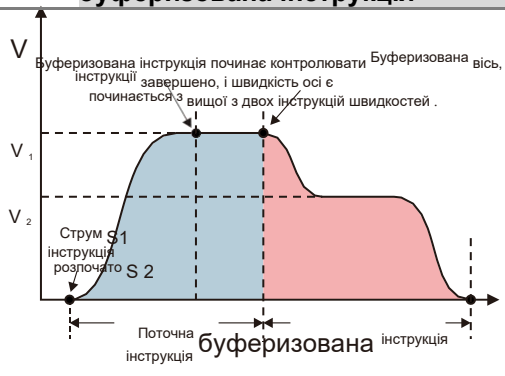


■ **Змішування з високою швидкістю: Buffermode=** mcBlendingHigh **. Дивіться три випадки, як показано нижче.**

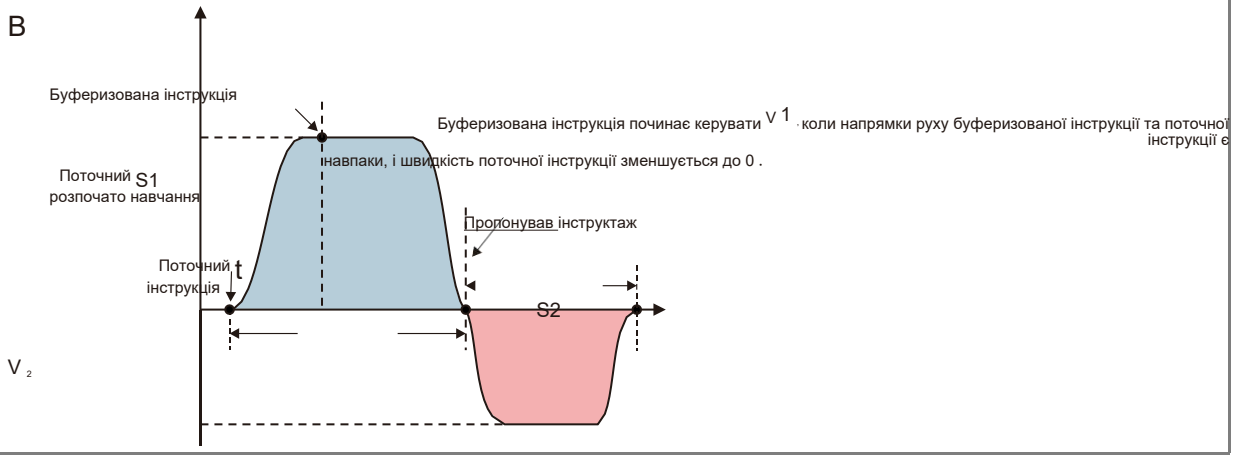
1. При цільовій швидкості течії інструкція менше, ніж у буферизована інструкція



2. При цільовій швидкості течії інструкція більша, ніж у буферизована інструкція



3. Коли напрямок швидкості поточної інструкції протилежний буферизованій інструкція



● **Режими буфера, які підтримують різні інструкції**

Буферний режим поточної інструкції та буферизованої інструкції встановлюється шляхом зміни значення параметра *BufferMode* . Значення BufferMode буферизованої інструкції вибирається відповідно до буферного режиму, який підтримує поточна інструкція, а параметр B *ufferMode* поточної інструкції недійсний.

Наприклад:

BufferMode MC_MoveRelative підтримує mcAborting, mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext і mcBlendingHigh . BufferMode MC_MoveVelocity підтримує mcAborting і mcBuffered .

Випадок 1 : якщо MC_MoveRelative є поточною інструкцією, а MC_MoveVelocity є буферизованою інструкцією. Параметр *BufferMode* MC_MoveVelocity може вибрати один із mcAborting, mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext і mcBlendingHigh.

Випадок 2 : якщо MC_MoveVelocity є поточною інструкцією, а MC_MoveRelative є буферизованою інструкцією. The

BufferMode MC_MoveRelative може вибрати один із mcAborting і mcBuffered.

Режим буферизації буферизованої інструкції можна вибрати відповідно до поточної інструкції, як зазначено нижче.

Поточний інструктаж	Вибір значення *BufferMode* буферизованої інструкції
MC_MoveAbsolute	【 mcAborting, mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext, mcBlendingHigh 】 * 1
MC_MoveRelative	【 mcAborting, mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext, mcBlendingHigh 】 * 1
MC_MoveAdditive	【 mcAborting, mcBuffered, mcBlendingLow, mcBlendingPrevious, mcBlendingNext, mcBlendingHigh 】 * 1
MC_MoveSuperimposed	mcAborting
MC_HaltSuperimposed	mcAborting
MC_MoveVelocity	mcAborting, mcBuffered
MC_Home	Тільки інструкція MC_Stop може перервати інструкцію MC_Home.
MC_Halt	mcAborting, mcBuffered
MC_ GearIn	mcAborting, mcBuffered
MC_ GearOut	mcAborting, mcBuffered
MC_CombineAxes	mcAborting, mcBuffered
MC_ CamIn	mcAborting, mcBuffered
MC_ CamOut	mcAborting, mcBuffered

* 1 : Параметр *BufferMode* буферизованих інструкцій MC_GearIn, MC_CamIn і MC_CombineAxes може вибирати лише mcAborting і mcBuffered.

Чи завершено поточне виконання інструкції, залежить від вихідного параметра завершення інструкції. Оскільки вихідний параметр завершення має значення TRUE, це вказує на те, що виконання інструкції завершено та починається виконання інструкції з буфером.

Перегляньте вихідні параметри завершення інструкцій у наступній таблиці, щоб судити про стан виконання інструкцій у буферному режимі.

Назва інструкції	Це буферизована інструкція? (Так або ні)	Чи може за ним слідувати буферизована інструкція? (Так або ні)	Вихідний параметр завершення інструкції
MC_Home	Немає	Немає	Готово
MC_Stop	Немає	Немає	Готово
MC_Halt	Так	Так	Готово
MC_MoveAbsolute	Так	Так	Готово
MC_MoveRelative	Так	Так	Готово
MC_MoveAdditive	Так	Так	Готово
MC_MoveSuperimposed	Немає	Немає	——
MC_HaltSuperimposed	Немає	Немає	——
MC_MoveVelocity	Так	Так	InVelocity
MC_CamIn	Так	Так	EndOfProfile
MC_CamOut	Немає	Так	Готово
MC_GearIn	Так	Так	InGear
MC_GearOut	Немає	Так	Готово
MC_CombineAxes	Так	Так	InSync

●Приклади буферних режимів.

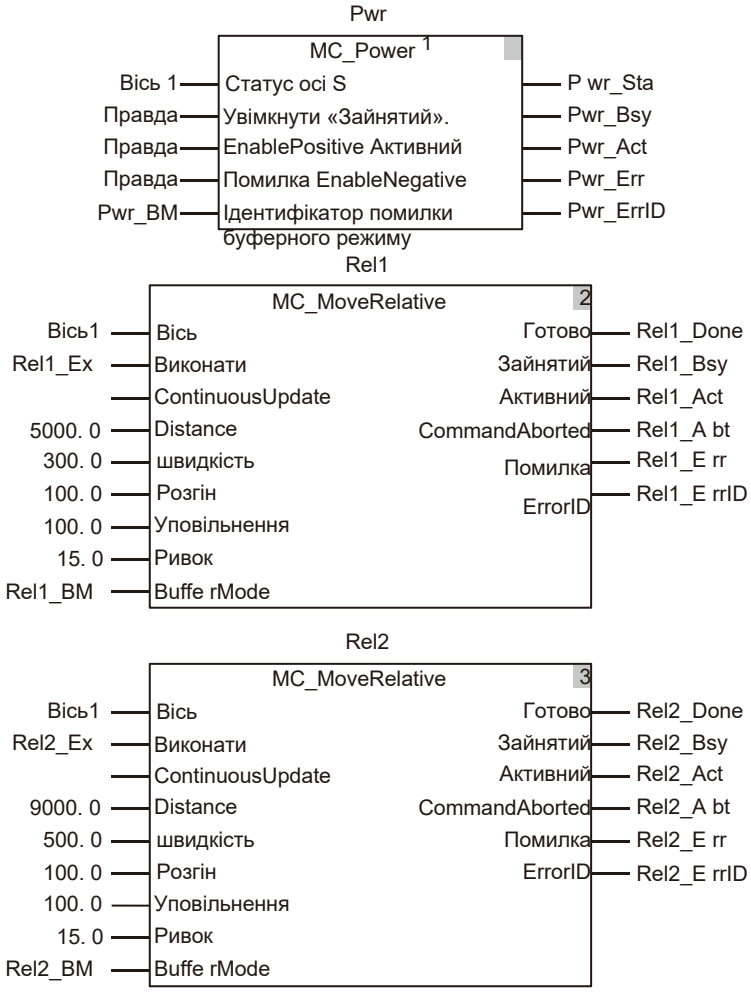
Приклад 1

У наступному прикладі пояснюється шість режимів буфера для перемикання з виконання однієї інструкції MC_MoveRelative на іншу інструкцію MC_MoveRelative.

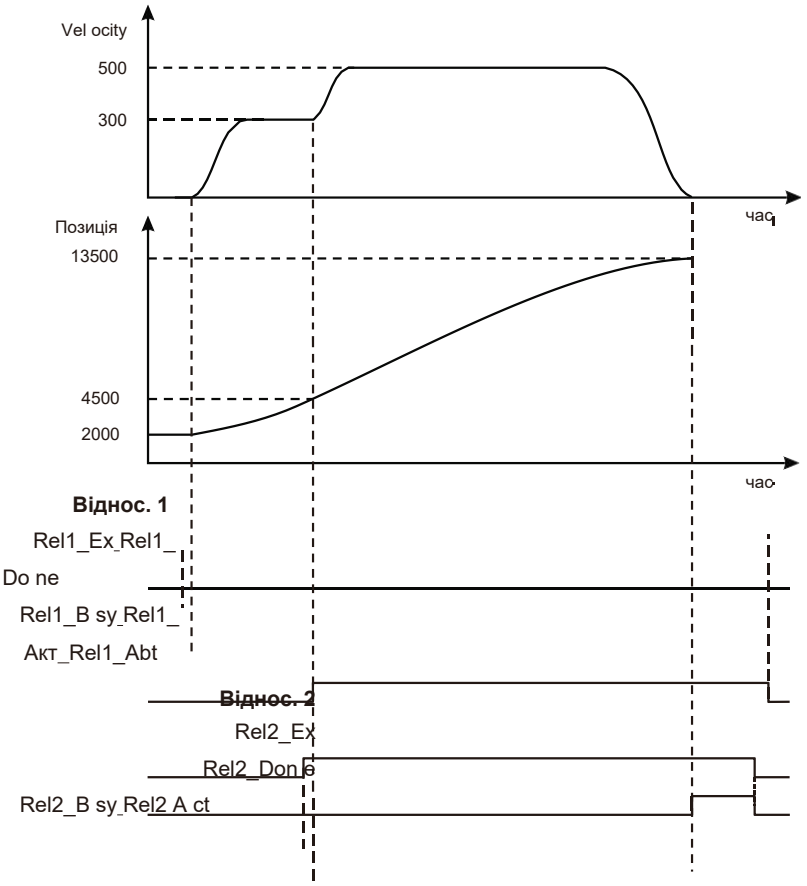
Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Вісь1	USINT	1
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	СЛОВО	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	ПОМИЛКОВИЙ
Rel1_BM	MC_Buffer_Mode	0
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	СЛОВО	

Ім'я змінної	Тип даних	Початкове значення
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	ПОМИЛКОВИЙ
Rel2_BM	MC_Buffer_Mode	
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	СЛОВО	



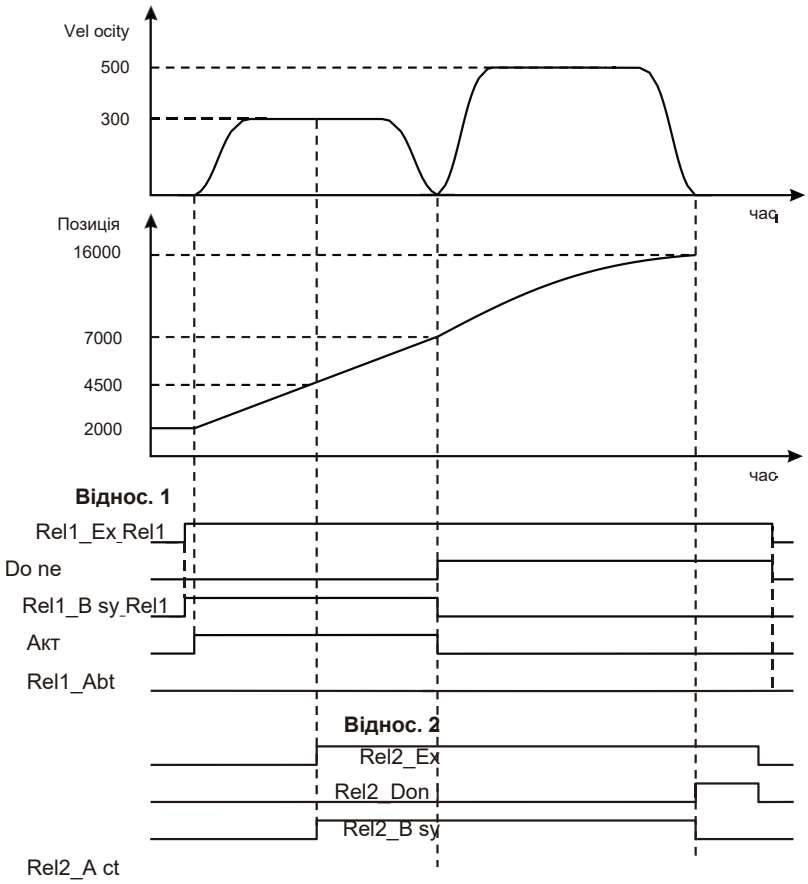
■ **Rel2_BM=mcAborting**



- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута, Rel2_Ex змінюється з FALSE на TRUE, а Rel2_Bsy змінюється на TRUE. Через один період Rel1_Abt і Rel2_Act змінюються на TRUE, а Rel1_Bsy і Rel1_Act змінюються на FALSE. Тим часом перша інструкція MC_MoveRelative переривається, і починається виконання другої інструкції MC_MoveRelative. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а Rel2_Bsy та Rel2_Act тим часом змінюються на FALSE.
- ❖ Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

8

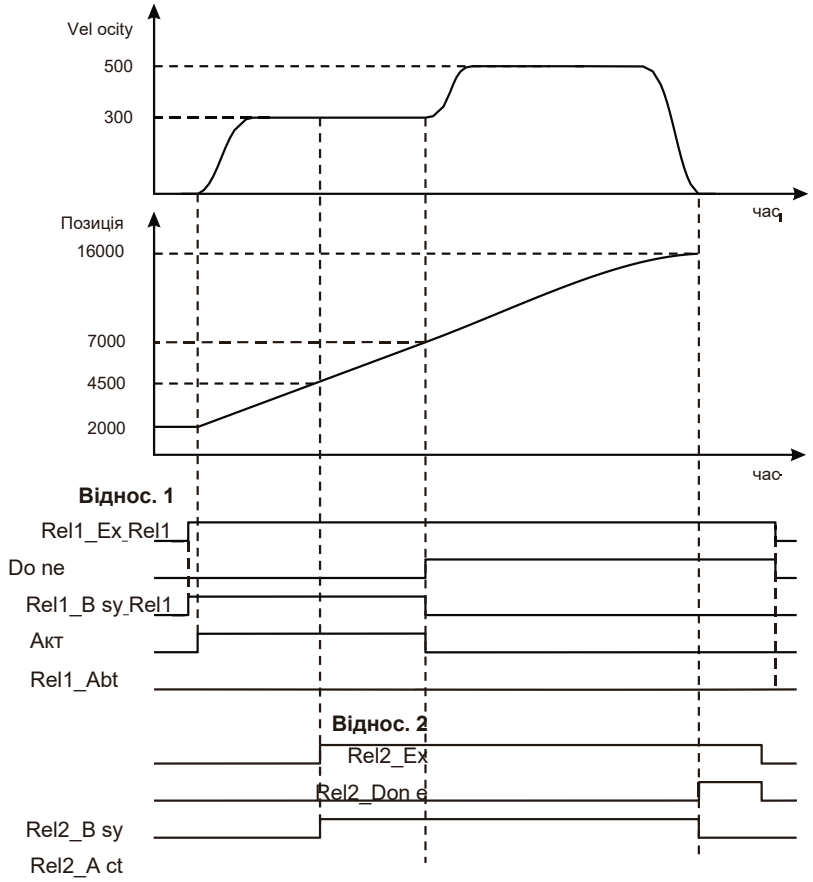
■ **Rel2_BM = mcMcBuffered**



- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута і Rel2_Ex змінюється з FALSE на TRUE, Rel2_Bsy змінюється на TRUE, Rel1_Bsy і Rel1_Act залишаються TRUE, а виконання першої інструкції MC_MoveRelative продовжується. Після досягнення цільової позиції Rel1_Done змінюється на TRUE, Rel1_Bsy і Rel1_Act змінюються на FALSE. Rel2_Act змінюється на TRUE, і виконання другої інструкції MC_MoveRelative починається негайно. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а тим часом Rel2_Bsy та Rel2_Act змінюються на FALSE.
- ❖ Коли Rel1_Ex змінюється з TRUE на FALSE, Rel1_Done змінюється на FALSE. Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

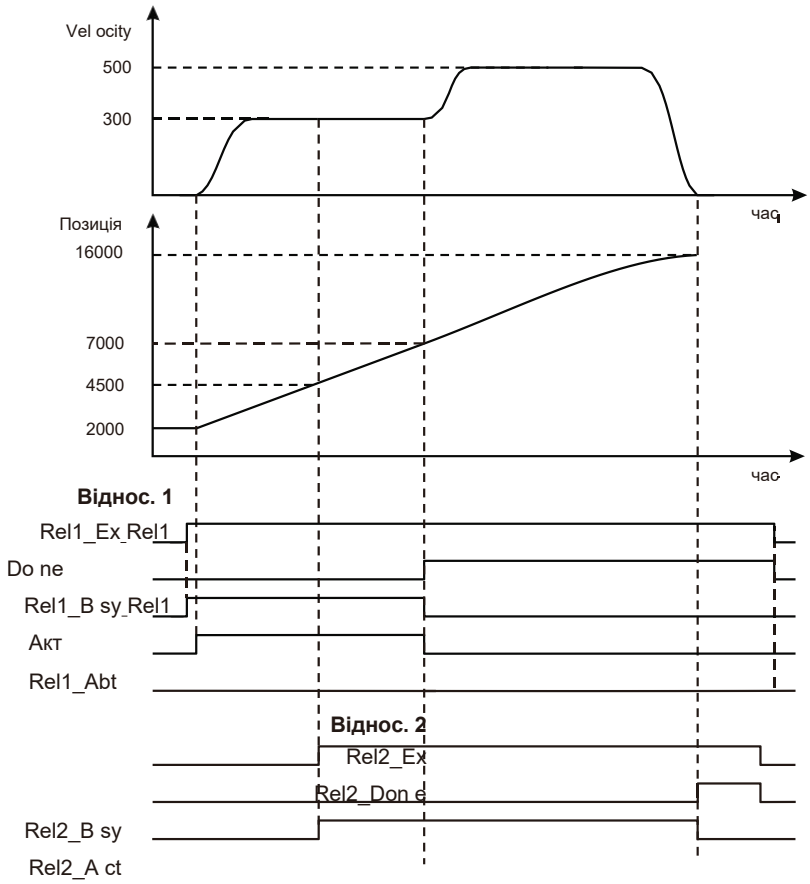
8

■ **Rel2_BM = mcBlendingLow**



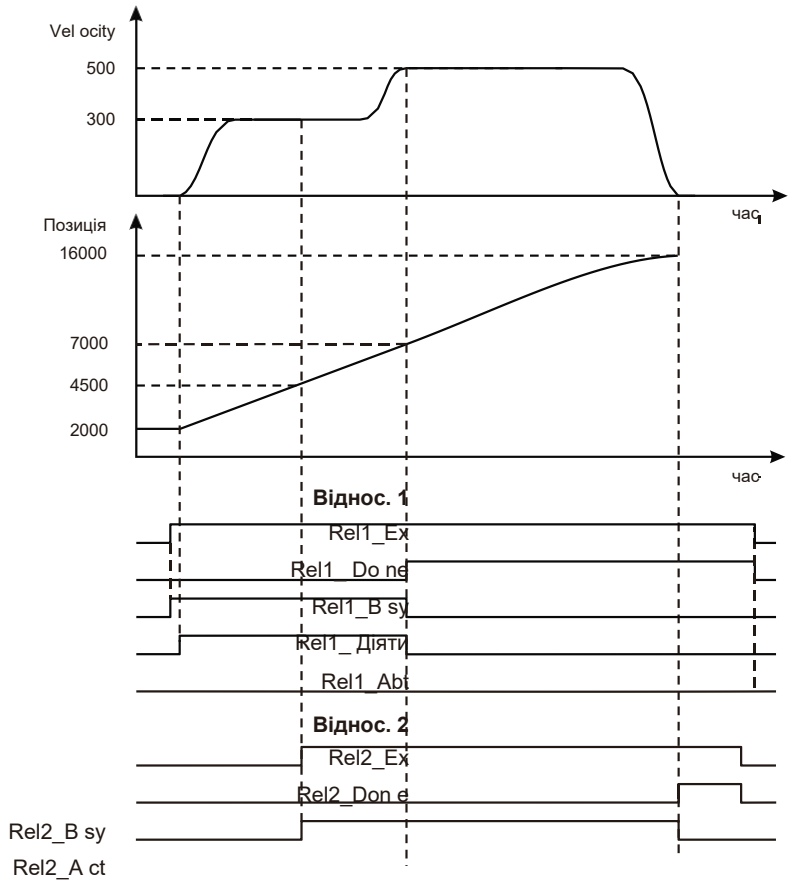
- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута і Rel2_Ex змінюється з FALSE на TRUE, Rel2_Bsy змінюється на TRUE, Rel1_Bsy і Rel1_Act залишаються TRUE, а виконання першої інструкції MC_MoveRelative продовжується. Після досягнення цільової позиції Rel1_Done змінюється на TRUE. На даний момент швидкість становить 300 одиниць/секунду, що є меншою з цільових швидкостей поточної інструкції та буферизованої інструкції, Rel1_Bsy та Rel1_Act змінюються на FALSE, Rel2_Act змінюється на TRUE, і виконання другої інструкції MC_MoveRelative починається негайно. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а Rel2_Bsy та Rel2_Act тим часом змінюються на FALSE.
- ❖ Коли Rel1_Ex змінюється з TRUE на FALSE, Rel1_Done змінюється на FALSE. Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

■ **Rel2_BM =mcBlending _Попередній**



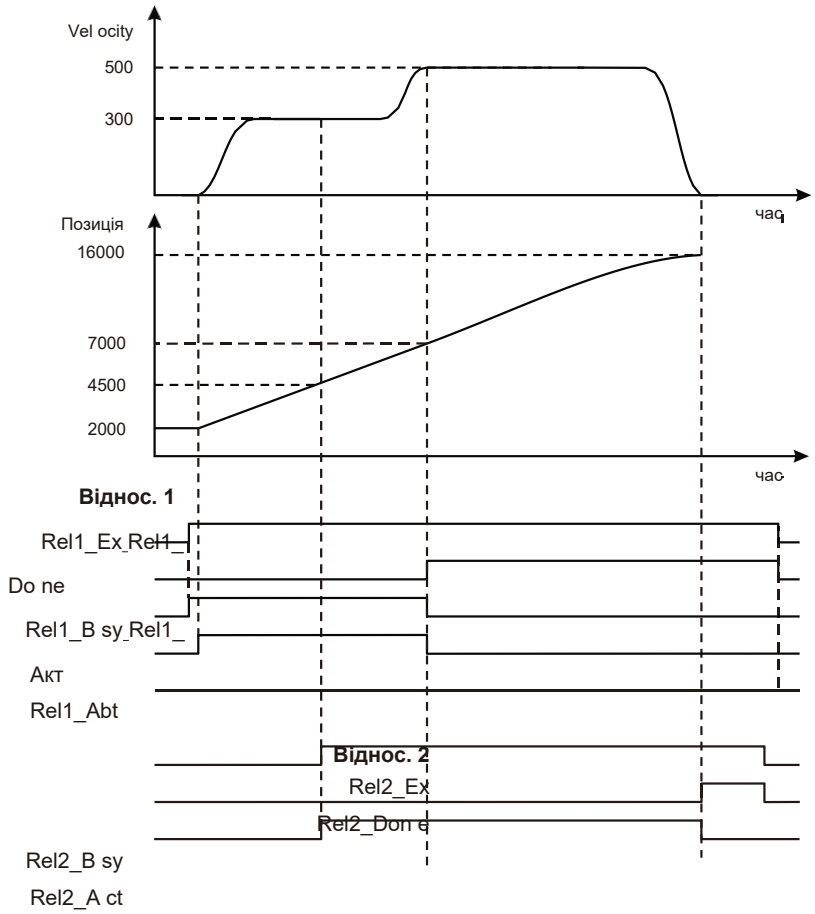
- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута і Rel2_Ex змінюється з FALSE на TRUE, Rel2_Bsy змінюється на TRUE, Rel1_Bsy і Rel1_Act залишаються TRUE, а виконання першої інструкції MC_MoveRelative продовжується. Після досягнення цільової позиції Rel1_Done змінюється на TRUE. На даний момент швидкість становить 300 одиниць на секунду, що є цільовою швидкістю поточної інструкції, Rel1_Bsy і Rel1_Act змінюються на FALSE, Rel2_Act змінюється на TRUE, і виконання другої інструкції MC_MoveRelative починається негайно. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а Rel2_Bsy та Rel2_Act тим часом змінюються на FALSE.
- ❖ Коли Rel1_Ex змінюється з TRUE на FALSE, Rel1_Done змінюється на FALSE. Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

■ **Rel2_BM =mcBlending _Далі**



- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута і Rel2_Ex змінюється з FALSE на TRUE, Rel2_Bsy змінюється на TRUE, Rel1_Bsy і Rel1_Act залишаються TRUE, а виконання першої інструкції MC_MoveRelative продовжується. Після досягнення цільової позиції Rel1_Done змінюється на TRUE. На даний момент швидкість становить 500 одиниць/секунду, що є цільовою швидкістю буферизованої інструкції; Rel1_Bsy і Rel1_Act змінюються на FALSE; Rel2_Act змінюється на TRUE і починається виконання другої інструкції MC_MoveRelative. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а Rel2_Bsy та Rel2_Act тим часом змінюються на FALSE.
- ❖ Коли Rel1_Ex змінюється з TRUE на FALSE, Rel1_Done змінюється на FALSE. Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

■ **Rel2_BM = mcBlending _High**



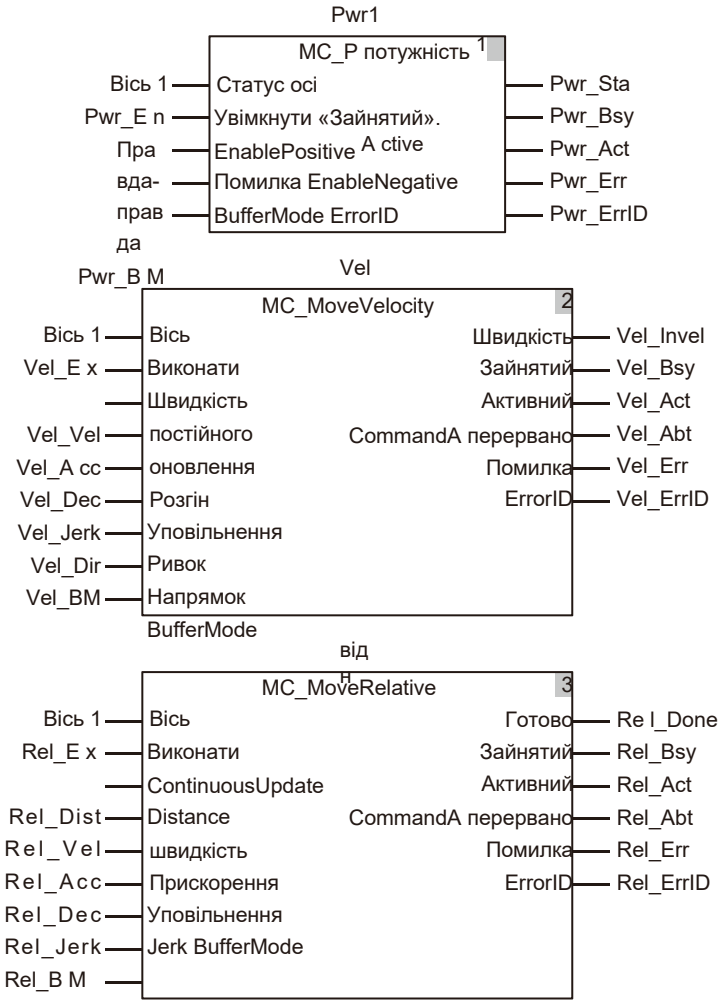
- ❖ Коли Rel1_Ex змінюється з FALSE на TRUE, Rel1_Bsy змінюється на TRUE. Через один період Rel1_Act змінюється на TRUE і починається виконання першої інструкції MC_MoveRelative. Поки цільова позиція ще не досягнута і Rel2_Ex змінюється з FALSE на TRUE, Rel2_Bsy змінюється на TRUE, Rel1_Bsy і Rel1_Act залишаються TRUE, а виконання першої інструкції MC_MoveRelative продовжується. Після досягнення цільової позиції Rel1_Done змінюється на TRUE. На даний момент швидкість становить 500 одиниць/секунду, що є вищою з цільових швидкостей поточної інструкції та буферизованої інструкції; Rel1_Bsy і Rel1_Act змінюються на FALSE; Rel2_Act змінюється на TRUE і починається виконання другої інструкції MC_MoveRelative. Коли цільова позиція досягнута, Rel2_Done змінюється на TRUE, а Rel2_Bsy та Rel2_Act тим часом змінюються на FALSE.
- ❖ Коли Rel1_Ex змінюється з TRUE на FALSE, Rel1_Done змінюється на FALSE. Коли Rel2_Ex змінюється з TRUE на FALSE, Rel2_Done змінюється на FALSE.

● **Приклад 2**

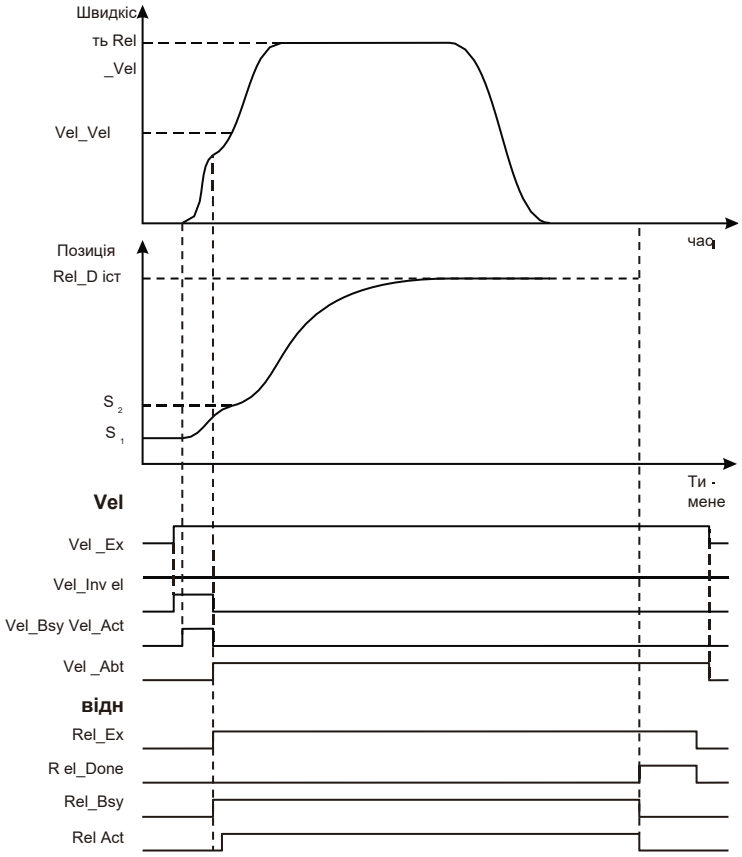
У наступному прикладі пояснюється стан осі для різних значень *BufferMode* за допомогою інструкції MC_MoveVelocity та інструкції MC_MoveReltave, яка є буферизованою інструкцією.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Вісь1	USINT	1
Pwr_En	BOOL	ПОМИЛКОВИЙ
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	СЛОВО	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	ПОМИЛКОВИЙ
Vel _Вел	LREAL	10000,0
Vel _Acc	LREAL	10000,0
Vel _гру	LREAL	10000,0
Vel _Jerk	LREAL	10000,0
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	СЛОВО	
відн	MC_MoveRelative	
Rel_Ex	BOOL	ПОМИЛКОВИЙ
Rel_Dist	LREAL	100000,0
Rel_Vel	LREAL	20000,0
Rel_Acc	LREAL	10000,0
Rel_Dec	LREAL	10000,0
Rel_Jerk	LREAL	10000,0
Рел_БМ	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	СЛОВО	

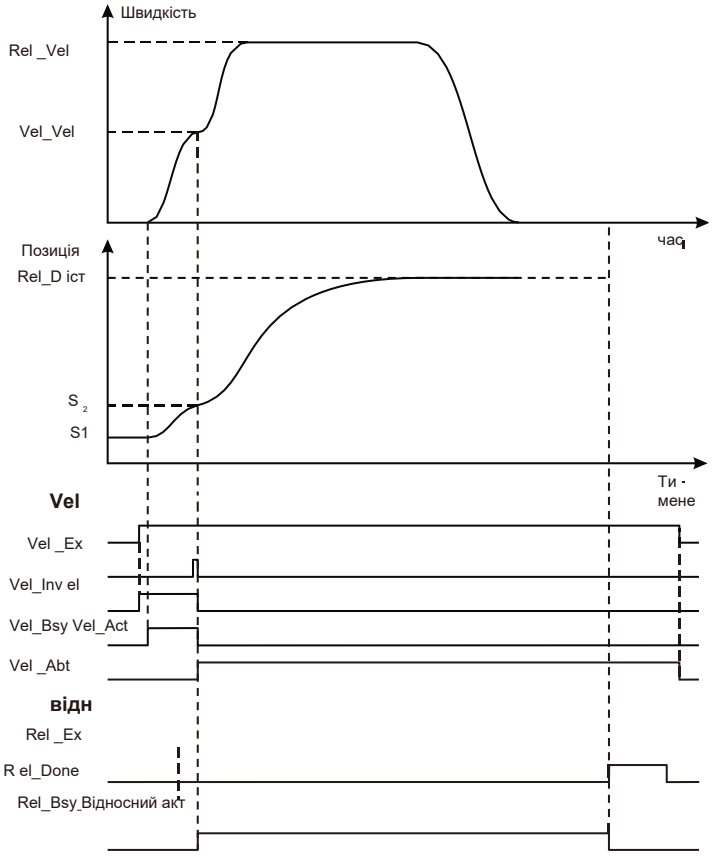

Pwr1
MC_P потужність 1
Вісь 1
Статус осі
Pwr_Sta
Pwr_E n
Увімкнути «Зайнятий».
Pwr_Bsy
Пра
EnablePositive
A ctive
Pwr_Act
вда-
Помилка EnableNegative
Pwr_Err
прав
BufferMode ErrorID
Pwr_ErrID
да
Pwr_B M
Vel
MC_MoveVelocity
2
Вісь 1
Вісь
Швидкість
Vel_Invel
Vel_E x
Виконати
Зайнятий
Vel_Bsy
Швидкість
Активний
Vel_Act
Vel_Vel
постійного
CommandA перервано
Vel_Abt
Vel_A cc
оновлення
Помилка
Vel_Err
Vel_Dec
Розгін
ErrorID
Vel_ErrID
Vel_Jerk
Уповільнення
Vel_Dir
Ривок
Vel_BM
Напрямок
BufferMode
від
н
MC_MoveRelative
3
Вісь 1
Вісь
Готово
Re l_Done
Rel_E x
Виконати
Зайнятий
Rel_Bsy
ContinuousUpdate
Активний
Rel_Act
Rel_Dist
Distance
CommandA перервано
Rel_Abt
Rel_Vel
швидкість
Помилка
Rel_Err
Rel_Acc
Прискорення
ErrorID
Rel_ErrID
Rel_Dec
Уповільнення
Rel_Jerk
Jerk BufferMode
Rel_B M

■ **Rel_BM =mcAborting**



❖ Коли Vel_Ex змінюється з FALSE на TRUE, Vel_Bsy змінюється на TRUE. Через один період Vel_Act змінюється на TRUE. До досягнення цільової швидкості вісь рухається зі швидкістю та прискоренням, визначеними інструкцією MC_MoveRelative, коли Rel_Ex змінюється з FALSE на TRUE. Коли Vel_Abt змінюється на TRUE, Vel_Bsy і Vel_Act змінюються на FALSE, інструкція швидкості переривається, інструкція MC_MoveRelative виконується, а Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли позиціонування завершено, Rel_Done змінюється на TRUE.

■ **Rel_BM = mcBuffered**



❖ Коли Vel_Ex змінюється з FALSE на TRUE, Vel_Bsy змінюється на TRUE. Через один період Vel_Act змінюється на TRUE. Rel_Ex змінюється з FASLE на TRUE, коли цільова швидкість не досягнута. Вісь не виконуватиме інструкцію MC_MoveRelatvie, доки не завершиться виконання інструкції швидкості. На даний момент Rel_Bsy змінюється на TRUE. Коли виконання інструкції швидкості завершено, Vel_Invel змінюється на TRUE, а через період інструкція MC_MoveRelatvie починає керувати віссю. Vel_Abt змінюється на TRUE, і інструкція швидкості переривається. Rel_Act має значення TRUE, що означає, що інструкція MC_MoveRelative починає керувати рухом осі. Rel_Done змінюється на TRUE після завершення позиціонування.

■ **(Ефект Rel_BM = mcBlendingLow, mcBlendingPrevious, mcBlendingNext або mcBlendingHigh такий самий, як і Rel_BM = mcBuffered.)**

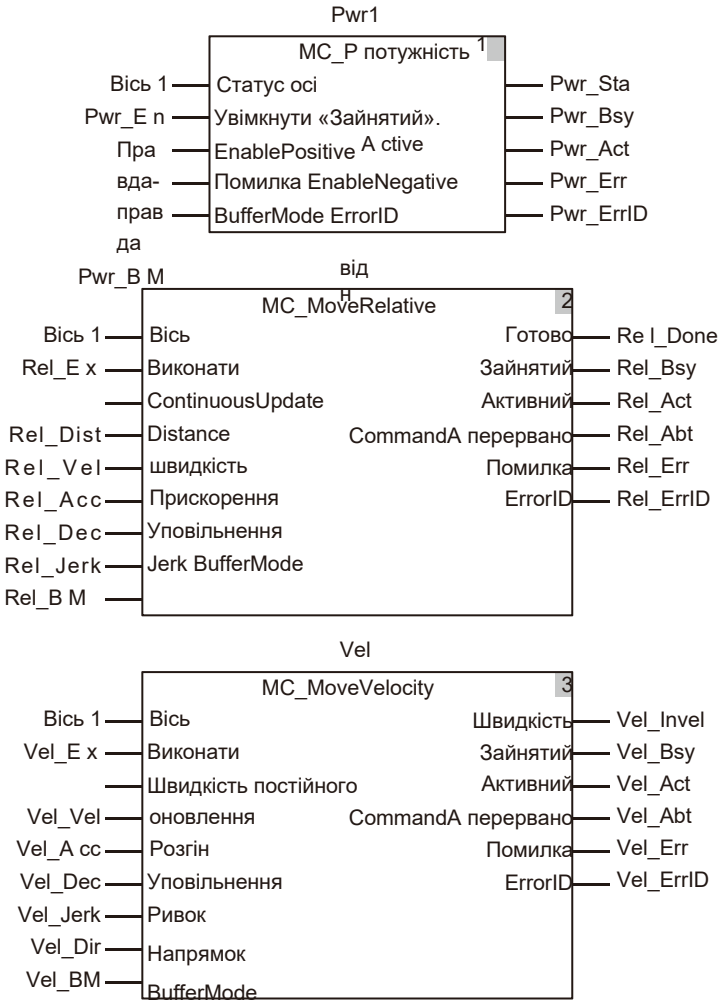
● **Приклад 3**

У прикладі пояснюється стан осі для різних значень *BufferMode* за допомогою інструкції MC_MoveRelative та інструкції MC_MoveVelocity, яка є буферизованою інструкцією.

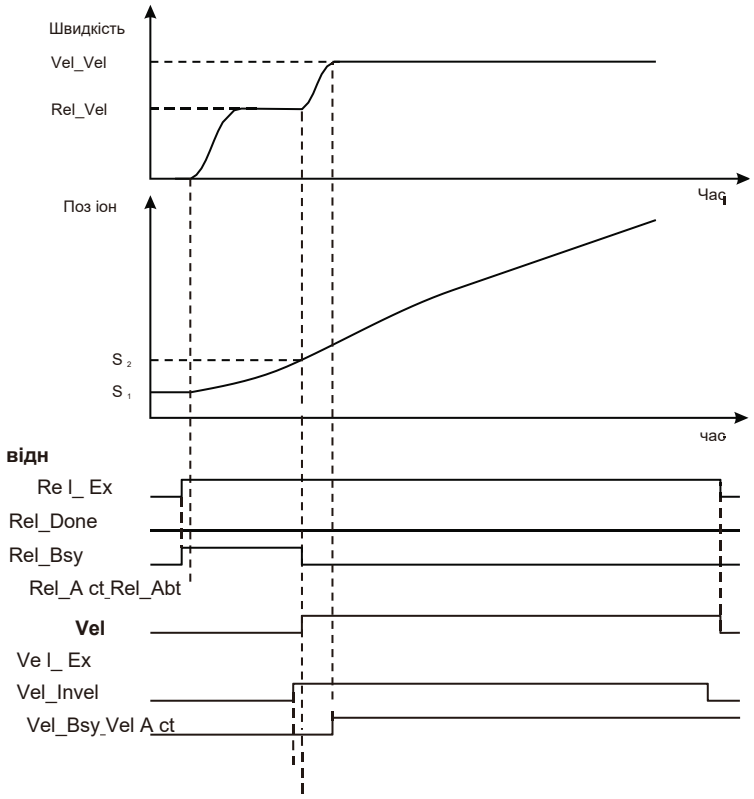
Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Вісь1	USINT	1
Pwr_En	BOOL	ПОМИЛКОВИЙ
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Pwr_Err	BOOL	
Pwr_ErrID	СЛОВО	
відн	MC_MoveRelative	
Rel_Ex	BOOL	ПОМИЛКОВИЙ
Rel_Dist	LREAL	100000,0
Rel_Vel	LREAL	10000,0
Rel_Acc	LREAL	10000,0
Rel_Dec	LREAL	10000,0
Rel_Jerk	LREAL	10000,0
Рел_БМ	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	СЛОВО	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	ПОМИЛКОВИЙ
Vel _Вел	LREAL	20000,0
Vel _Acc	LREAL	10000,0
Vel _гру	LREAL	10000,0
Vel _Jerk	LREAL	10000,0
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel_Err	BOOL	
Vel _ErrID	СЛОВО	

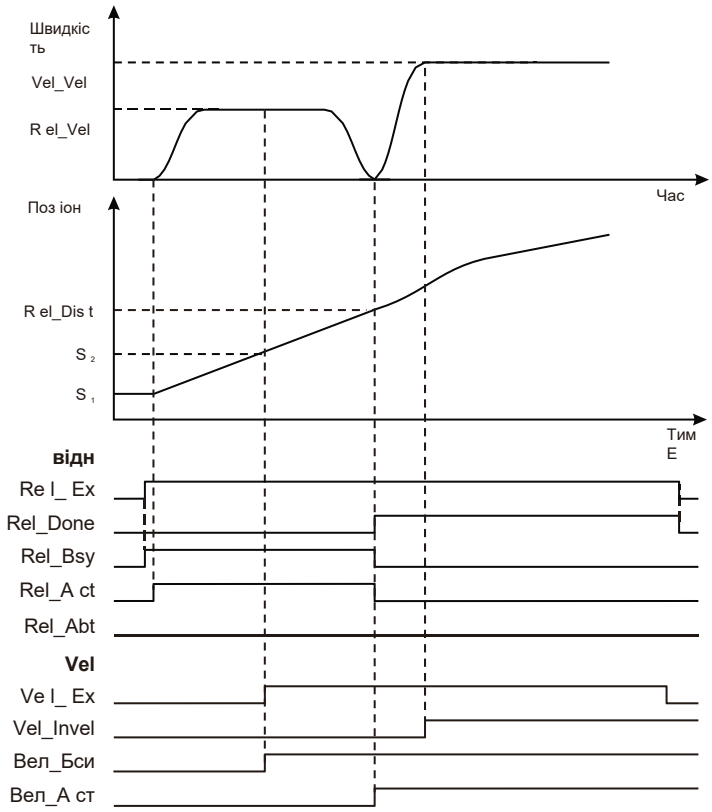

Pwr1
MC_P потужність
1
Вісь 1
Статус осі
Pwr_Sta
Pwr_E n
Увімкнути «Зайнятий».
Pwr_Bsy
Пра
EnablePositive
A ctive
Pwr_Act
вда-
Помилка EnableNegative
Pwr_Err
прав
BufferMode ErrorID
Pwr_ErrID
да
Pwr_B M
від
н
MC_MoveRelative
2
Вісь 1
Вісь
Готово
Re l_Done
Rel_E x
Виконати
Зайнятий
Rel_Bsy
ContinuousUpdate
Активний
Rel_Act
Rel_Dist
Distance
CommandA перервано
Rel_Abt
Rel_Vel
швидкість
Помилка
Rel_Err
Rel_Acc
Прискорення
ErrorID
Rel_ErrID
Rel_Dec
Уповільнення
Rel_Jerk
Jerk BufferMode
Rel_B M
Vel
MC_MoveVelocity
3
Вісь 1
Вісь
Швидкість
Vel_Invel
Vel_E x
Виконати
Зайнятий
Vel_Bsy
Швидкість постійного
Активний
Vel_Act
Vel_Vel
оновлення
CommandA перервано
Vel_Abt
Vel_A cc
Розгін
Помилка
Vel_Err
Vel_Dec
Уповільнення
ErrorID
Vel_ErrID
Vel_Jerk
Ривок
Vel_Dir
Напрямок
Vel_BM
BufferMode

■ **Vel _BM =mcAborting**



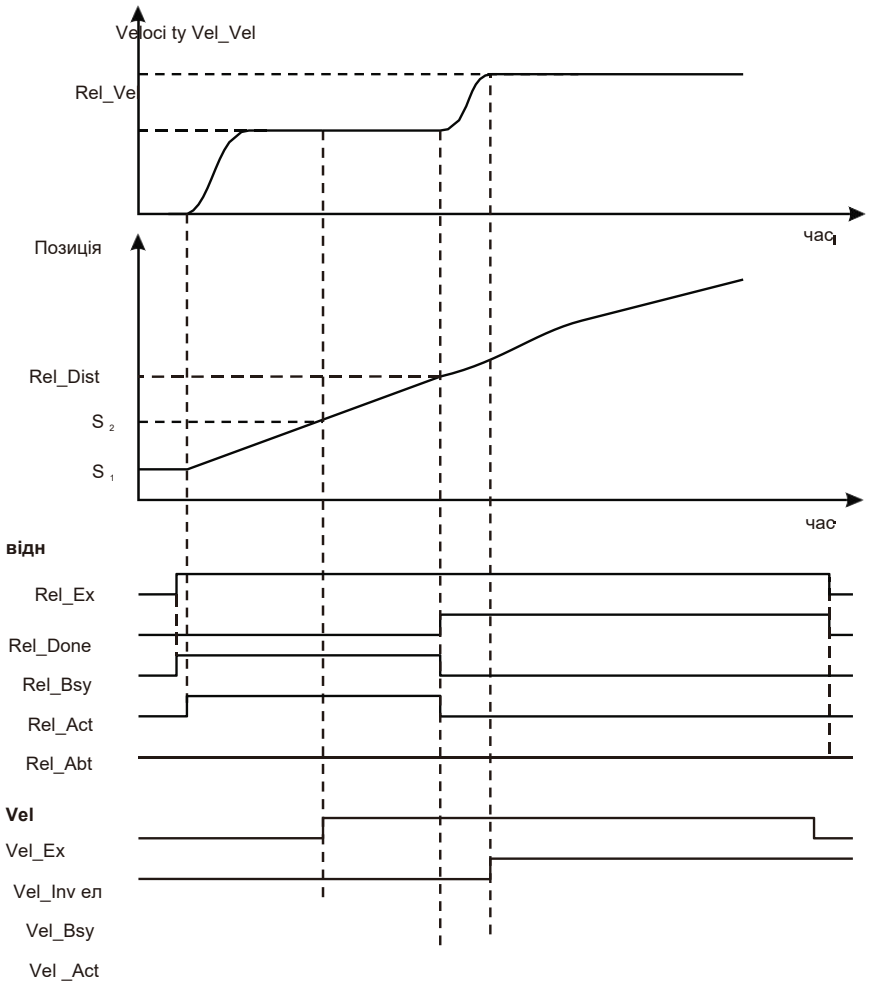
❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільове положення не досягнуто, Vel_Ex змінюється з FALSE на TRUE, вісь рухається зі швидкістю та прискоренням, визначеними інструкцією швидкості. Коли Rel_Abt змінюється на TRUE, Rel_Bsy і Rel_Act змінюються на FALSE, інструкція MC_MoveRelative переривається, а інструкція швидкості виконується. Vel_Bsy має значення TRUE, а через період Vel_Act змінюється на TRUE. Коли швидкість досягається, Vel_Invel змінюється на TRUE.

■ **Vel _BM = mcBuffered**



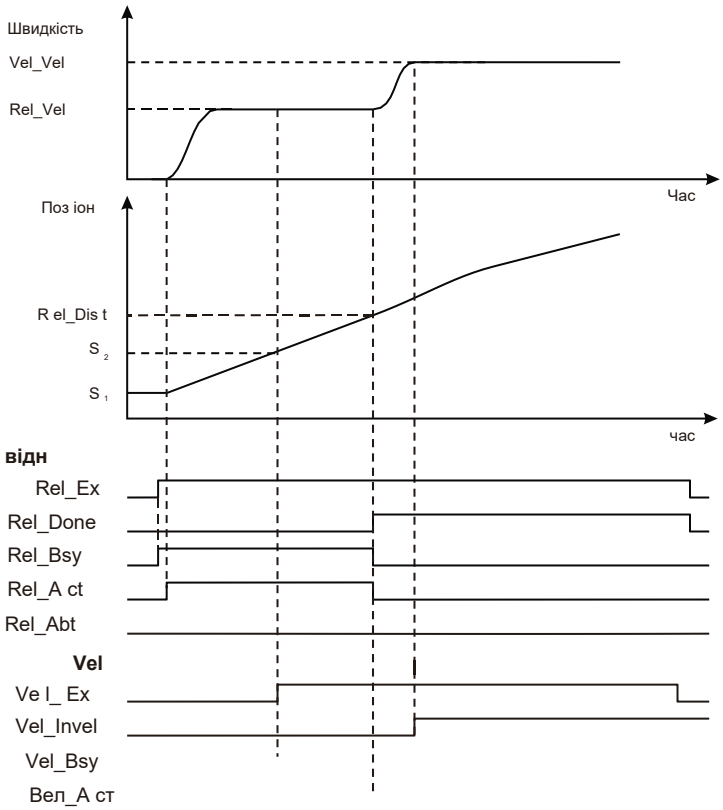
❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільова позиція не досягнута, Vel_Ex змінюється з FALSE на TRUE. Вісь сповільнюється до 0, коли виконання інструкції MC_MoveRelative завершено. Тоді Rel_Done змінюється на TRUE, Rel_Bsy і Rel_Act змінюються на FALSE, і вісь рухається зі швидкістю та прискоренням, визначеними інструкцією швидкості. Vel_Bsy змінюється на TRUE, а через період Vel_Act змінюється на TRUE. Rel_Invel змінюється на TRUE, коли досягається цільова швидкість.

■ **Vel _BM = mcBlendingLow**



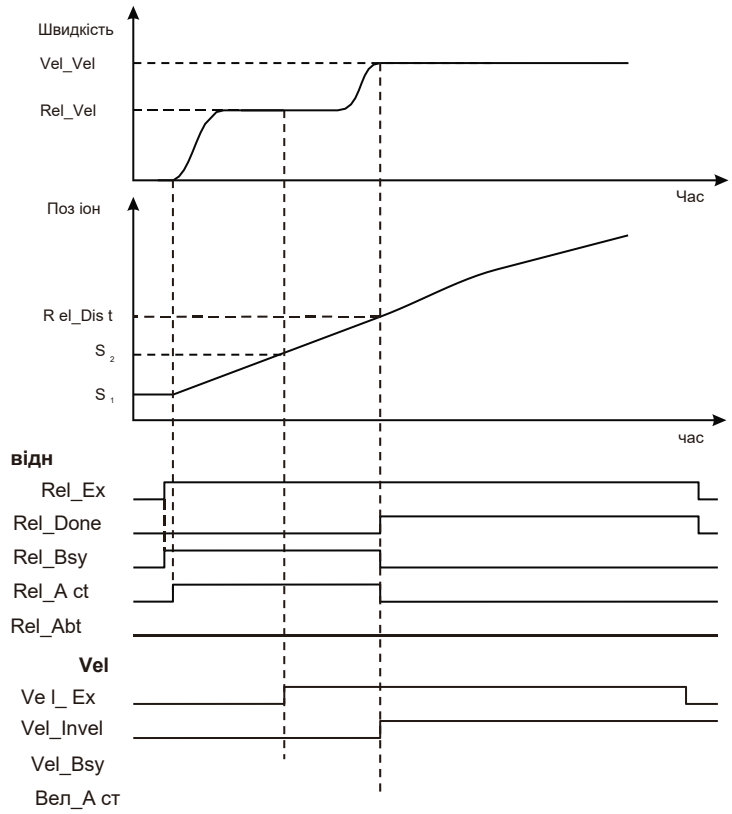
❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільова позиція не досягнута, Vel_Ex змінюється з FALSE на TRUE, а Vel_Bsy змінюється на TRUE. Вісь буде чекати завершення виконання MC_MoveRelative. Після завершення виконання MC_MoveRelative Rel_Done змінюється на TRUE, Rel_Bsy змінюється на FALSE і Rel_Act змінюється на FALSE. Тим часом Vel_Act змінюється на TRUE. На даний момент швидкість становить 10000 одиниць/секунду, що є нижчою з цільових швидкостей поточної інструкції та буферизованої інструкції. Виконання інструкції швидкості починається після завершення виконання інструкції MC_MoveRelative. Vel_Invel змінюється на TRUE, коли досягається цільова швидкість.

■ **Vel _BM = mcBlendingPrevious**



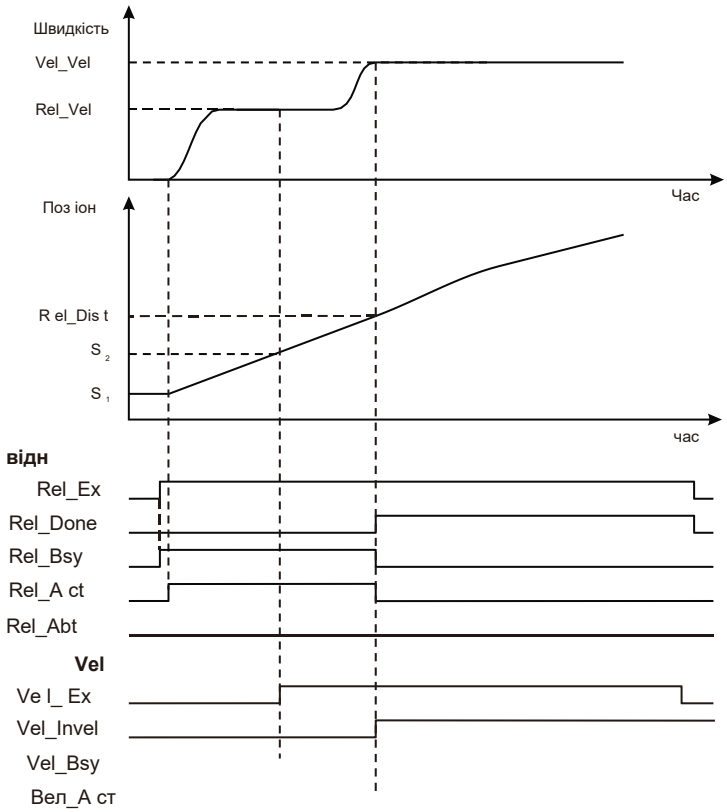
❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільова позиція не досягнута, Vel_Ex змінюється з FALSE на TRUE, а Vel_Bsy змінюється на TRUE. Вісь буде чекати завершення виконання MC_MoveRelative. Після завершення виконання MC_MoveRelative Rel_Done змінюється на TRUE, Rel_Bsy змінюється на FALSE, Rel_Act змінюється на FALSE, а тим часом Vel_Act змінюється на TRUE. На даний момент швидкість становить 10000 одиниць/секунду (це цільова швидкість поточної інструкції). Vel_Invel змінюється на TRUE, коли досягається цільова швидкість.

■ **Vel _BM = mcBlendingNext**



❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільова позиція не досягнута, Vel_Ex змінюється з FALSE на TRUE, а Vel_Bsy змінюється на TRUE. Вісь буде чекати завершення виконання MC_MoveRelative. Після завершення виконання MC_MoveRelative Rel_Done змінюється на TRUE, Rel_Bsy змінюється на FALSE, Rel_Act змінюється на FALSE, а тим часом Vel_Act змінюється на TRUE. На даний момент швидкість становить 20000 одиниць/секунду (це цільова швидкість буферизованої інструкції). Vel_Invel змінюється на TRUE, коли досягається цільова швидкість.

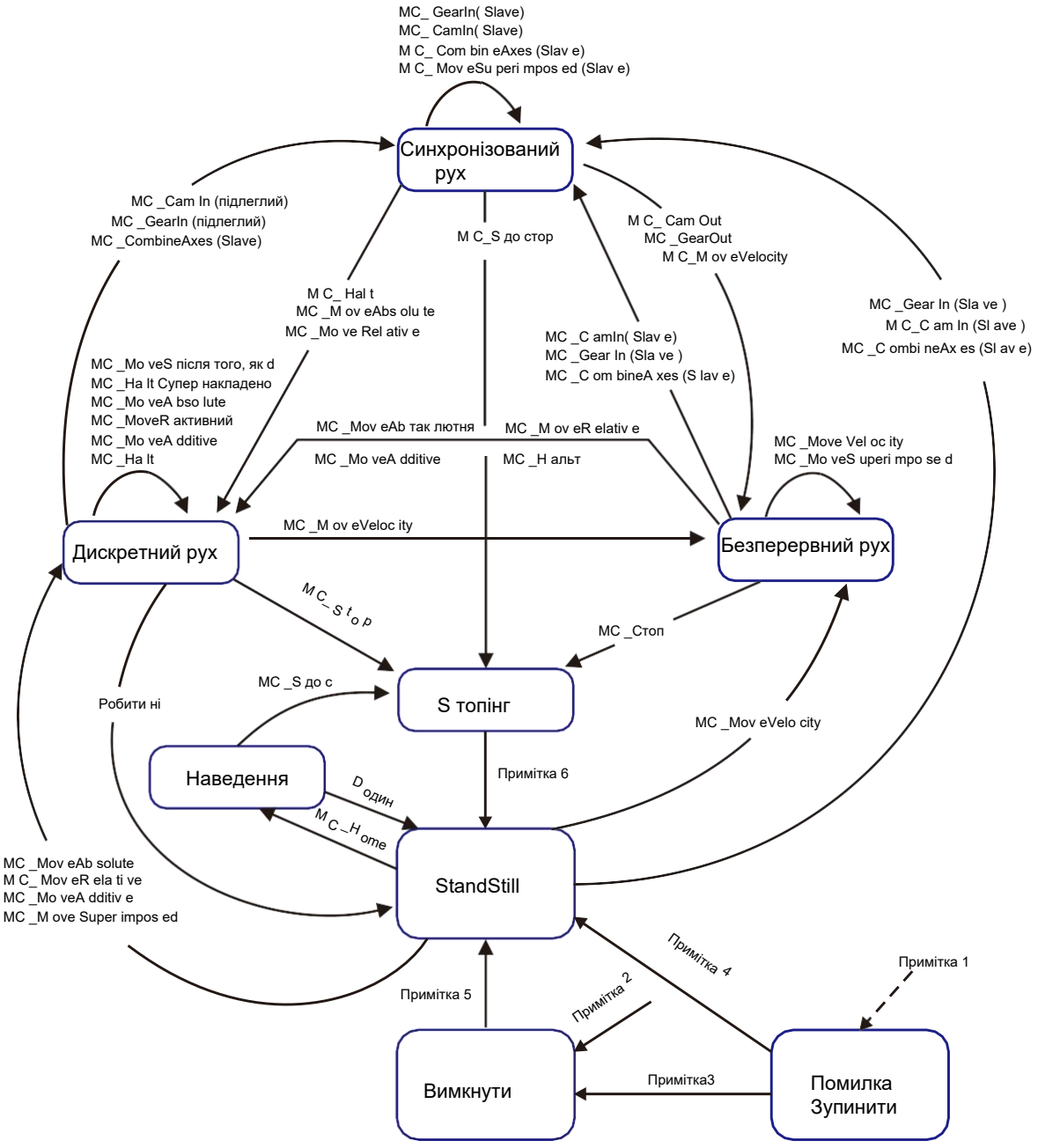
■ **Vel _BM = mcBlendingHigh**



❖ Коли Rel_Ex змінюється з FALSE на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE. Коли цільова позиція не досягнута, Vel_Ex змінюється з FALSE на TRUE, а Vel_Bsy змінюється на TRUE. Вісь буде чекати завершення виконання MC_MoveRelative. Після завершення виконання MC_MoveRelative Rel_Done змінюється на TRUE, Rel_Bsy змінюється на FALSE і Rel_Act змінюється на FALSE. На даний момент швидкість становить 20000 одиниць/секунду (що є вищою з цільових швидкостей поточної інструкції та буферизованої інструкції). Потім вісь рухається відповідно до швидкості, прискорення та уповільнення, визначених інструкцією швидкості. Vel_Invel змінюється на TRUE, коли досягається цільова швидкість.

10.4 Державна машина

Коли контролер руху використовує інструкцію керування рухом для керування кожною віссю, для кожної осі існує один стан внутрішнього запуску, а стани осі перемикаються відповідно до інструкцій кінцевого автомата, наведених нижче. Кінцевий автомат визначає інструкції руху, які можуть бути виконані в усіх станах, і стани після виконання інструкцій руху. Використовуючи інструкції руху, користувачі могли судити, чи можна використовувати певну інструкцію в поточному стані через кінцевий автомат.
Кінцевий автомат контролера руху зображено, як показано нижче, а стрілка вказує на стан осі.



Примітка 1 : вісь у будь-якому стані перейде в стан ErrorStop, доки на осі виникає помилка.

Примітка 2 : вісь переходить у стан Disabled, якщо в жодному стані не виникає помилка осі, а *Enable* для MC_Power має значення FALSE.

Примітка 3 **:** коли *статус* MC_Power має значення FALSE, інструкція MC_Reset використовується для скидання осі до стану Disabled.

Примітка 4 **:** коли *Enable* і *Status* MC_Power мають значення TRUE, інструкція MC_Reset використовується для скидання осі до стану зупинки.

Примітка 5 **:** вісь переходить із вимкненого стану в стан *зупинки* , коли інструкція MC_Power використовується для ввімкнення осі, а *стан* MC_Power має значення TRUE.

Примітка 6 **:** вісь переходить із стану зупинки в стан *нерухомості* , коли *Done* для MC_Stop має значення TRUE і

Виконання MC_Stop є FALSE.

Немає.	Стан осі	Індикація
1	StandStill	Стан перед виконанням
2	Вимкнено	Стан відсутності виконання
3	ErrorStop	Стан помилки
4	Зупинка	Стан зупинки
5	Наведення	Початковий стан
6	Дискретний рух	Стан дискретного руху
7	Безперервний рух	Стан безперервного руху
8	Синхронізований рух	Стан синхронного руху

Примітка. Стан осі можна оцінити відповідно до вихідних параметрів інструкції MC_ReadStatus. Зверніться до розділу 11.3.17, щоб дізнатися більше про інструкцію MC_ReadStatus.

ПАМ'ЯТКА

11

Розділ 11 Інструкції з керування рухом

Зміст

11.1 Table of Motion Control Instructions 11-4

11.2 About Motion Control Instructions 11-6

11.2.1 Композиція інструкції керування рухом 11-6

11.2.2 Мови програми, які підтримують інструкції керування рухом..... 11-6

11.2.3 Конфігурація інструкцій керування рухом 11-6

11.3 Single-axis Instructions 11-7

11.3.1 MC_Power 11-7

11.3.2 MC_Home..... 11-17

11.3.3 MC_MoveVelocity 11-22

11.3.4 MC_Halt 11-30

11.3.5 MC_Stop 11-35

11.3.6 MC_MoveRelative 11-40

11.3.7 MC_MoveAdditive 11-48

11.3.8 MC_MoveAbsolute 11-56

11.3.9 MC_MoveSuperimposed 11-65

11.3.10 MC_HaltSuperimposed 11-73

11.3.11 MC_SetPosition 11-78

11.3.12 MC_SetOverride..... 11-89

11.3.13 MC_Reset..... 11-93

11.3.14 DMC_SetTorque 11-96

11.3.15 MC_ReadAxisError..... 11-100

11.3.16 MC_ReadActualPosition 11-102

11.3.17 MC_ReadStatus 11-107

11.3.18 MC_ReadMotionState..... 11-112

11.3.19 DMC_ReadParameter_Motion..... 11-117

11.3.20 DMC_WriteParameter_Motion 11-122

11.3.21 DMC_TouchProbe 11-126

11.3.22 DMC_TouchProbeCyclically 11-135

11.3.23 DMC_Jog 11-139

11.3.24 DMC_MoveVelocityStopByPos 11-142

11.3.25 DMC_MoveVelocityStopByLinePos 11-146

11.3.26 DMC_ReadPositionLagStatus 11-150

11.3.27 DMC_WritePositionLagSetting 11-153

11.3.28 DMC_ChangeMechanismGearRatio 11-155
11.3.29 DMC_TorqueControl 11-157
11.3.30 DMC_MoveVelocity 11-162
11.3.31 DMC_SwitchSoftLimit 11-163

11.4 Coupling Instructions.....11-165
11.4.1 MC_GearIn 11-165
11.4.2 MC_GearOut..... 11-172
11.4.3 MC_CombineAxes 11-177
11.4.4 Впровадження електронної камери 11-185
11.4.5 MC_CamIn 11-186
11.4.6 MC_CamOut 11-207
11.4.7 DMC_CamReadPoint 11-213
11.4.8 DMC_CamWritePoint 11-217
11.4.9 DMC_CamSet 11-219
11.4.10 DMC_CamReadTappetStatus 11-223
11.4.11 DMC_CamReadTappetValue 11-228
11.4.12 DMC_CamWriteTappetValue..... 11-231
11.4.13 DMC_CamAddTappet 11-235
11.4.14 DMC_CamDeleteTappet 11-239

11.5 Application Instructions.....11-242
11.5.1 Технологія ротаційного різання 11-242
11.5.2 Параметри ротаційного різання 11-243
11.5.3 Функція керування функцією поворотного різання 11-244
11.5.4 Вступ до кривої кулачка з функцією поворотного різання 11-245
11.5.5 Інструкції з ротаційного різання 11-249
11.5.5.1 APF_RotaryCut_Init..... 11-249
11.5.5.2 APF_RotaryCut_In 11-252
11.5.5.3 APF_RotaryCut_Out 11-254
11.5.6 Приклад застосування інструкцій з ротаційного різання 11-256

11.6 G Code Instructions 11-260
11.6.1 Введення в ЧПУ 11-260
11.6.2 Формат введення коду G 11-260
11.6.3 Пояснення форматів коду G 11-261
11.6.4 Функції коду G 11-262
11.6.4.1 G90 (абсолютний режим) 11-262
11.6.4.2 G91 (відносний режим) 11-263
11.6.4.3 G0 (Швидке позиціонування)..... 11-264
11.6.4.4 G1 (лінійна інтерполяція) 11- 267
11.6.4.5 G2 (Кругова/спіральна інтерполяція за годинниковою стрілкою).11-271
11.6.4.6 G3 (Кругова/спіральна інтерполяція проти годинникової стрілки)11-278
11.6.4.7 G17/G18/G19 (вкажіть кругову площину інтерполяції)..... 11-284
11.6.4.8 G4 (інструкція щодо зупинки) 11-285

11.6.4.9 G50 (точна зупинка) 11-285
11.6.4.10 G51 (перехід по круговій траєкторії) 11-286
11.6.4.11 G52 (плавний перехід траєкторії) 11-287
11.6.4.12 Код M 11-289
11.6.5 Інструкції щодо коду G 11-291
11.6.5.1 DMC_CartesianCoordinate 11-291
11.6.5.2 DMC_ReadMFunction ..... 11 -297
11.6.5.3 DMC_ResetMFunction 11- 300
11.6.5.4 DMC_SetG0Para..... 11-303
11.6.5.5 DMC_SetG1Para..... 11-306
11.6.5.6 DMC_SetStartPosition 11- 309

11.7 Axes Group Instructions..... 11-313
11.7.1 DMC_AddAxisToGroup..... 11-313
11.7.2 DMC_RemoveAxisFromGroup 11-315
11.7.3 DMC_UngroupAllAxes..... 11-317
11.7.4 DMC_GroupEnable 11-319
11.7.5 DMC_ GroupStop 11-322
11.7.6 DMC_GroupInterrupt 11-328
11.7.7 DMC_GroupContinue..... 11-334
11.7.8 DMC_MoveDirectAbsolute 11-336
11.7.9 DMC_MoveDirectRelative 11-341
11.7.10 DMC_MoveLinearAbsolute 11- 346
11.7.11 DMC_MoveLinearRelative ..... 11 -360
11.7.12 DMC_MoveCircularAbsolute 11-373
11.7.13 DMC_MoveCircularRelative 11-384
11.7.14 DMC_GroupSetOverride 11-395
11.7.15 DMC_GroupReadActualPosition 11-400

11.8 Coordination Instructions..... 11-402
11.8.1 DMC_ControlAxisByPos 11-402
11.8.2 DMC_NC..... 11-407

11.1 Таблиця інструкцій з керування рухом

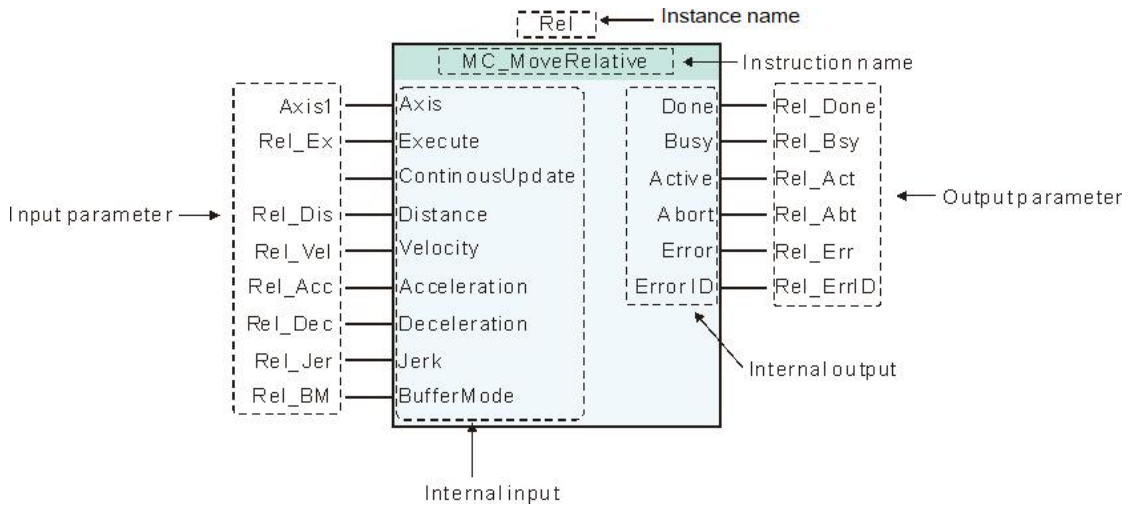
Набір інструкцій	Код інструкції	функція
Одноосьові інструкції	MC_Power	Power Servo
	MC_Home	Наведення
	MC_MoveVelocity	швидкість
	MC_Halt	Тимчасова зупинка
	MC_Stop	СТІЙ
	MC_MoveRelative	Відносне позиціонування
	MC_MoveAdditive	Адитивне позиціонування
	MC_MoveAbsolute	Абсолютне позиціонування
	MC_MoveSuperimposed	Накладене позиціонування
	MC_HaltSuperimposed	Припинення накладання
	MC_SetPosition	Встановити позицію
	MC_SetOverride	Встановити коефіцієнти перевизначення
	MC_Reset	Скинути
	DMC_SetTorque	Встановити крутний момент
	MC_ReadAxisError	Error читання осі
	MC_ReadActualPosition	Прочитайте фактичну позицію
	MC_ReadStatus	Читайте стан осі
	MC_ReadMotionState	Читайте стан руху
	DMC_ReadParameter_Motion	Прочитати параметр
	DMC_WriteParameter_Motion	Напишіть значення параметра
	DMC_TouchProbe	Захопити положення осі
	DMC_TouchProbeCyclically	Циклічно фіксуйте положення осі
	DMC_Jog	Біг підтюпцем
	DMC_MoveVelocityStopByPos	Зупинка на визначеній фазі
	DMC_MoveVelocityStopByLinePos	Зупинись у визначеному положенні
	DMC_ReadPositionLagStatus	Виявлення затримки позиції
	DMC_WritePositionLagSetting	Встановити затримку позиції
	DMC_ChangeMechanismGearRatio	Змінити значення параметра осі
	DMC_TorqueControl	Контроль крутного моменту
	DMC_MoveVelocity	Змініть швидкість і негайно зробіть її дійсною
	DMC_SwitchSoftLimit	Програмний кінцевий вимикач
Інструкція по зчепленню	MC_GearIn	Запустіть операцію E-Gear
	MC_GearOut	Завершення роботи E-Gear
	MC_CombineAxes	Комбіновані сокири
	MC_CamIn	Запустіть операцію E-Cam
	MC_CamOut	Завершення роботи E-Cam
	DMC_CamReadPoint	Прочитайте інформацію про камеру
	DMC_CamWritePoint	Встановіть параметри точки кулачка
	DMC_CamSet	Зробіть змінену інформацію про камеру ефективною
	DMC_CamReadTappetStatus	Прочитати статус штовхача

Набір інструкцій	Код інструкції	функція
	DMC_CamReadTappetValue	Зчитування параметрів штовхача
	DMC_CamWriteTappetValue	Встановіть параметри штовхача
	DMC_CamAddTappet	Додайте точку штовхача
	DMC_CamDeleteTappet	Видалити точку натискання
Інструкції із застосування	APF_RotaryCut_Init	Ініціалізація Rotary Cut
	APF_RotaryCut_In	Ротаційна врізка
	APF_RotaryCut_Out	Поворотний виріз
Інструкції з коду G	DMC_CartesianCoordinate	Робот з декартовими координатами
	DMC_ReadMFunction	Прочитайте стан коду M
	DMC_ResetMFunction	Скинути код M
	DMC_SetG0Para	Встановіть параметр G0
	DMC_SetG1Para	Встановіть параметр G1
	DMC_SetStartPosition	Встановити початкове положення
Інструкції групи Axes	DMC_ AddAxisToGroup	Додати вісь до групи осей
	DMC_RemoveAxisFromGroup	Видалити вісь із групи осей
	DMC_UngroupAllAxes	Видалити всі осі в групі осей
	DMC_ GroupEnable	Увімкніть групу осей
	DMC_ GroupStop	Зупинити рух поточної групи осей
	DMC_ GroupInterrupt	Тимчасово призупинити рух поточної групи осей
	DMC_ GroupContinue	Змусити призупинену групу осей продовжувати роботу
	DMC_MoveDirectAbsolute	Пряме абсолютне позиціонування
	DMC_MoveDirectRelative	Пряме взаємне позиціонування
	DMC_MoveLinearAbsolute	Лінійна абсолютна інтерполяція
	DMC_MoveLinearRelative	Лінійна відносна інтерполяція
	DMC_MoveCircularAbsolute	Кругова абсолютна інтерполяція
	DMC_MoveCircularRelative	Кругова відносна інтерполяція
	DMC_GroupSetOverride	Установити перевизначення
	DMC_GroupReadActualPosition	Прочитайте фактичне положення осей у групі
Координація	DMC_ControlAxisByPos	Інкрементний контроль позиції
	DMC_NC	Розбір файлів з ЧПУ

11.2 Про інструкції з керування рухом

11.2.1 Склад інструкції керування рухом

Інструкції, що починаються з «MC_» або «DMC», належать до інструкцій руху.



11.2.2 Мови програм, які підтримують інструкції керування рухом

Інструкції руху підтримують наступні два типи програмних мов. Докладніше див. у файлі довідки програмного забезпечення.

- Сходова діаграма (LD)
- Структурований текст (ST)

11.2.3 Конфігурація інструкцій керування рухом

Інструкції щодо руху можна додати лише до завдання події руху. Інакше вони не можуть бути виконані, якщо їх додати до інших типів завдань.

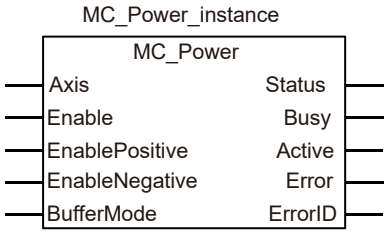
У наведеній нижче таблиці показано типи завдань і те, чи можна до цих завдань додати інструкції щодо руху.

Тип завдання		Чи можна додавати інструкції руху чи ні
Циклічний		Ні
Вільний хід		Ні
Ініціюється подією	Подія руху	Так
	Подія без руху	Ні

11.3 Одноосьові інструкції

11.3.1 MC_Power

FB/FC	Пояснення	Застосовна модель
FB	MC_Power використовується для ввімкнення або вимкнення відповідної осі сервоприводу.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE
EnablePositive	Зазначеній осі дозволено рухатися вперед лише за умови, що *Enable* має значення TRUE, а *EnablePositive* також має значення TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE
EnableNegative	Зазначеній осі дозволено рухатися назад лише за умови, що *Enable* має значення TRUE, а *EnableNegative* також має значення TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE
BufferMode	Укажіть поведінку MC_Power, коли *Enable* змінюється на FALSE	MC_Buffer_ Mode	0: mcAborting 1: mcBuffered (0)	Коли *Enable* змінюється на TRUE

Примітка:

Інструкції керування рухом можуть керувати осями сервоприводу для відповідних рухів лише після УВІМКНЕННЯ живлення. Коли живлення OFF, жодні інструкції керування рухом не можуть бути виконані.

● **Вихідні параметри**

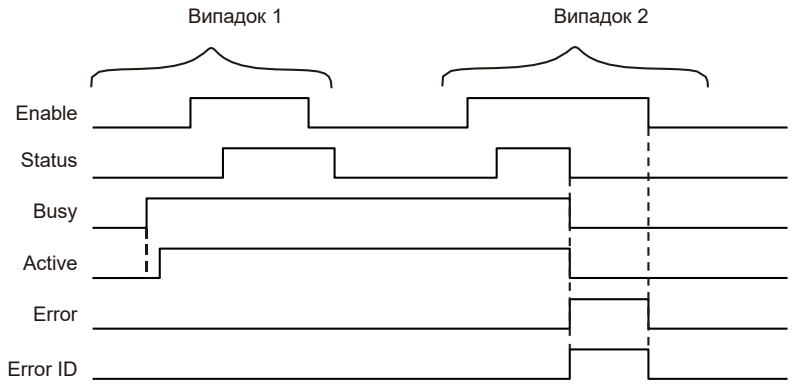
Назва параметра	функція	Тип даних	Допустимий діапазон
Status	TRUE, коли вісь увімкнено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Status	◆ Коли вісь включена.	◆ Коли *Enable* змінюється на FALSE. ◆ Коли *помилка* змінюється на TRUE.
Busy	◆ Коли інструкція виконується.	◆ Коли *помилка* змінюється на TRUE.
Active	◆ Інструкція починає керувати віссю.	◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли нестандартна ситуація усунена.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : Коли інструкція MC_Power виконується вперше, *Busy* змінюється на TRUE, а через один цикл *активний* змінюється на TRUE. Після того, як *Enable* змінюється з FALSE на TRUE і вісь увімкнено, *Status* змінюється на TRUE. Після того, як *Enable* змінюється з TRUE на FALSE, а вісь вимкнено, *Status* змінюється з TRUE на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, відповідний код помилки міститься в *ErrorID* , а *cmamyc* , *Busy* та *Active* змінюються на FALSE. *Error* змінюється на FALSE, коли помилку очищено.

- **функція**

Ця інструкція використовується для ввімкнення або вимкнення відповідної осі сервоприводу.

1. *Status* не зміниться на TRUE, якщо вісь ще не ввімкнено після того, як *Enable* встановлено на TRUE. Перед тим, як вісь почне рухатися, переконайтеся, що *Status уже змінено на TRUE.*
2. Коли *Enable* і *EnablePositive* мають значення TRUE, вісь, визначена інструкцією руху, може рухатися в позитивному напрямку.
3. Коли *Enable* має значення TRUE, а *EnablePositive* — FALSE, вісь, визначена інструкцією руху, є заборонено рухатися в позитивному напрямку. У цьому випадку виникне помилка, якщо якась інструкція руху використовується для переміщення осі вперед. Якщо вісь рухається назад вперед, то

Інструкція, яка керує рухом осі, буде перервана, а вісь припинить рух і перейде в стан зупинки.

4. Коли *Enable* і *EnableNegative* мають значення TRUE, вісь, визначена інструкцією руху, може рухатися в негативному напрямку.
5. Коли *Enable* має значення TRUE, а *EnableNegative* — FALSE, вісь, визначена інструкцією руху, є заборонено рух у негативному напрямку. У цьому випадку виникне помилка, якщо якась інструкція руху використовується для переміщення осі назад. Якщо вісь рухається вперед назад, інструкція, яка керує рухом осі, буде перервана, і вісь припинить рух і перейде в стан зупинки.
6. Коли вісь рухається в позитивному напрямку і *EnablePositive* змінюється з TRUE на FALSE, вісь сповільнить свою швидкість зі швидкістю уповільнення, визначеною поточною інструкцією руху, що керує віссю, і нарешті зупиниться на швидкості 0. Коли вісь рухається в негативному напрямку та *EnableNegative* змінюється з TRUE на FALSE, вісь сповільнить свою швидкість зі швидкістю уповільнення, визначеною поточною інструкцією руху, що керує віссю, і, нарешті, зупинитися на швидкості 0.
7. В принципі, для однієї осі можна використовувати лише один MC_Power. Якщо в програмі є дві інструкції MC_Power, де керується та сама вісь, зверніться до результату виконання MC_Power, який виконується із запізненням.
8. Поки інструкція руху керує віссю, *Enable* MC_Power змінюється з TRUE на FALSE, і те, чи перейде вісь у стан Disable, залежить від значення Buffermode.
9. Буферний режим
   BufferMode визначає поведінку MC_Power, коли *Enable* змінюється з TRUE на FALSE.

Введення	Вибір BufferMode	функція
Enable	0: mcAborting (переривати)	Коли *параметр «Увімкнути»* змінюється з «TRUE» на «ХИБНЬ», вісь негайно припиняє рух і вимикається (автомат переходить у стан «Вимкнено»). **Застереження: Будьте обережні під час роботи у разі будь-якої небезпеки для персоналу або пристроїв!**
	1: mcBuffered (очікування)	Коли *Enable* змінюється з TRUE на FALSE, вісь не переходить у стан Disabled одразу. Тільки коли вісь припиняє рух, кінцевий автомат спочатку переходить у стан зупинки, а через один цикл переходить у стан вимкнено.

Приклад програмування 1

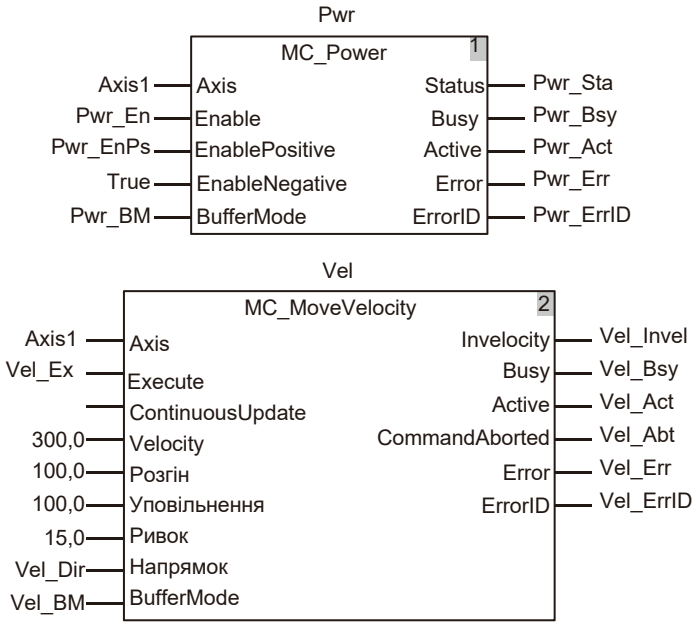
Приклад виконання інструкції MC_Power

Коли Pwr_En має значення TRUE, а Pwr_EnPs — FALSE, вісь, визначену інструкцією руху, заборонено рухатися в позитивному напрямку. Поки вісь рухається в позитивному напрямку і Pwr_EnPs змінюється з TRUE на FALSE, вісь сповільнюватиме свою швидкість зі швидкістю уповільнення, визначеною поточною інструкцією руху, що керує віссю, доки швидкість осі не досягне 0.

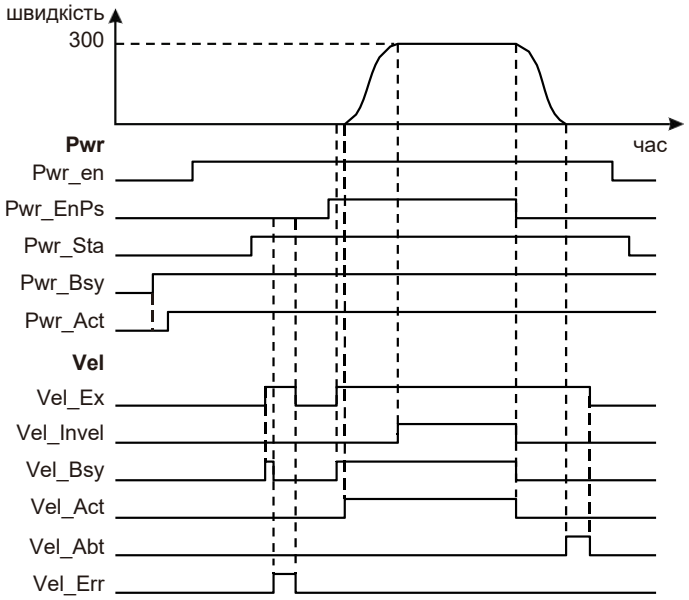
1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	0
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel_Err	BOOL	
Vel _ErrID	WORD	



2. Крива руху та часовий графік



- Коли Vel _Ex змінюється на TRUE вперше, Vel _Bsy змінюється на TRUE, а через один цикл Vel _Err змінюється на TRUE. У цей момент серводвигун не міг рухатися, оскільки Pwr_EnPs має значення FALSE.
- Коли Pwr_EnPs має значення TRUE, а Vel _Ex змінюється на TRUE вдруге, Vel _Bsy змінюється на TRUE; через один цикл Vel _Act змінюється на TRUE і серводвигун починає рухатися в позитивному напрямку. Коли серводвигун досягає цільової швидкості, Vel _Invel змінюється на TRUE.
- Коли Pwr_EnPs змінюється на FALSE, інструкція MC_Velocity переривається, і серводвигун починає уповільнювати свою швидкість зі швидкістю уповільнення, визначеною інструкцією MC_Velocity. Коли швидкість зменшується до 0, CommandAborted змінюється на TRUE.
- Коли Vel _Ex змінюється на FALSE, Vel _Abt змінюється на FALSE.
- Коли Pwr_En змінюється на FALSE, Pwr_Sta змінюється на FALSE після вимкнення осі.

Приклад програмування 2

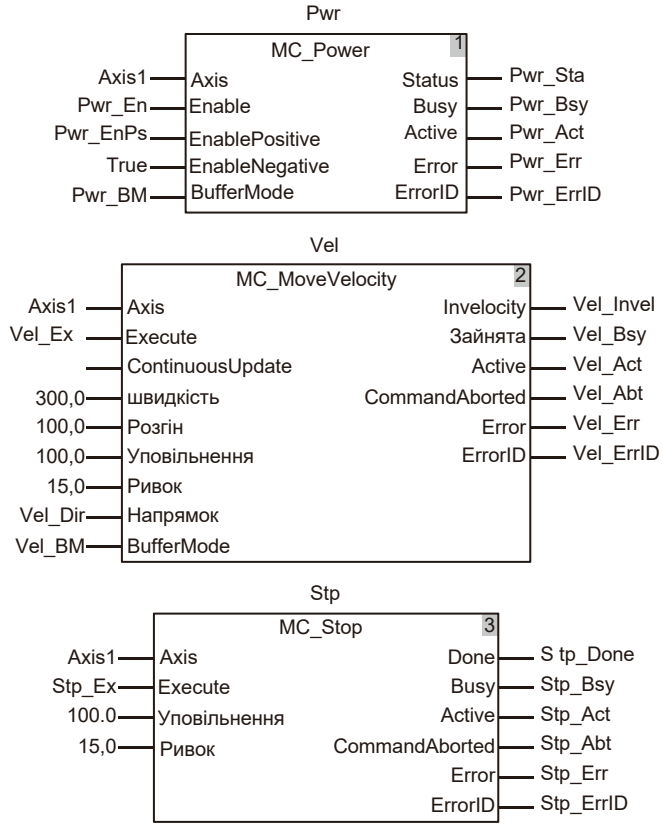
Приклад Vel _BM =0

Коли значення Vel _BM встановлено на 0 і Pwr_En змінюється з TRUE на FALSE, вісь перейде в стан Disabled, а швидкість буде негайно зменшена до 0.

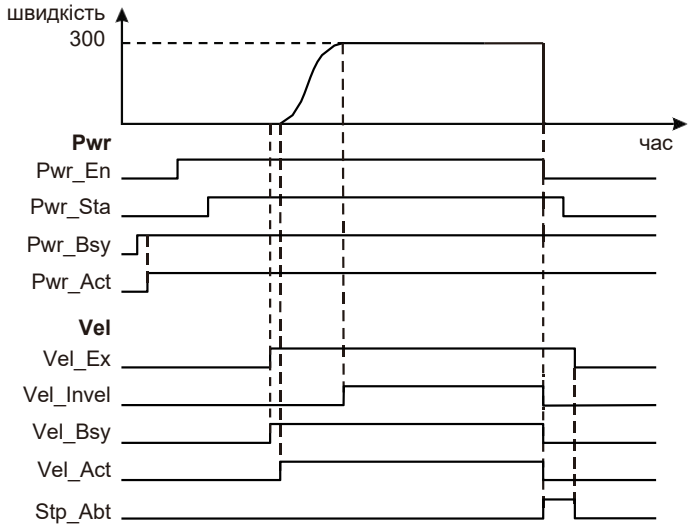
1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel _Ex	BOOL	FALSE

Ім'я змінної	Тип даних	Початкове значення
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	0
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	WORD	
Stp	MC_Stop	
Stp _Ex	BOOL	FALSE
Stp _Done	BOOL	
Stp _Bsy	BOOL	
Stp _Act	BOOL	
Stp _Abt	BOOL	
Stp _Err	BOOL	
Stp _ErrID	WORD	



2. Крива руху та часовий графік



- Коли Vel _Ex змінюється на TRUE, серводвигун починає рухатися в позитивному напрямку. Коли швидкість серводвигуна досягає цільової швидкості, Vel _Invel змінюється на TRUE.
- Коли Pwr_En змінюється на FALSE, швидкість серводвигуна зменшується до 0, і вісь одразу переходить у стан зупинки. У той же час Vel _Abt змінюється на TRUE і Vel _Bsy і Vel _Act змінюються на FALSE. Pwr_Sta змінюється на FALSE після вимкнення осі.
- Коли Vel _Ex змінюється на FALSE, Vel _Abt змінюється на FALSE.

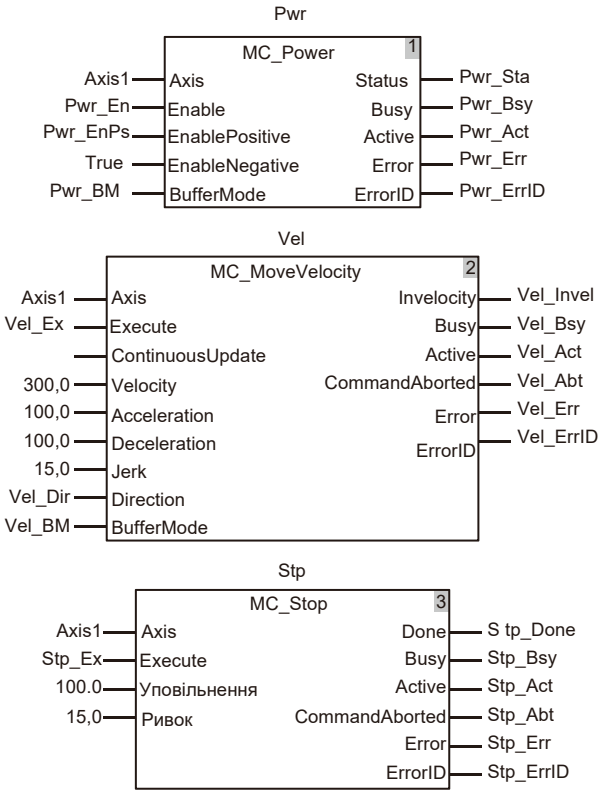
Приклад програмування 3

Приклад Vel_BM =1

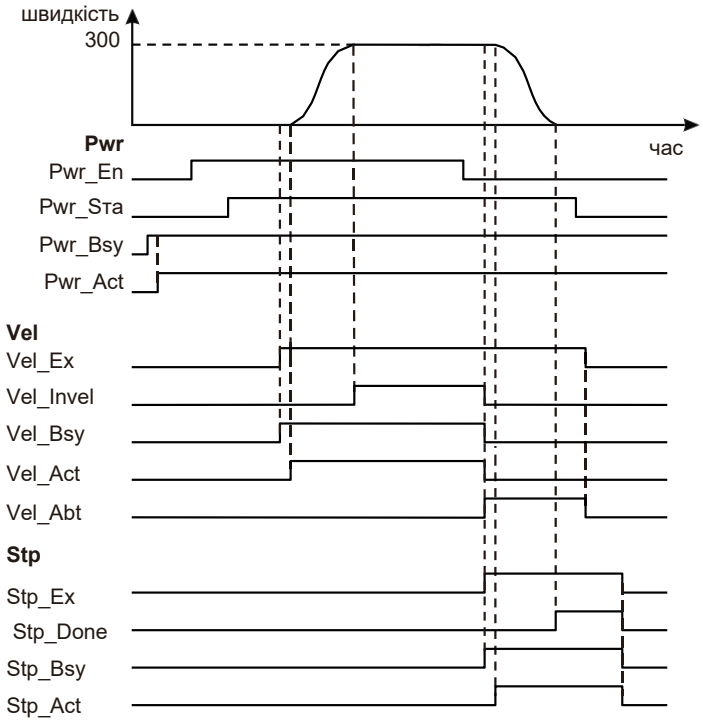
Buffermode встановлено значення 1 і *Enable змінюється з TRUE на FALSE, статус MC_Power* не зміниться, якщо вісь не припинить рух. Коли вісь припиняє рух, вісь спочатку перейде в стан зупинки, а через один цикл перейде в стан вимкнено.

1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	0
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel_Err	BOOL	
Vel _ErrID	WORD	
Stp	MC_Stop	
Stp _Ex	BOOL	FALSE
Stp _Done	BOOL	
Stp _Bsy	BOOL	
Stp _Act	BOOL	
Stp _Abt	BOOL	
Stp _Err	BOOL	
Stp _ErrID	WORD	



2. Крива руху та часовий графік



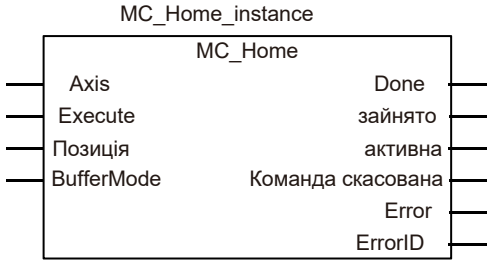
- Коли Vel_Ex змінюється на TRUE, Vel_Bsy змінюється на TRUE; через один цикл Vel_Act змінюється на TRUE і серводвигун починає рухатися в позитивному напрямку. Коли швидкість серводвигуна досягає цільової швидкості, Vel_Invel змінюється на TRUE.
- Коли Pwr_En змінюється на FALSE, вісь не переходить у стан зупинки негайно. Коли Stp_Ex змінюється на TRUE, Stp_Bsy змінюється на TRUE; через один цикл Stp_Act змінюється на TRUE і серводвигун починає гальмувати. Коли швидкість сервомотора падає до

0, Stp _Done змінюється на TRUE. Тим часом вісь переходить у стан зупинки, а Pwr_Sta змінюється на FALSE. Через один цикл вісь переходить у вимкнений стан.

- ❖ Коли Vel _Ex змінюється на FALSE, Vel _Abt змінюється на FALSE.
- ❖ Коли Stp _Ex змінюється на FALSE, Stp _Done, Stp _Bsy і Stp _Act змінюються на FALSE.

11.3.2 MC_Home

FB/FC	Пояснення	Застосовна модель
FB	MC_Home керує серводвигуном для виконання дії наведення відповідно до встановленого режиму та швидкості.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Позиція	Зсув початкової точки сервоприводу, одиниця: одиниця.	LREAL	Від'ємне число, додатне число і 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
BufferMode	Зарезервований	-	-	-

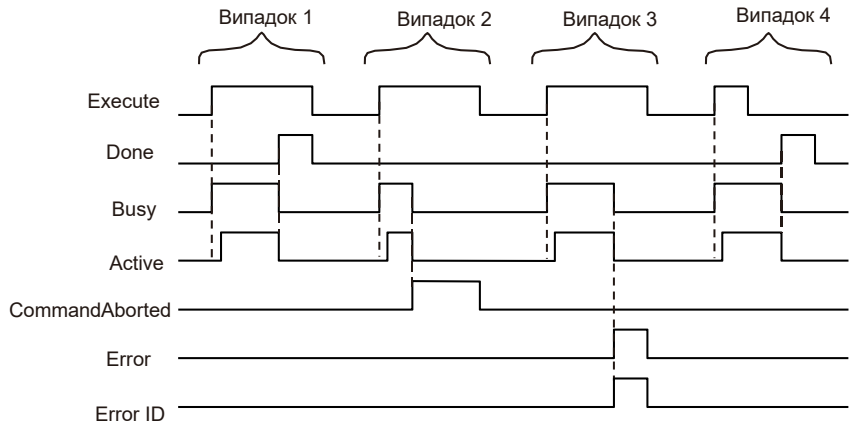
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли наведення завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли самонаведення завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а одним циклом пізніше *Active* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE.

Випадок 2 : коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з FALSE на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли виникає помилка, наприклад аварійні сигнали осі або Offline після того, як *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

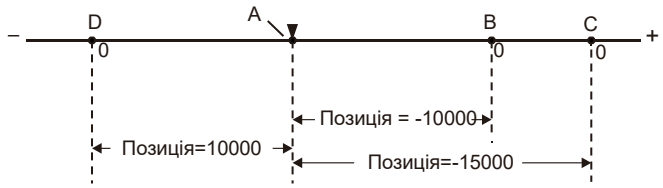
Випадок 4 : *Зроблені* зміни на TRUE, коли виконання інструкції завершено після зміни *Execute* з TRUE на FALSE під час виконання інструкції. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через один цикл *«Готово»* змінюється на FALSE.

● **функція**

1. Відповідно до встановленого режиму повернення до початкової точки, інструкція MC_Hme використовується для підключення вихідного вимикача та позитивного кінцевого вимикача або негативного кінцевого вимикача до зовнішніх вхідних точок сервоприводу, щоб досягти функції повернення до початкової точки.
2. Для справжніх осей режим наведення та швидкість фази 1 і швидкість фази 2 наведення встановлюються в налаштуваннях параметрів осі програмного забезпечення. Дивіться Додаток D для детальної інформації про режими наведення. Для віртуальних осей режим наведення може бути встановлений лише на режим 35.
3. Інструкція може бути виконана лише тоді, коли вісь перебуває в стані нерухомості. Інакше виникне помилка.
4. Параметр позиції визначає зміщення між механічною нульовою точкою та опорною нульовою точкою сервоприводу, як показано на малюнку нижче:

A	Механічна нульова точка, де знаходиться фотоелектричний датчик.
▼	Позиція - це місце, де знаходиться сервопривід після завершення виконання інструкції.

Для іншого значення *позиції* сервопривід зупиниться в механічній точці A під керуванням цієї інструкції. Але контрольна нульова точка положення сервоприводу зміниться, як показано нижче.



Оскільки Position=10000, контрольною нульовою точкою положення сервоприводу є точка D, а положення точки A дорівнює 10000;

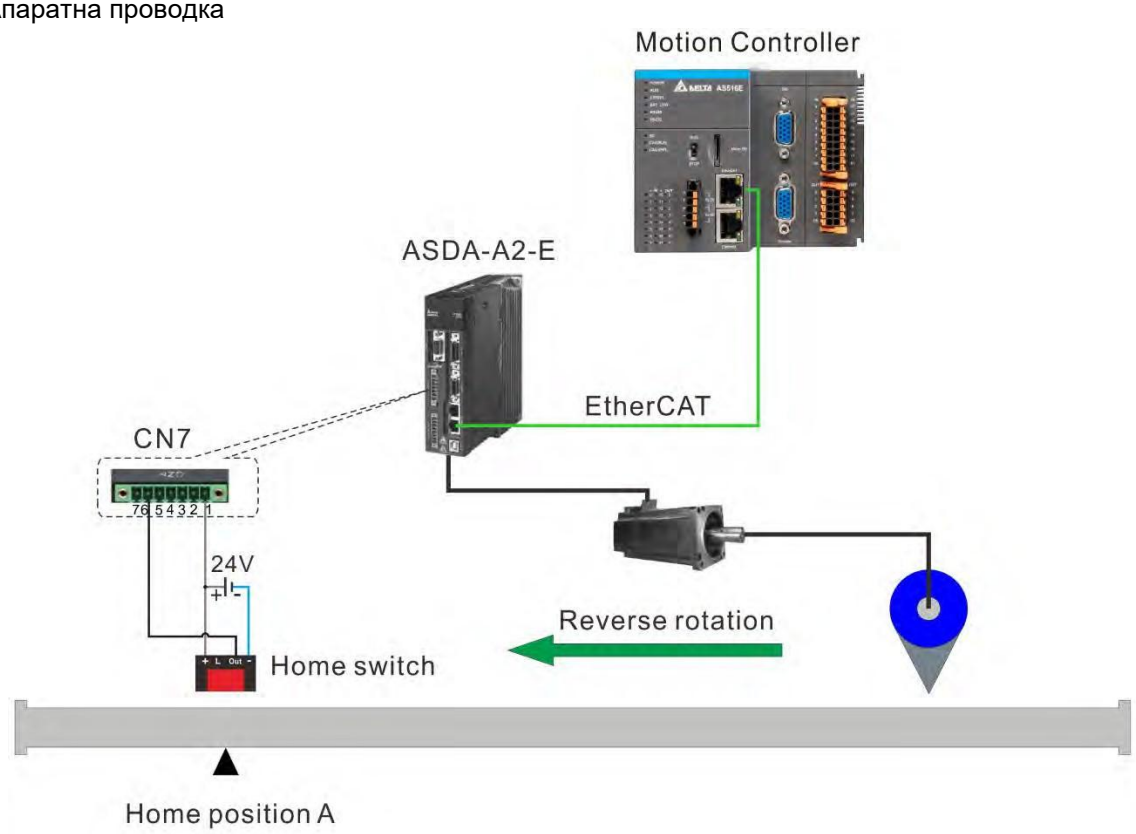
Оскільки Position=-15000, контрольною нульовою точкою положення сервоприводу є точка C, а положенням точки A є -15000;

Оскільки Position=-10000, контрольною нульовою точкою положення сервоприводу є точка B, а положенням точки A є -10000.

Приклад програмування

Виберіть відповідний режим наведення за допомогою положень механізму та фотоелектричного перемикача. Коли Хом_Ex змінюється з FALSE на TRUE, контролер руху керує серводвигуном для обертання та приводу механізму для повернення в механічне положення нульової точки A.

- Апаратна проводка



Примітка:

1. Коричнева клема фотоелектричного перемикача (24 В+) підключена до COM+ (контакт 1) CN7, а чорна клема (вихід) підключена до EDI13 (контакт 6).
2. Вхідна точка (EDI13) CN7 сервоприводу встановлена як вихідний перемикач, тобто P2-40 встановлено на 124.

- Вибір режиму самонаведення

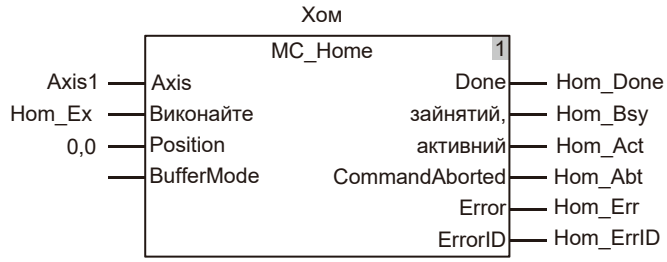
На малюнку підключення апаратного забезпечення видно, що механізм розглядає положення перемикача будинку як механічне положення нульової точки A. Перемикач будинку вимкнено перед пошуком будинку. У той час як механізм шукає початкову точку, сервопривод обертається в зворотному напрямку на початку, і режим наведення 21 може бути обраний для досягнення самонаведення. Налаштування для повернення до початкової точки у відповідних параметрах осі наступні.

Режим самонаведення	21
Швидкість першої фази (швидкість пошуку домашнього вимикача, одиниця вимірювання: об/м)	100
Швидкість другої фази (швидкість від знаходження домашнього вимикача до досягнення механічної нульової точки, одиниця: об/м)	10

Примітка: встановлені параметри осі дійсні після завантаження.

● **Таблиця змінних і програма**

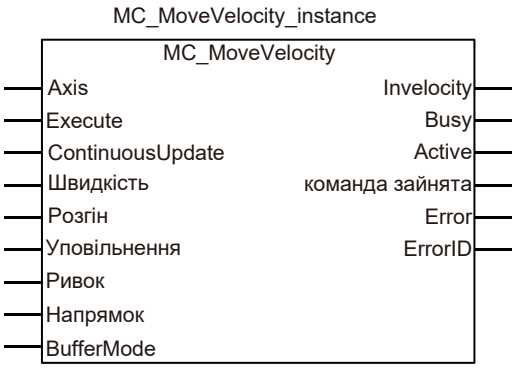
Ім'я змінної	Тип даних	Початкове значення
Хом	MC_Home	
Axis1	USINT	1
Hom_Ex	BOOL	FALSE
Hom_Done	BOOL	
Hom_Bsy	BOOL	
Hom_Act	BOOL	
Hom_Abt	BOOL	
Hom_Err	BOOL	
Hom_ErrID	WORD	



- ❖ Коли Hom_Ex змінюється з FALSE на TRUE, контролер руху керує рухом сервомотора. Механізм починає працювати в зворотному напрямку, повертається вперед після досягнення вихідного перемикача, а потім зупиняється в механічній нульовій точці. І таким чином механізм повертається до механічної нульової точки A.
- ❖ Коли буде досягнуто перемикання вдома, перехід до самовідведення завершено, і Hom_Done встановлено на ON.

11.3.3 MC_MoveVelocity

FB/FC	Пояснення	Застосовна модель
FB	MC_MoveVelocity керує рухом осі на основі встановленого прискорення та уповільнення до досягнення заданої цільової швидкості, а потім вісь рухається із заданою швидкістю.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
Invelocity	Укажіть цільову швидкість (одиниця: одиниця/секунда)	LREAL	Додатне число, від'ємне число і 0 (Необхідно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Укажіть цільове прискорення (одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (Необхідно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Укажіть цільове уповільнення (одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (Необхідно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни прискорення або уповільнення цілі. (Одиниця: одиниця/с ^{3})	LREAL	Додатне число (Необхідно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Напрямок	Вкажіть напрямок обертання 1: Позитивний напрямок 3: Негативний напрямок 4: Напрямок струму (Коли двигун зупинено, напрямок струму є позитивним).	MC_Direction	1: mcPositiveDirection, 3: mcNegativeDirection 4: mcCurrentDirection, (1)	Коли *Execute* змінюється з FALSE на TRUE
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	MC_Buffer_Mode	0: mcAborting 1: mcBuffered 2: mcBlendingLow 3: mcBlendingPrevious 4: mcBlendingNext 5: mcBlendingHigh (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_MoveVelocity виконується, коли *Execute* змінюється з FALSE на TRUE. Інструкція може бути повторно виконана, коли *Execute* інструкції знову змінюється з FALSE на TRUE незалежно від того, чи завершено виконання інструкції. На даний момент такі параметри, як *швидкість* , *прискорення* , *уповільнення, ривок* і *напрямок,* знову діють, а інші параметри не діють. Коли інструкція швидкості має зв'язок BufferMode з іншою інструкцією руху, параметри будуть дійсними після того, як параметри інструкції змінено та інструкцію повторно запущено. Попереднє співвідношення буферного режиму залишається, і швидкість переходу буде перераховано.
2. *Invelocity* залишається TRUE, навіть якщо цільову швидкість змінено через MC_SetOverride після завершення виконання інструкції velocity (тобто *Invelocity* змінюється з FALSE на TRUE). *Invelocity* зміниться з FALSE на TRUE, коли нова цільова швидкість буде досягнута після цільової швидкості. змінено через MC_SetOverride до завершення виконання MC_MoveVelocity (коли *Invelocity* має значення FALSE).
3. Зверніться до розділу 10.2 щодо співвідношення між *положенням, швидкістю, прискоренням і ривком* .
4. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

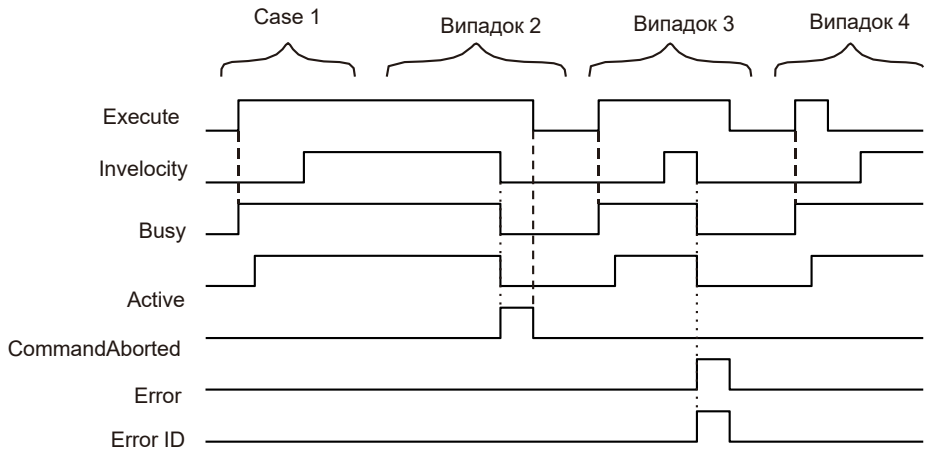
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Invelocity	TRUE, коли досягнуто цільової швидкості.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Invelocity	◆ При досягненні цільової швидкості.	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ *Invelocity* змінюється на FALSE негайно, коли *Execute* знову змінюється з FALSE на TRUE, якщо значення вхідних параметрів переглядаються після досягнення цільової швидкості. Якщо значення вхідних параметрів не змінено після завершення виконання інструкції та *Execute* знову змінюється з FALSE на TRUE, *Invelocity* негайно змінюється на FALSE, а *Invelocity* змінюється на TRUE у наступному циклі.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а одним циклом пізніше *Active* змінюється на TRUE. Коли цільова швидкість досягнута, *Invelocity* змінюється на TRUE, а параметри *Busy* та *Active* залишаються TRUE.

Випадок 2 : коли *Execute* має значення TRUE, інструкція переривається іншою інструкцією, а *CommandAborted* змінюється на TRUE. Тим часом *Invelocity* , *Busy і Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли виникає помилка, наприклад помилка параметра, коли *для Execute* встановлено значення TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *Invelocity, Busy* і *Active* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : під час виконання інструкції *Invelocity* змінюється на TRUE, коли цільова швидкість досягається після зміни *Execute* з TRUE на FALSE. При цьому *параметри «Зайнятий»* і *«Активний»* залишаються значеннями TRUE.

● функція

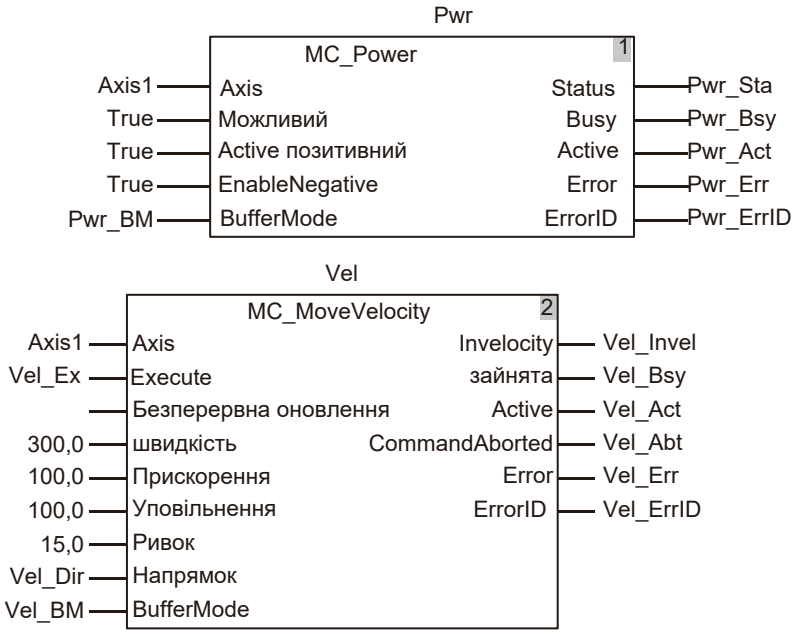
MC_MoveVelocity керує віссю для прискорення вгору або вниз відповідно до встановленого прискорення, уповільнення та ривка до досягнення заданої цільової швидкості, після чого вісь рухається з цільовою швидкістю. Напрямок рівномірного руху визначається вхідним параметром *Direction* . Значення *Direction* 1 вказує на позитивний напрямок, 3 – на негативний, а 4 – на поточний напрямок. Якщо значення *Direction* встановлено на 4 і вісь перебуває в стані STOP перед виконанням інструкції MC_MoveVelocity, вісь рухатиметься в позитивному напрямку.

Приклад програмування 1

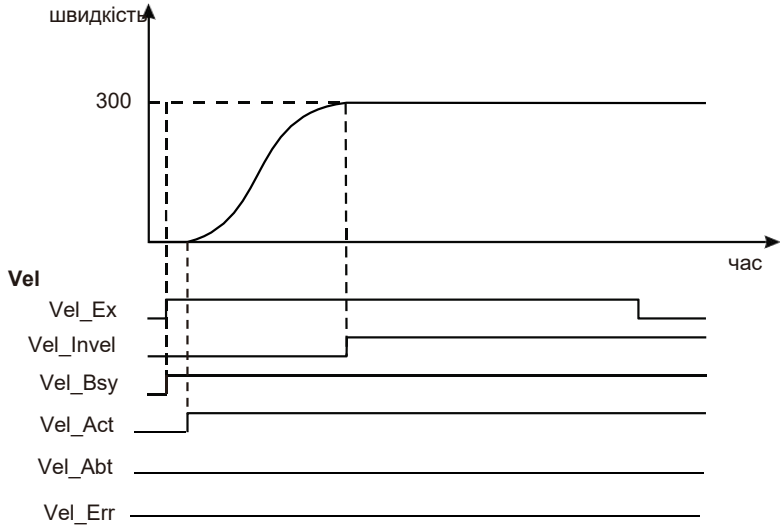
Приклад програмування такий, коли використовується одна інструкція MC_ MoveVelocity.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	


Pwr
MC_Power
1
Axis1
Axis
True
Можливий
True
Active позитивний
True
EnableNegative
Pwr_BM
BufferMode
Status
Busy
Active
Error
ErrorID
Pwr_Sta
Pwr_Bsy
Pwr_Act
Pwr_Err
Pwr_ErrID
Vel
MC_MoveVelocity
2
Axis1
Axis
Vel_Ex
Execute
Безперервна оновлення
300,0
швидкість
100,0
Прискорення
100,0
Уповільнення
15,0
Ривок
Vel_Dir
Напрямок
Vel_BM
BufferMode
Invelocity
зайнята
Active
CommandAborted
Error
ErrorID
Vel_Invel
Vel_Bsy
Vel_Act
Vel_Abt
Vel_Err
Vel_ErrID

2. Крива руху та часовий графік



- ❖ Коли Vel_Ex змінюється з FALSE на TRUE, Vel_Bsy змінюється на TRUE. Через один цикл Vel_Act змінюється на TRUE і починається виконання команди швидкості. Коли цільова швидкість досягнута, Vel_Invel змінюється на TRUE, а Vel_Bsy і Vel_Act залишаються TRUE.
- ❖ Коли Vel_Ex змінюється з TRUE на FALSE, Vel_Inve, Vel_Bsy і Vel_Act залишаються TRUE.

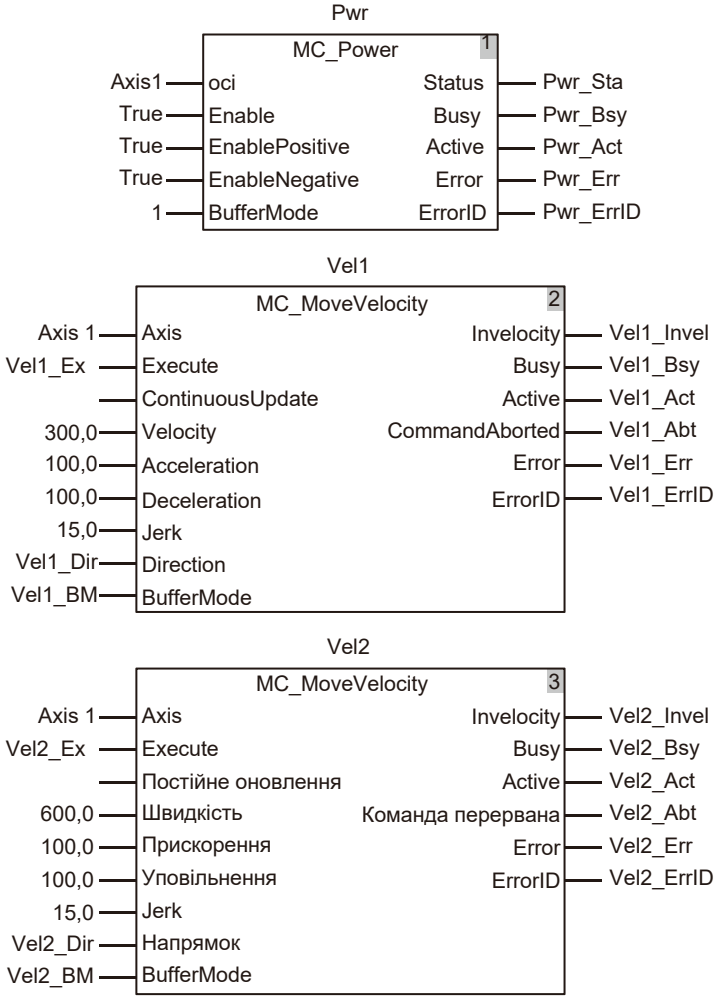
Приклад програмування 2

Нижче наведено приклад, коли одна інструкція MC_MoveVelocity перериває іншу інструкцію MC_MoveVelocity.

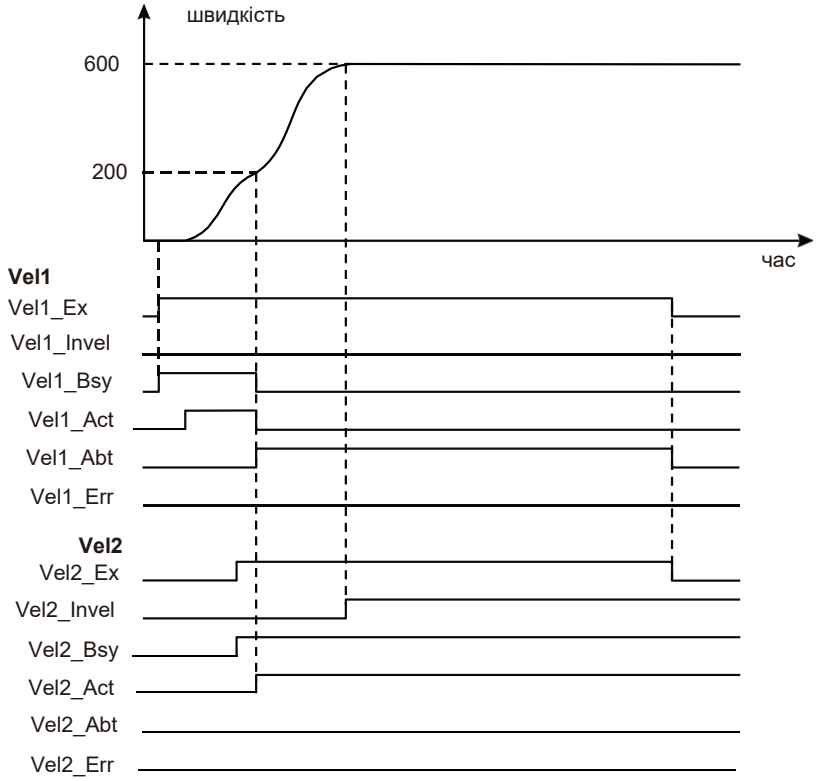
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	1
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel1	MC_MoveVelocity	
Vel1_Ex	BOOL	FALSE
Vel1_Dir	MC_DIRECTION	1
Vel1_BM	MC_Buffer_Mode	0
Vel1_Invel	BOOL	
Vel1_Bsy	BOOL	
Vel1_Act	BOOL	
Vel1_Abt	BOOL	
Vel1_Err	BOOL	
Vel1_ErrID	WORD	
Vel2	MC_MoveVelocity	
Vel2_Ex	BOOL	FALSE
Vel2_Dir	MC_DIRECTION	1
Vel2_BM	MC_Buffer_Mode	0

Ім'я змінної	Тип даних	Початкове значення
Vel2_Invel	BOOL	
Vel2_Bsy	BOOL	
Vel2_Act	BOOL	
Vel2_Abt	BOOL	
Vel2_Err	BOOL	
Vel2_ErrID	WORD	



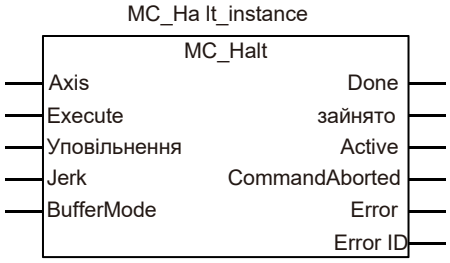
2. Крива руху та часовий графік



- Коли Vel1_Ex змінюється з FALSE на TRUE, Vel1_Bsy змінюється на TRUE. Через один цикл Vel1_Act змінюється на TRUE і починає виконуватися перша інструкція MC_MoveVelocity. Коли цільова швидкість не досягнута, Vel2_Ex змінюється з FALSE на TRUE, а Vel2_Bsy змінюється на TRUE. Через один цикл Vel2_Act змінюється на TRUE, перша інструкція MC_MoveVelocity переривається, Vel1_Abt змінюється на TRUE, і вісь починає виконувати другу інструкцію MC_MoveVelocity. Коли цільова швидкість досягнута, Vel2_Invel змінюється на TRUE, а Vel2_Bsy та Vel2_Act тим часом залишаються TRUE.
- Коли Vel1_Ex змінюється з TRUE на FALSE, Vel1_Abt змінюється на FALSE. Коли Vel2_Ex змінюється з TRUE на FALSE, Vel2_Invel, Vel2_Bsy і Vel2_Act залишаються TRUE.

11.3.4 MC_Halt

FB/FC	Пояснення	Застосовна модель
FB	MC_Halt використовується, щоб сповільнити вісь із заданою швидкістю уповільнення, поки вона не зупиниться.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано	MC_Buffer_Mode	0: mcAborting 1: mcBuffered (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. Інструкція MC_Halt виконується, коли *Execute* змінюється з FALSE на TRUE. Немає впливу на виконання інструкції, коли *Execute* інструкції змінюється з TRUE на FALSE під час виконання інструкції.
2. Поки *Execute* знову змінюється з FALSE на TRUE під час виконання MC_Halt, це не впливає на виконання інструкції, і інструкція продовжить виконуватися в попередньому режимі. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна виконати повторно.
3. Зверніться до розділу 10.2 щодо зв'язку між *уповільненням і ривком* .
4. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

● **Вихідні параметри**

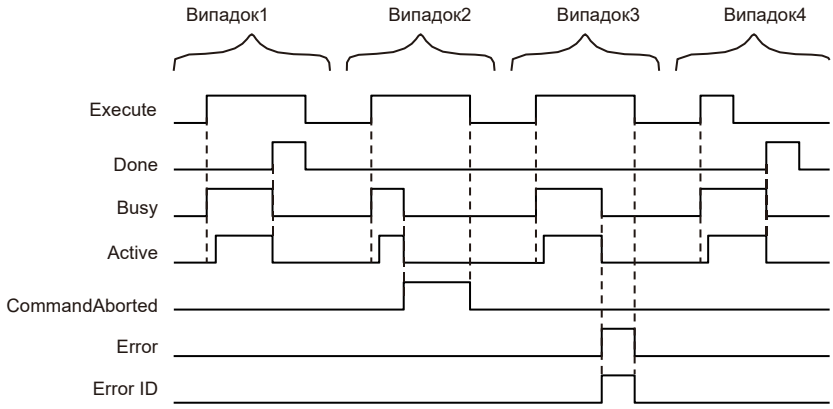
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE/FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE/FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE/FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	-

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли уповільнення закінчується, швидкість осі зменшується до 0.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли CommandAborted змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли CommandAborted змінюється на TRUE.
Команду скасовано	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ CommandAborted змінюється на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період CommandAborted змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

11

● Діаграма часу оновлення вихідних даних



Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли уповільнення закінчується та швидкість осі зменшується до 0, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE.

Випадок 2 : Після того, як *Execute* змінюється з FALSE на TRUE і інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : Коли виникає помилка, як-от аварійні сигнали осі або Offline after *Execute,* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE під час *виконання*
змінюється з TRUE на FALSE.

Випадок 4 : Під час виконання інструкції *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через період «Готово» змінюється на FALSE.

● функція

MC_Halt використовується, щоб сповільнити вісь із заданою швидкістю уповільнення, поки вона не зупиниться.

- Кінцевий автомат переходить у DiscreteMotion, коли MC_Halt починає виконуватися. Коли швидкість осі зменшується до 0, *Done* змінюється на TRUE, а тим часом кінцевий автомат переходить у стан зупинки.
- У порівнянні з інструкцією MC_Stop, інструкція MC_Halt не може заблокувати вісь, і, таким чином, контролер може виконувати інші інструкції щодо руху.
  MC_Halt можна перервати шляхом виконання іншої інструкції руху, коли вісь сповільнюється під час виконання MC_Halt. Інші інструкції руху можуть бути виконані контролером для перезапуску осі після завершення виконання MC_Halt і зупинки осі.

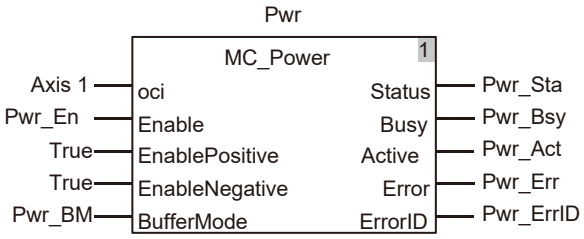
Приклад програмування

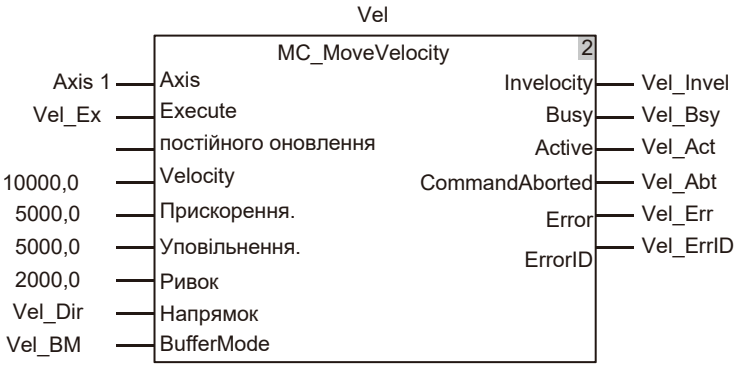
Приклад виконання MC_Halt наведено нижче.

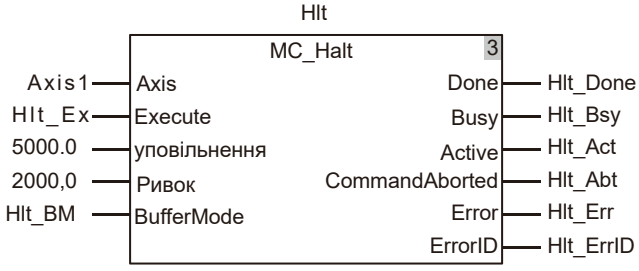
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	

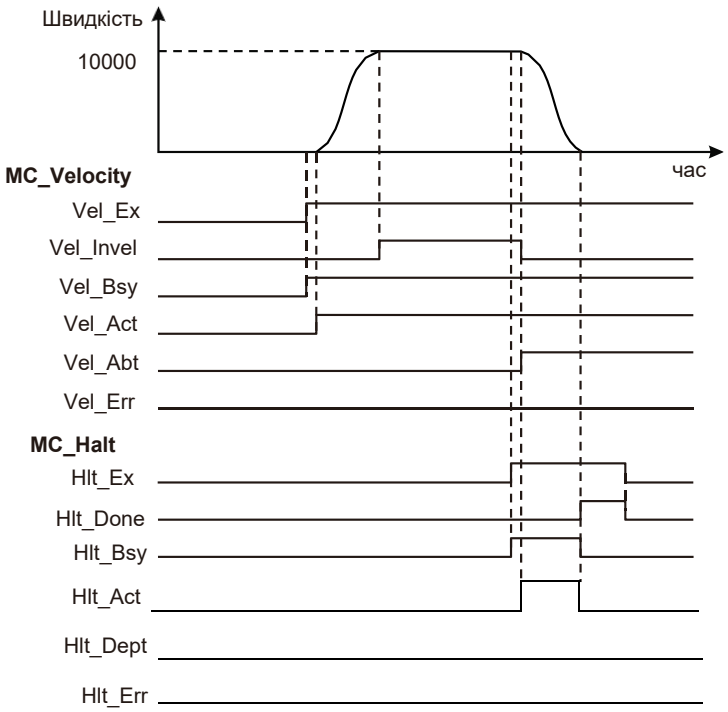
Ім'я змінної	Тип даних	Початкове значення
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Dept	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Ht	MC_Halt	
Hlt_Ex	BOOL	FALSE
Hlt_BM	MC_Buffer_Mode	0
Hlt_Done	BOOL	
Hlt_Bsy	BOOL	
Hlt_Act	BOOL	
Hlt_Dept	BOOL	
Hlt_Err	BOOL	
Hlt_ErrID	WORD	







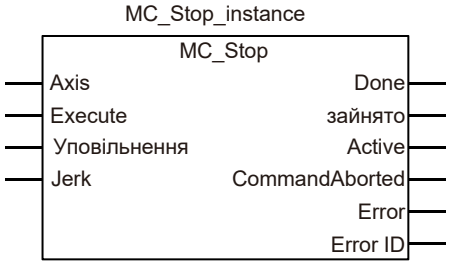
2. Крива руху та часові діаграми:



- Коли Vel_Ex змінюється на TRUE, Vel_Bsy змінюється на TRUE, а через період Vel_Act змінюється на TRUE, і серводвигун починає рухатися вперед. Vel_Invel змінюється на TRUE, коли серводвигун досягає цільової швидкості.
- Коли Hlt_Ex змінюється на TRUE, Hlt_Bsy змінюється на TRUE, а через період Hlt_Act змінюється на TRUE. Тим часом Vel_Invel змінюється на FALSE, а Vel_Abt змінюється на TRUE, після чого серводвигун починає гальмувати.
- Коли швидкість осі зменшується до 0, Hlt_Done змінюється на TRUE, а тим часом Hlt_Bsy і Hlt_Act змінюються на FALSE.
- Коли Hlt_Ex змінюється на FALSE, Hlt_Done змінюється на FALSE.

11.3.5 MC_Stop

FB/FC	Пояснення	Застосовна модель
FB	MC_Stop використовується, щоб вісь зменшувала свою швидкість із заданою швидкістю уповільнення до зупинки, а потім вісь переходить у стан зупинки.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. Інструкція MC_Stop виконується, коли *Execute* змінюється з FALSE на TRUE. Немає впливу на виконання інструкції, коли *Execute* інструкції змінюється з TRUE на FALSE під час виконання інструкції.
2. Поки *Execute* знову змінюється з FALSE на TRUE під час виконання MC_Halt, це не впливає на виконання інструкції, і інструкція продовжить виконуватися в попередньому режимі. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна виконати повторно.
3. Зверніться до розділу 10.2 щодо зв'язку між *уповільненням і ривком* .

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE/FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE/FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE/FALSE

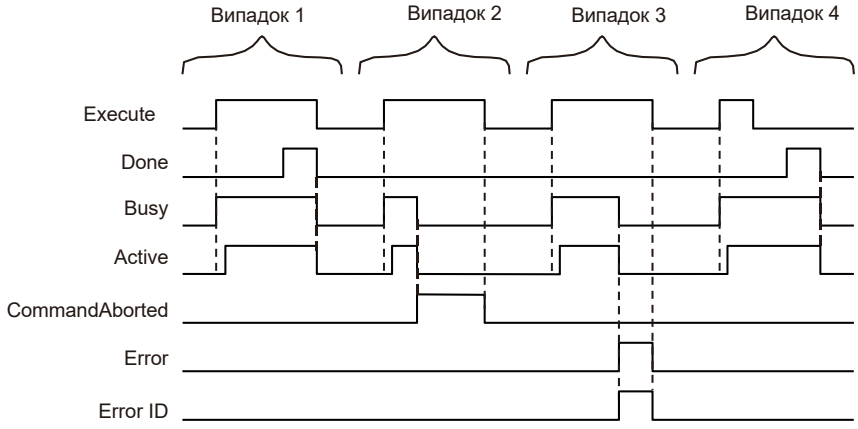
Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

11

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли уповільнення закінчується, швидкість осі зменшується до 0.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *Done* змінюється з TRUE на FALSE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *Done* змінюється з TRUE на FALSE.
CommandAborted	◆ Коли виконання інструкції переривається іншим MC_Stop.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* змінюється на TRUE, коли інструкцію переривається іншим MC_Stop після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● Діаграма часу оновлення вихідних даних



Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли уповільнення закінчується і швидкість осі зменшується до 0, *Done* змінюється на TRUE, а *Busy* та *Active* залишаються TRUE.

Випадок 2 : Коли інструкція MC_Stop переривається іншою інструкцією MC_Stop після того, як *Execute* змінюється з FALSE на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : Коли виникає помилка, наприклад аварійний сигнал осі або Offline after *Execute,* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. А тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : Під час виконання інструкції *Done* змінюється на TRUE, а *Busy* та *Active* залишаються TRUE, коли виконання інструкції завершено після зміни *Execute* з TRUE на FALSE. Через одну точку всі *параметри «Готово», «Зайнятий»* і *«Активний»* змінюються на FALSE.

● функція

MC_Stop використовується, щоб вісь зменшувала свою швидкість із заданою швидкістю уповільнення до зупинки.

- Поки *Execute* має значення TRUE після завершення виконання MC_Stop і швидкість осі зменшено до 0, стан осі буде весь час у стані зупинки. І протягом цього періоду інші інструкції руху не можуть бути виконані.
- Якщо в програмі є дві інструкції MC_Stop для керування однією віссю, попередньо виконувана інструкція MC_Stop буде перервана пізніше виконаною інструкцією MC_Stop.
- У порівнянні з інструкцією MC_Halt, інструкція MC_Stop зробить вісь заблокованою, і, таким чином, контролер не може виконувати інші інструкції руху, окрім MC_Stop, під час виконання MC_Stop. Контролер все ще не може виконувати інші інструкції щодо руху, коли виконання MC_Stop завершено, а вісь зупинено. Інші інструкції руху не можуть бути виконані, доки *Execute* of MC_Stop не зміниться з TRUE на FALSE.

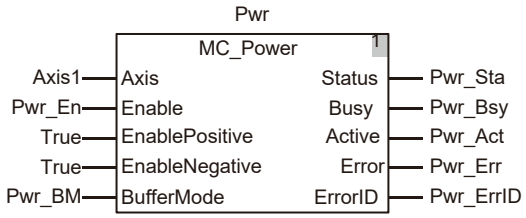
Приклад програмування 1

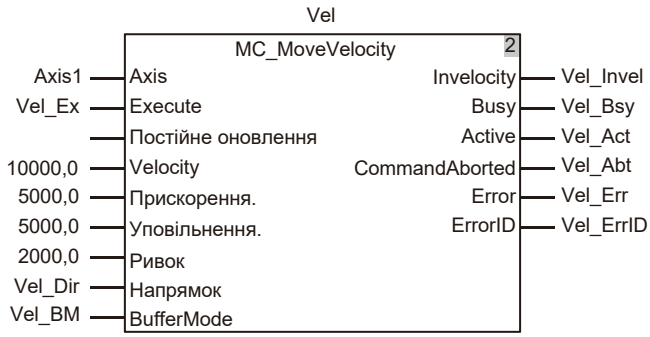
Приклад виконання MC_Stop показано нижче.

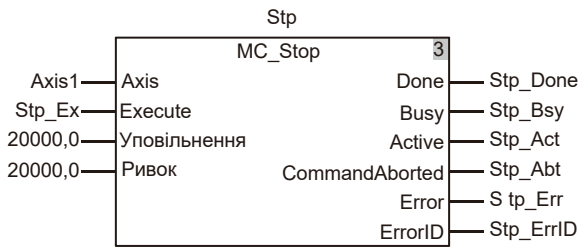
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0

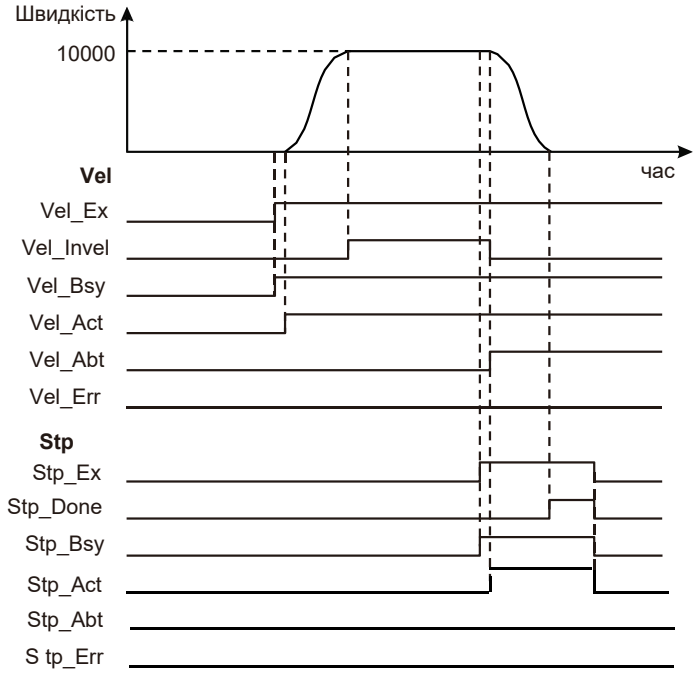
Ім'я змінної	Тип даних	Початкове значення
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
Stp	MC_Stop	
Stp_Ex	BOOL	FALSE
Stp_Done	BOOL	
Stp_Bsy	BOOL	
Stp_Act	BOOL	
Stp_Abt	BOOL	
Stp_Err	BOOL	
Stp_ErrID	WORD	







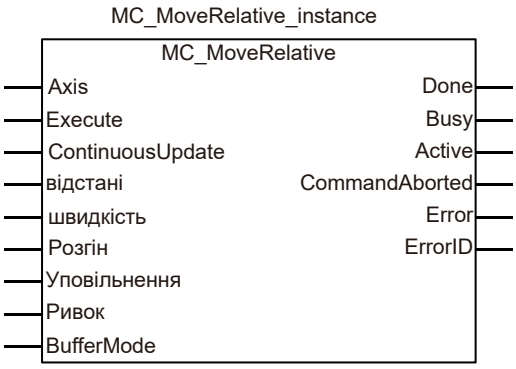
2. Крива руху та часові діаграми:



- ❖ Коли Vel_Ex змінюється на TRUE, Vel_Bsy змінюється на TRUE. Через один період Vel_Act змінюється на TRUE, і серводвигун починає рухатися вперед. Vel_Invel змінюється на TRUE, коли серводвигун досягає цільової швидкості.
- ❖ Коли Stp_Ex змінюється на TRUE, Stp_Bsy змінюється на TRUE. Через один період Stp_Act змінюється на TRUE, тим часом Vel_Invel змінюється на FALSE, Vel_Abt змінюється на TRUE, і серводвигун починає гальмувати.
- ❖ Коли швидкість осі зменшується до 0, Stp_Done змінюється на TRUE, а Stp_Bsy, Stp_Act тим часом залишаються TRUE.
- ❖ Оскільки Stp_Ex змінюється на FALSE, Stp_Done, Stp_Bsy та Stp_Act змінюються на FALSE одночасно.

11.3.6 MC_MoveRelative

FB/FC	Пояснення	Застосовна модель
FB	MC_MoveRelative використовується, щоб змусити вісь переміщатися на задану відстань, починаючи з поточної позиції команди з заданою швидкістю, прискоренням і уповільненням і ривком.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
Відстань	Вкажіть відстань руху від поточної позиції команди. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число або 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/секунда)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	MC_Buffer_Mode	● mcAborting ● mcBuffered ● mcBlendingLow ● mcBlendingsPrevious ● mcBlendingNext ● mcBlendingHigh (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_MoveRelative виконується, коли *Execute* змінюється з FALSE на TRUE. Немає впливу на виконання інструкції, коли *Execute* інструкції змінюється з TRUE на FALSE під час виконання.
2. Поки інструкція виконується і *Execute* знову змінюється з FALSE на TRUE, буде не впливає на виконання інструкції, і інструкція продовжуватиме виконуватися попереднім способом. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна повторно виконати та запустити звичайним способом.
3. Зверніться до розділу 10.2 щодо співвідношення між *швидкістю, прискоренням і ривком* .
4. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується інструкцією.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідних кодів помилок.	WORD	

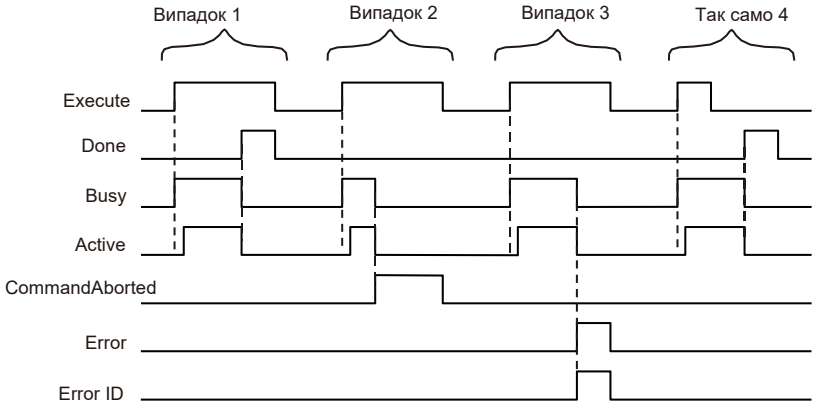
● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли позиціонування завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE.

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
		◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ *Зроблено* зміни на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

11

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE, а через один цикл *активний* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. *CommandAborted* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : коли виникає помилка, наприклад аварійний сигнал осі або Offline після того, як *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : під час виконання інструкції *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через один цикл *«Готово»* змінюється на FALSE.

- **функція**

MC_MoveRelative використовується для переміщення осі на задану відстань, починаючи з поточної позиції осі команди з заданою швидкістю, прискоренням, уповільненням і ривком.

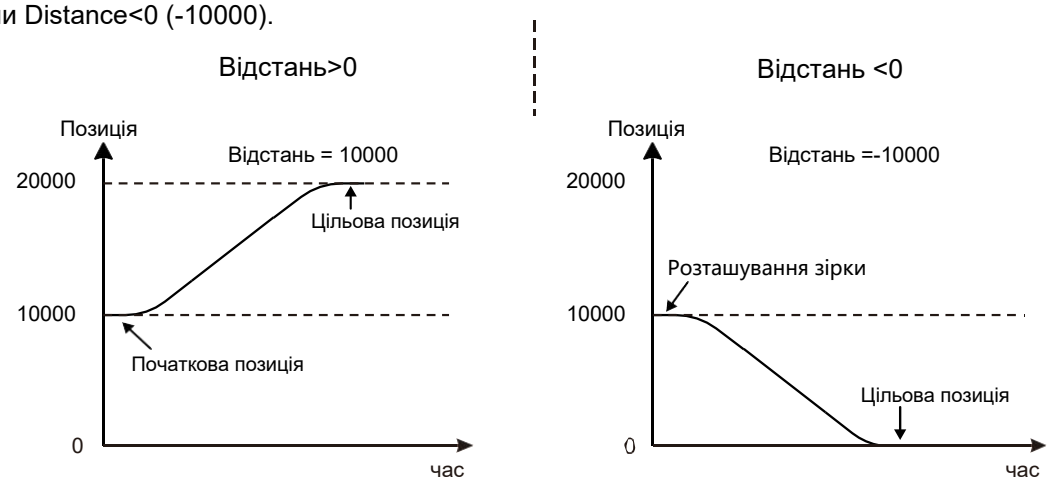
■ **Відстань**

Відстань і початкове положення для відліку разом визначають цільове положення, якого досягне вісь під керуванням інструкції. Цільове положення = початкове положення для відліку + ***відстань*** .

Якщо для *параметра «Дистанція»* встановлено значення 0, цільове положення для руху осі встановлюється як поточне положення. Виконання інструкції завершується в наступному циклі після її виконання, і *Done* змінюється на TRUE.

Як показано на наступному малюнку зліва, початкова позиція для порівняння становить 10 000. Вісь рухається в позитивному напрямку, а цільова позиція становить 20 000 (10 000+10 000), якщо відстань>0 (10 000).

На наступному правому малюнку вісь рухається в негативному напрямку, а цільове положення дорівнює 0 (10000-10000), коли Distance<0 (-10000).

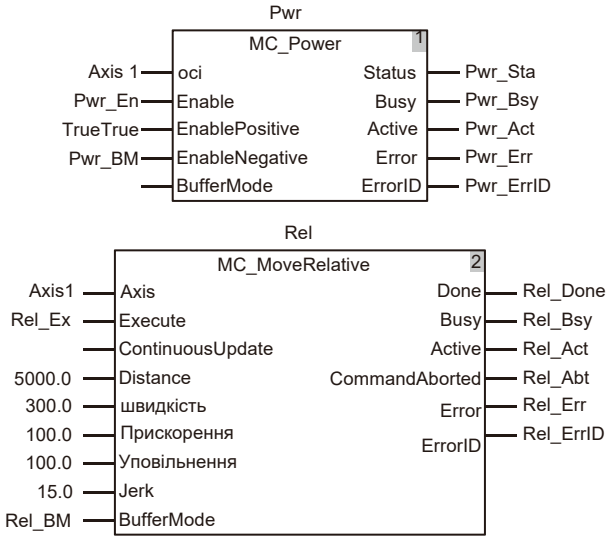


Приклад програмування 1

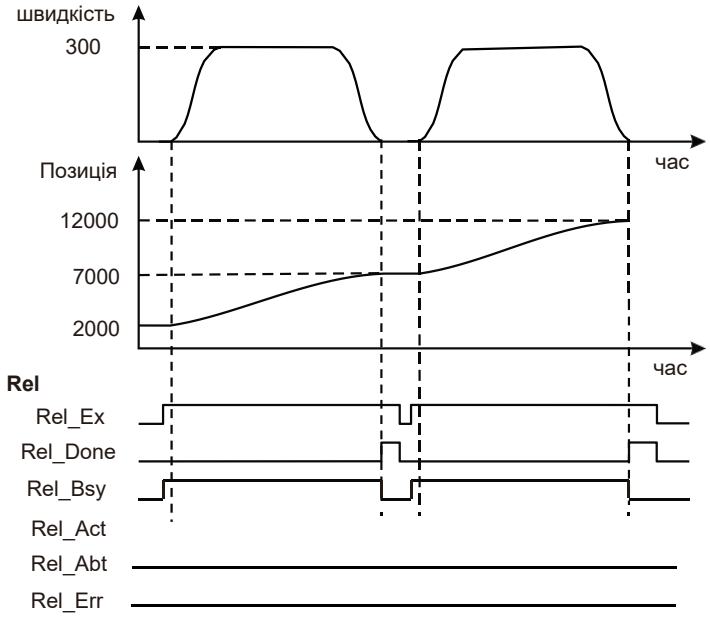
Приклад програмування такий, коли використовується одна інструкція MC_MoveRelative.

◆ **Змінні та програма**

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel _Ex	BOOL	FALSE
Rel _BM	MC_Buffer_Mode	0
Rel _Done	BOOL	
Rel _Bsy	BOOL	
Rel _Act	BOOL	
Rel _Abt	BOOL	
Rel _Err	BOOL	
Rel _ErrID	WORD	



◆ **Крива руху та часовий графік**



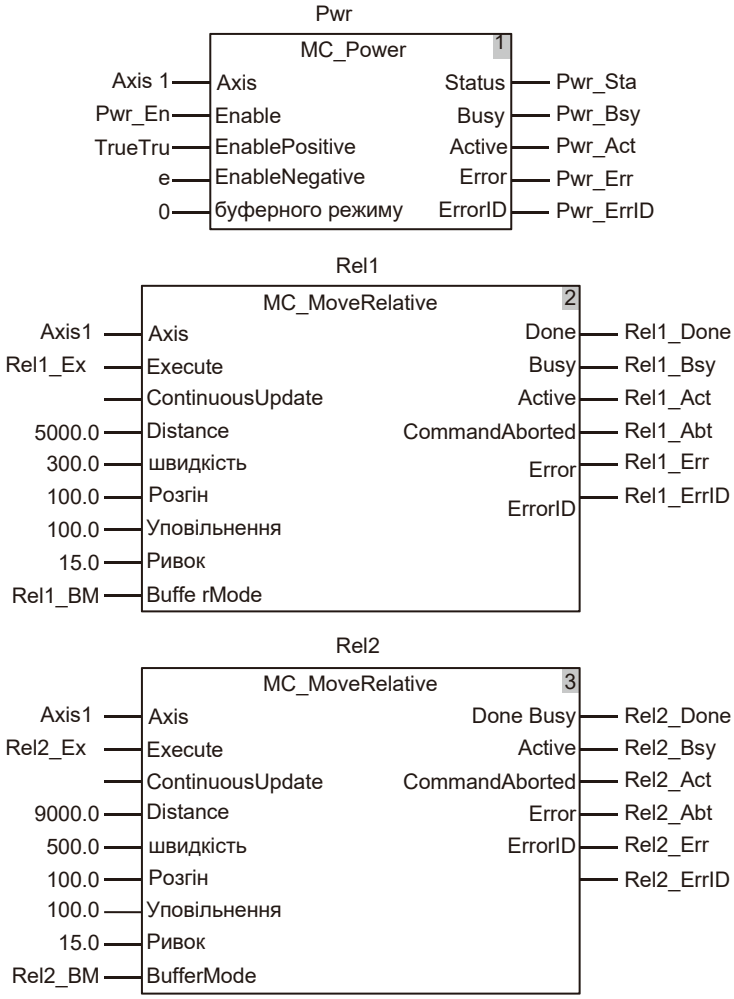
- ❖ Інструкція MC_MoveRelative виконується вперше, коли Rel_Ex змінюється з FALSE на TRUE вперше. На даний момент поточне положення осі – 2000, а цільове – 7000 (7000=2000+5000).
- ❖ Коли положення осі досягнуто 7000, виконання інструкції завершено і *виконано* змінюється на TRUE.
- ❖ Інструкція MC_MoveRelative починає виконання вдруге, коли Rel_Ex змінюється з FALSE на TRUE вдруге. На даний момент поточне положення осі становить 7000, а цільове положення – 12000 (12000=7000+5000).
- ❖ Коли положення осі досягнуто 12000, виконання інструкції вдруге завершується, а *значення Done* вдруге змінюється на TRUE.

Приклад програмування 2

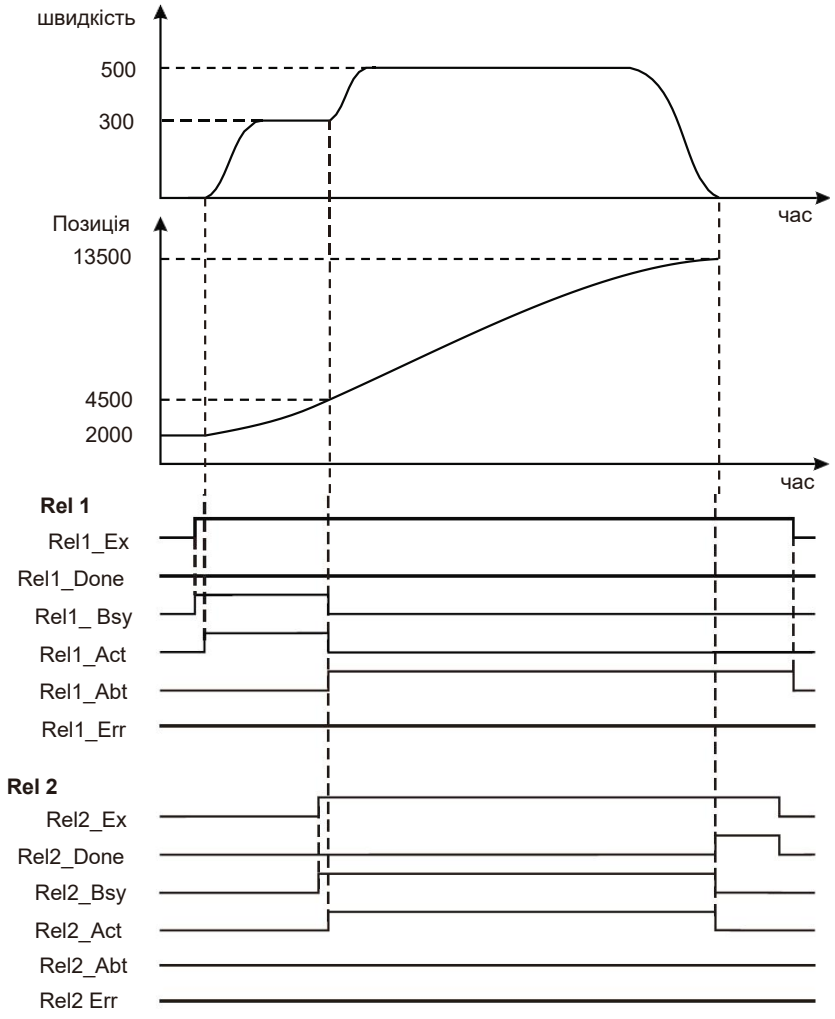
Нижче наведено приклад, коли MC_MoveRelative, який виконується, переривається.

1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1 _Ex	BOOL	FALSE
Rel1 _BM	MC_Buffer_Mode	0
Rel1 _Done	BOOL	
Rel1 _Bsy	BOOL	
Rel1 _Act	BOOL	
Rel1 _Abt	BOOL	
Rel1 _Err	BOOL	
Rel1 _ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2 _Ex	BOOL	FALSE
Rel2 _BM	MC_Buffer_Mode	0
Rel2 _Done	BOOL	
Rel2 _Bsy	BOOL	
Rel2 _Act	BOOL	
Rel2 _Abt	BOOL	
Rel2 _Err	BOOL	
Rel2 _ErrID	WORD	


Pwr
MC_Power
1
Axis 1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
TrueTru
EnablePositive
Active
Pwr_Act
e
EnableNegative
Error
Pwr_Err
0
буферного режиму
ErrorID
Pwr_ErrID
Rel1
MC_MoveRelative
2
Axis1
Axis
Done
Rel1_Done
Rel1_Ex
Execute
Busy
Rel1_Bsy
ContinuousUpdate
Active
Rel1_Act
5000.0
Distance
CommandAborted
Rel1_Abt
300.0
швидкість
Error
Rel1_Err
100.0
Розгін
ErrorID
Rel1_ErrID
100.0
Уповільнення
15.0
Ривок
Rel1_BM
Buffe rMode
Rel2
MC_MoveRelative
3
Axis1
Axis
Done Busy
Rel2_Done
Rel2_Ex
Execute
Active
Rel2_Bsy
ContinuousUpdate
CommandAborted
Rel2_Act
9000.0
Distance
Error
Rel2_Abt
500.0
швидкість
ErrorID
Rel2_Err
100.0
Розгін
Rel2_ErrID
100.0
Уповільнення
15.0
Ривок
Rel2_BM
BufferMode

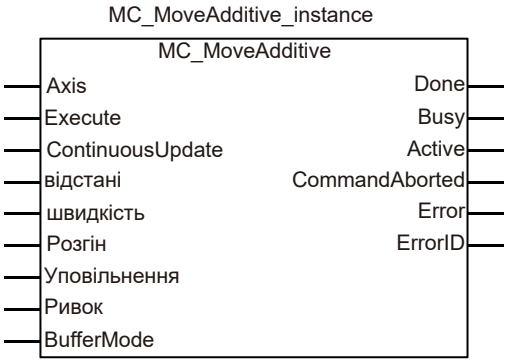
2. Крива руху та часовий графік



- Перша інструкція MC_MoveRelative починає виконуватися, коли Rel1_Ex змінюється з FALSE на TRUE. На даний момент поточне положення осі – 2000, а цільове – 7000 (7000=2000+5000).
- Коли досягається позиція осі 4500, Rel2_Ex змінюється з FALSE на TRUE, починає виконуватися друга інструкція MC_MoveRelative, а виконання першої інструкції MC_MoveRelative припиняється, а Rel1_Abt змінюється на TRUE.
- Коли досягається позиція осі 13500 (13500=4500+9000), виконання другої інструкції MC_MoveRelative завершується, а Rel2_Done змінюється на TRUE.

11.3.7 MC_MoveAdditive

FB/FC	Пояснення	Застосовна модель
FB	MC_MoveAdditive використовується для переміщення осі на адитивну відстань із заданою швидкістю, прискоренням і уповільненням.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
Відстань	Вкажіть адитивну відстань. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число або 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/с)	LREAL	Позитивне число або 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Вкажіть цільове прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
Уповільнення	Вкажіть цільове уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни прискорення та уповільнення цілі. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано 2: BlendingLow 3: Змішування попереднього 4: BlendingNext 5: Високий рівень змішування	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered 2 : mcBlendingLow 3 : mcBlendingПопередній нас 4 : mcBlendingNext 5 : mcBlendingHigh (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_MoveAdditive виконується, коли *Execute* змінюється з FALSE на TRUE. Немає впливу на виконання інструкції, коли *Execute* інструкції під час виконання змінюється з TRUE на FALSE.
2. Коли *Execute* інструкції, що виконується, знову змінюється з FALSE на TRUE, немає вплив на виконання інструкції, і інструкція продовжуватиме виконуватися попереднім способом. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна повторно виконати та запустити звичайним способом.
3. Зверніться до розділу 10.2 щодо співвідношення між *положенням, швидкістю, прискоренням і ривком* .
4. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode.*

● **Вихідні параметри**

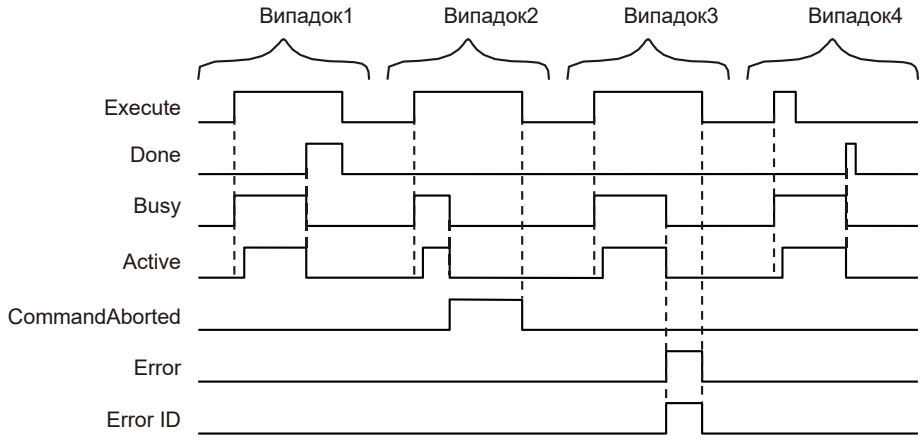
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли інструкція виконана.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли позиціонування добавки завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією керування рухом	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлено на TRUE, коли виконання інструкції переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE.

Випадок 2 : Коли *Execute* змінюється з FALSE на TRUE і виконання інструкції переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : Коли *Execute* змінюється з FALSE на TRUE і виникає помилка, наприклад аварійний сигнал осі або Offline, *Error* змінюється на TRUE, а *ErrorID* відображає відповідні коди помилок. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE під час *виконання* змінюється з TRUE на FALSE.

Випадок 4 : Під час виконання інструкції *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через період «Готово» змінюється на FALSE.

● **функція**

MC_MoveAdditive може керувати приводом для переміщення додаткової відстані із заданою швидкістю та прискоренням.

Виконання попередньої інструкції, пов’язаної з позиціонуванням, ще не завершено, і відстань, яку перемістить термінальний актуатор, включає незавершену відстань, залишену попередньою інструкцією, і задану відстань цієї інструкції під час виконання MC_MoveAdditive. Коли виконання MC_MoveAdditive завершено, кінцеве положення термінального приводу є сумою заданих відстаней попередньої інструкції та поточної інструкції MC_MoveAdditive.

Якщо попередня інструкція є інструкцією швидкості, MC_MoveAdditive перерве виконання інструкції швидкості, а виконавчий механізм терміналу зупиниться після переміщення заданої відстані MC_MoveAdditive із заданою швидкістю, прискоренням і уповільненням.

Якщо MC_MoveAdditive виконується під час окремого виконання MC_MoveSuperimposed, інструкція припинить виконання MC_MoveSuperimposed негайно, коли значення *BufferMode* MC_MoveAdditive стане 0. Відстань, на яку переміститься термінальний актуатор, включає встановлену відстань цієї інструкції та незавершену відстань, залишену MC_MoveSuperimposed під час виконання MC_MoveAdditive.
Якщо значення *BufferMode* знаходиться в діапазоні 1~5, в інструкції відразу виникне помилка, і виконання інструкції MC_MoveSuperimposed продовжиться.

Якщо MC_MoveAdditive виконується, коли MC_MoveSuperimposed використовується разом із інструкцією позиціонування, ця інструкція переривається MC_MoveSuperimposed та інструкцію позиціонування, коли значення *BufferMode* для MC_MoveAdditive дорівнює 0. Відстань, на яку переміститься виконавчий механізм терміналу, є сумою заданої відстані MC_MoveAdditive і незавершена відстань, залишена інструкцією позиції, яка використовується разом із MC_MoveSuperimposed, за винятком незавершеної відстані, залишеної MC_MoveSuperimposed під час виконання MC_MoveAdditive. Інструкція MC_MoveAdditive буде виконана після завершення виконання інструкції позиціонування, яка використовується в поєднанні з MC_MoveSuperimposed, якщо значення *BufferMode* для MC_MoveAdditive становить 1~5.

- **MC_MoveAdditive запускається під час виконання MC_MoveSuperimposed.**

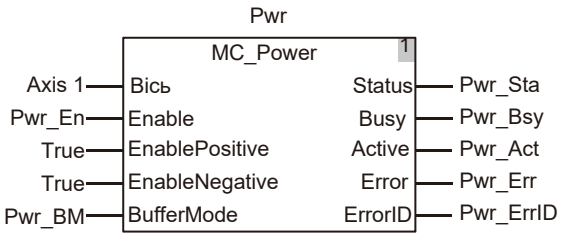
BufferMode MC_MoveAdditive	Чи виконується MC_MoveSuperimposed у поєднанні з іншою інструкцією позиціонування	опис
0 （ Скасувати ）	Так	◆ Виконання MC_MoveSuperimposed та іншої позиційної інструкції буде негайно перервано. ◆ Коли виконується MC_MoveAdditive, відстань, яку пройде термінальний актуатор, є встановленою відстанню MC_MoveAdditive плюс незавершена відстань, залишена MC_MoveSuperimposed, плюс незавершена відстань, залишена інструкцією позиції в поєднанні з MC_MoveSuperimposed.
	Ні	◆ MC_MoveSuperimposed негайно переривається. ◆ Коли виконується MC_MoveAdditive, виконавчий механізм терміналу пройде відстань, яка є сумою незавершеної відстані, залишеної MC_MoveSuperimposed, і встановленої відстані MC_MoveAdditive .
1~5 （ Буферизовано ）	Так	◆ MC_MoveSuperimposed не вплине і продовжує виконуватися. ◆ Після завершення виконання позиційної інструкції в поєднанні з MC_MoveSuperimposed запуститься MC_MoveAdditive.
	Ні	◆ Виконання MC_MoveSuperimposed продовжиться. ◆ MC_MoveAdditive негайно повідомить про помилку.

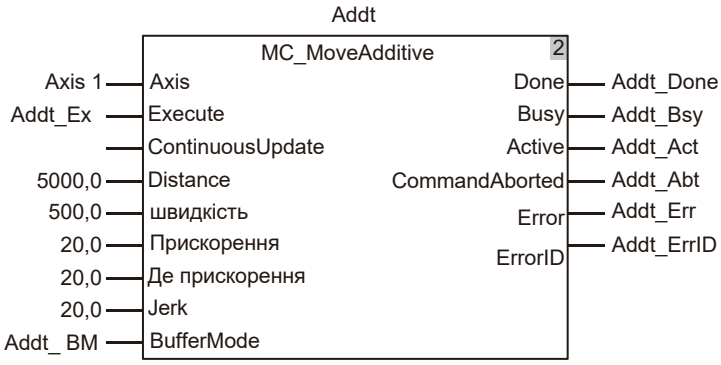
Приклад програмування 1

Нижче наведено приклад виконання однієї інструкції MC_MoveAbsolute.

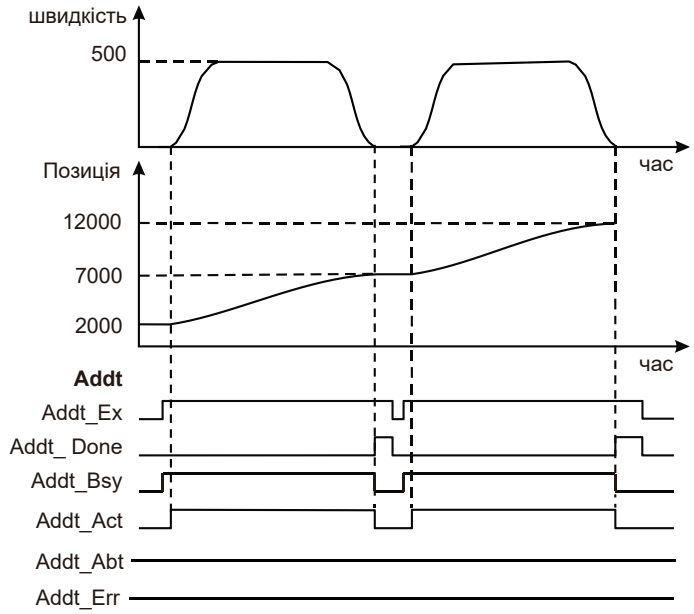
1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Addt	MC_MoveAdditive	
Addt_Ex	BOOL	FALSE
Addt_BM	MC_Buffer_Mode	0
Addt_Done	BOOL	
Addt_Bsy	BOOL	
Addt_Act	BOOL	
Addt_Abt	BOOL	
Addt_Err	BOOL	
Addt_ErrID	WORD	





2. Крива руху та часові діаграми:



- ❖ Коли Addt_Ex змінюється з FALSE на TRUE, контролер руху керує рухом серводвигуна, беручи поточне положення як точку відліку. Тим часом Addt_Bsy змінюється на TRUE, а через період Addt_Act змінюється на TRUE. Коли серводвигун досягає встановленої відстані, Addt_Done змінюється з FALSE на TRUE, а Addt_Bsy і Addt_Act тим часом змінюються з TRUE на FALSE.
- ❖ Коли Addt_Ex змінюється з TURE на FALSE, Addt_Done скидається.
- ❖ Коли Addt_Ex знову змінюється з FALSE на TRUE після того, як серводвигун досягає заданої відстані, контролер руху керує рухом серводвигуна, а Addt_Done знову змінюється з FALSE на TRUE після того, як серводвигун досягає встановленої відстані.

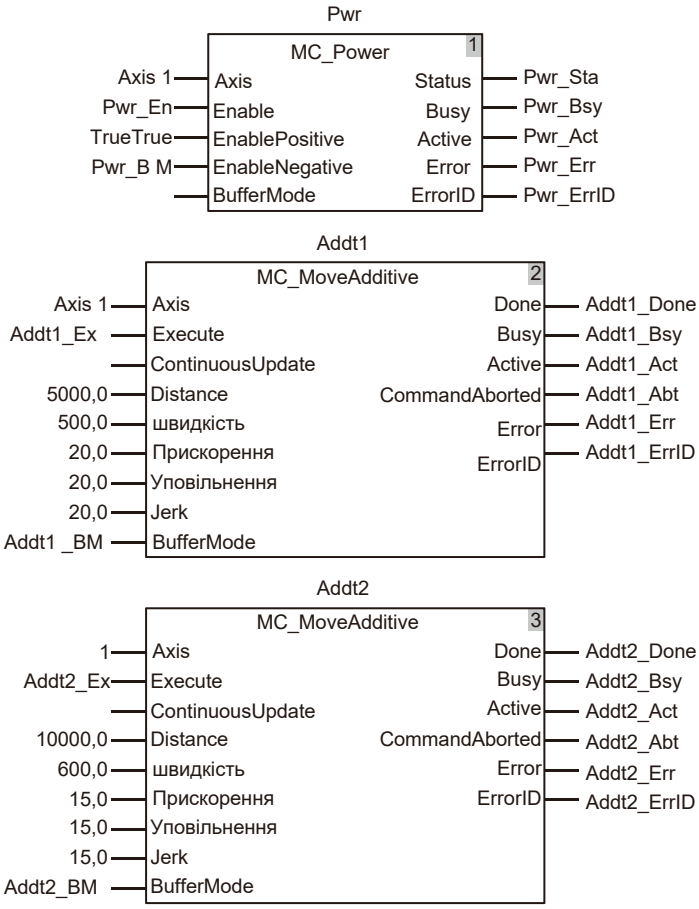
Приклад програмування 2

Нижче наведено приклад виконання двох інструкцій MC_MoveAdditive в одному списку завдань.

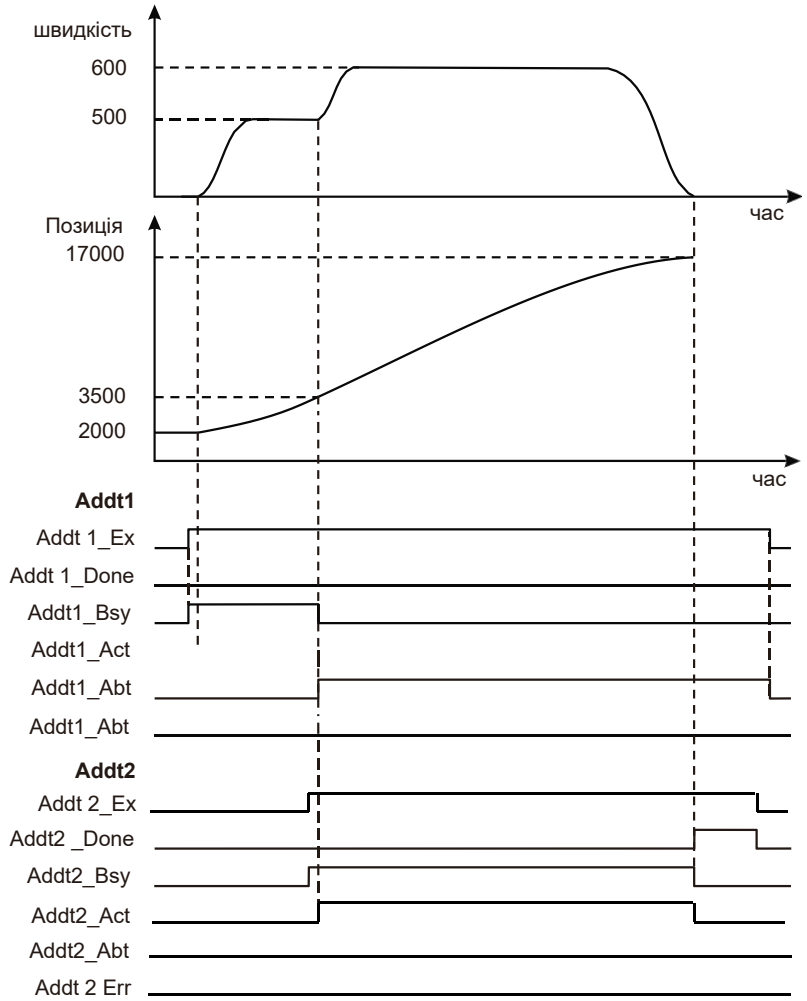
1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Addt1	MC_MoveAdditive	
Addt1_Ex	BOOL	FALSE
Addt1_BM	MC_Buffer_Mode	0
Addt1_Done	BOOL	
Addt1_Bsy	BOOL	
Addt1_Act	BOOL	
Addt1_Abt	BOOL	
Addt1_Err	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Addt1_ErrID	WORD	
Addt2	MC_MoveAdditive	
Addt2_Ex	BOOL	FALSE
Addt2_BM	MC_Buffer_Mode	0
Addt2_Done	BOOL	
Addt2_Bsy	BOOL	
Addt2_Act	BOOL	
Addt2_Abt	BOOL	
Addt2_Err	BOOL	
Addt2_ErrID	WORD	



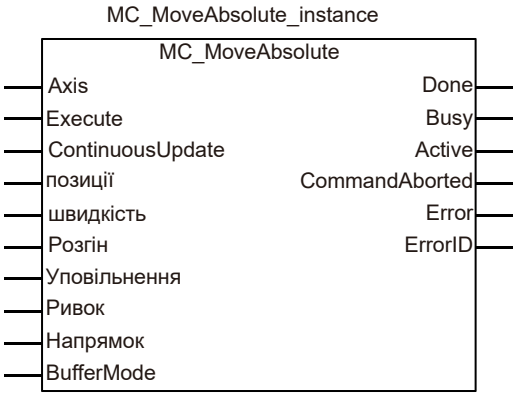
2. Крива руху та часові діаграми:



- Коли Addt1_Ex змінюється з FALSE на TRUE, контролер руху керує рухом серводвигуна, беручи поточне положення як точку відліку. Коли Addt2_Ex змінюється з FALSE на TRUE, Addt2_Bsy змінюється з FALSE на TRUE, а через один період перша інструкція MC_MoveAdditive переривається, а Addt1_Abt змінюється з FALSE на TRUE. Тим часом серводвигун рухається відповідно до параметрів другої інструкції MC_MoveAdditive. Addt2_Done змінюється з FALSE на TRUE, коли серводвигун завершує встановлену відстань, яка є загальною сумою двох встановлених відстаней двох інструкцій.
- Коли Addt2_Ex змінюється з TRUE на FALSE, Addt2_Done скидається.

11.3.8 MC_MoveAbsolute

FB/FC	Пояснення	FB/FC
FB	MC_MoveAbsolute використовується, щоб змусити вісь рухатися до вказаної абсолютної цільової позиції з заданою швидкістю, прискоренням і уповільненням.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
Позиція	Вкажіть абсолютну цільову позицію. Обертова вісь: 0≤ Позиція< Модуло Лінійна вісь: Позиція не обмежена. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число або 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/с)	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Вкажіть цільове прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Вкажіть цільове уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
ривок	Вкажіть швидкість зміни прискорення або уповільнення цілі. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
Напрямок	Укажіть напрямок обертання (це дійсне лише тоді, коли вісь є віссю обертання). 1: Позитивний напрямок 2: Найкоротший шлях 3: Негативний напрямок 4: Поточний напрямок	MC_Direction	1: mcPositiveDirection, 2: mcShortestWay, 3: mcNegativeDirection , 4: mcCurrentDirection (1)	Коли *Execute* змінюється з FALSE на TRUE і вісь знаходиться в режимі поворотної осі
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0 : McAborting 1 : McBuffered 2 : МакБлендінгЛоу 3 : McBlendingPrevious 4 : McBlendingNext 5 : McBlendingHigh	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered 2 : mcBlendingLow 3 : mcBlendingPrevious 4 : mcBlendingNext 5 : mcBlendingHigh (0)	Коли Execute змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_MoveAbsolute виконується, коли *Execute* змінюється з FALSE на TRUE. Немає впливу на виконання інструкції, коли *Execute* інструкції під час виконання змінюється з TRUE на FALSE.
2. Коли *Execute* інструкції, що виконується, знову змінюється з FALSE на TRUE, це не впливає на виконання інструкції, і інструкція продовжить виконуватися в попередньому порядку. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна виконати повторно.
3. Якщо вісь є віссю обертання, позиція може бути значенням у діапазоні від 0 до значення за модулем, за винятком значення за модулем. У інструкції виникне помилка, якщо абсолютне значення Position більше або дорівнює значенню modulo. Значення Position не має відношення до значення modulo, і його можна встановити як будь-яку константу, якщо вісь є лінійною віссю.
4. *Напрямок* дійсний лише тоді, коли вісь є віссю обертання. Щоб дізнатися більше про *напрямок,* зверніться до розділу «Напрямок» у наступному розділі «Функція» .
5. Зверніться до розділу 10.2 щодо співвідношення між *положенням, швидкістю, прискоренням і ривком* .
6. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode.*

● **Вихідні параметри**

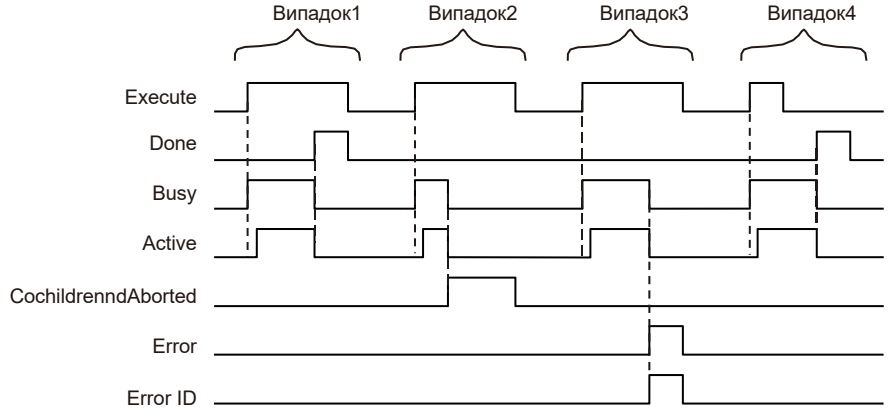
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується інструкцією.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка у виконанні інструкції.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідних кодів помилок.	WORD	

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли абсолютне позиціонування завершено	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлено на TRUE, коли виконання інструкції переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE, а тим часом Busy та Active змінюються на FALSE.

Випадок 2 : Коли виконання інструкції переривається іншою інструкцією руху після того, як *Execute* змінюється з FALSE на TRUE, *Abort* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : Коли *Execute* змінюється з FALSE на TRUE і виникає помилка, наприклад аварійний сигнал осі або Offline, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. А тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : Під час виконання інструкції *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через період «Готово» змінюється на FALSE.

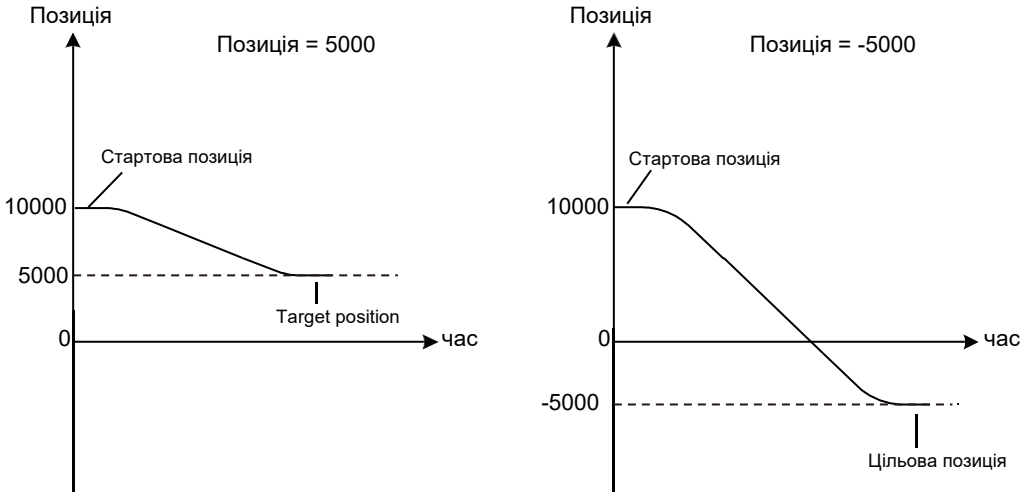
● **функція**

MC_MoveAbsolute використовується, щоб змусити вісь рухатися до вказаної абсолютної цільової позиції з заданою швидкістю, прискоренням і уповільненням.

Коли виконується інструкція MC_MoveAbsolute, початкове положення осі становить 10000. Вісь рухатиметься назад, коли *положення* >0 (5000). Дивіться малюнок нижче, коли *позиція* дорівнює 5000.

Вісь рухатиметься назад, коли *Позиція* <0 (-5000). Дивіться малюнок нижче, коли *Position* дорівнює -5000.

Позиція>0 | Позиція<0

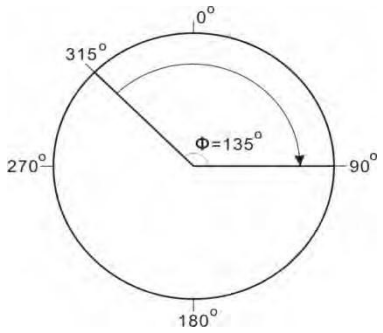
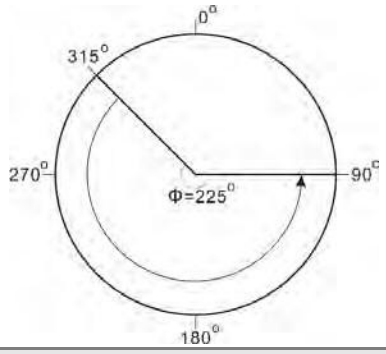
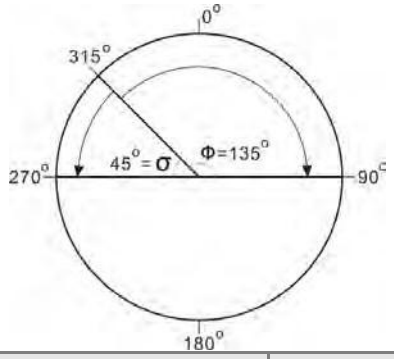
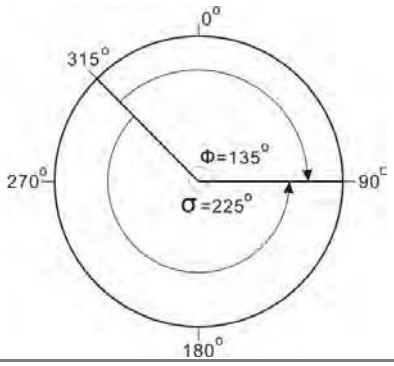


Примітка. Поки інструкція MC_MoveAbsolute, яка виконується, переривається, її незавершена відстань буде відкинута, а нова інструкція буде виконана.

■ **Напрямок**

Напрямок дійсний, якщо вісь є віссю обертання, а різні напрямки руху осі перераховані в наступній таблиці на основі різного значення напряму. (Модуль: 360)

11

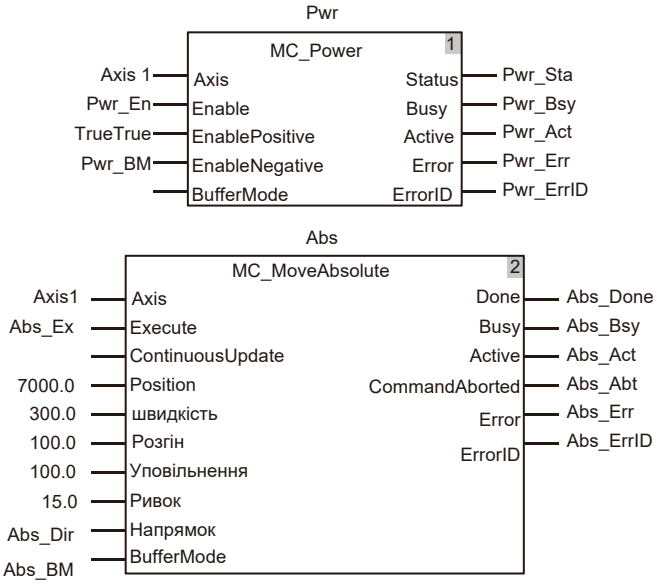
Напрямок: 1 (позитивний напрямок) Поточне положення: 315° Цільове положення: 90° Кут руху: 135°	Напрямок: 3 (Негативний напрямок) Поточне положення: 315° Цільове положення: 90° Кут руху: 225°
	
Напрямок: 2 (найкоротший шлях) Поточне положення: 315° Цільове положення: 90° Кут руху: 135°	Напрямок: 2 (найкоротший шлях) Поточне положення: 315° Цільове положення: 270° Кут руху: 45°
	
Напрямок: 4 (Поточний напрямок) Статус поворотної осі перед інструкцією виконано: Рух у негативному напрямку. Поточне положення: 315° Цільове положення: 90° Кут руху: 225°	Напрямок: 4 (Поточний напрямок) Стан поворотної осі перед вказівка виконується: нерухомо або рухаючись у позитивному напрямку. Поточне положення: 315° Цільове положення: 90° Кут руху: 135°
	

Приклад програмування 1

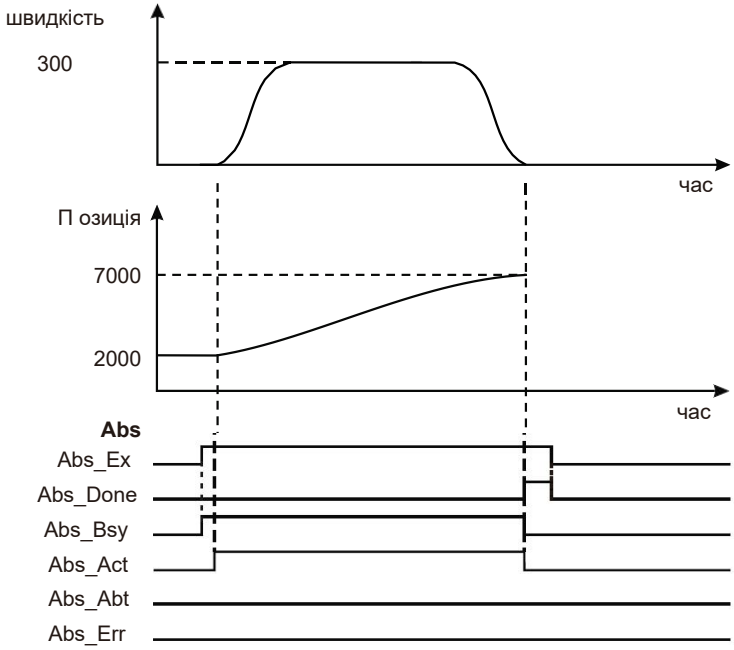
Один MC_MoveAbsolute виконується наступним чином.

1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Abs	MC_MoveAbsolute	
Abs_Ex	BOOL	FALSE
Abs_Dir	MC_DIRECTION	0
Abs_BM	MC_Buffer_Mode	0
Abs_Done	BOOL	
Abs_Bsy	BOOL	
Abs_Act	BOOL	
Abs_Abt	BOOL	
Abs_Err	BOOL	
Abs_ErrID	WORD	



2. Крива руху та часові діаграми



- Коли Abs_Ex змінюється з FALSE на TRUE, починає виконуватися інструкція MC_MoveAbsolute, і на даний момент поточна позиція осі становить 2000, а цільова позиція – 7000.
- Виконання інструкції завершується, коли вісь досягає 7000.

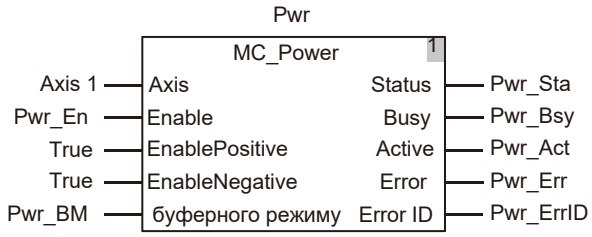
Приклад програмування 2

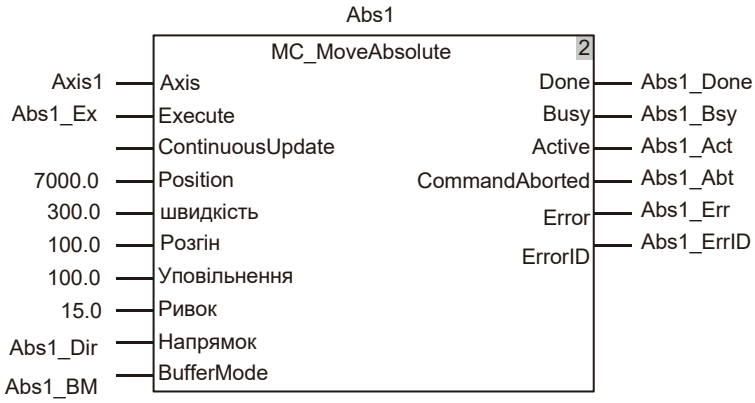
Нижче наведено приклад того, як одна інструкція MC_MoveAbsolute перериває виконання іншої інструкції MC_MoveAbsolute.

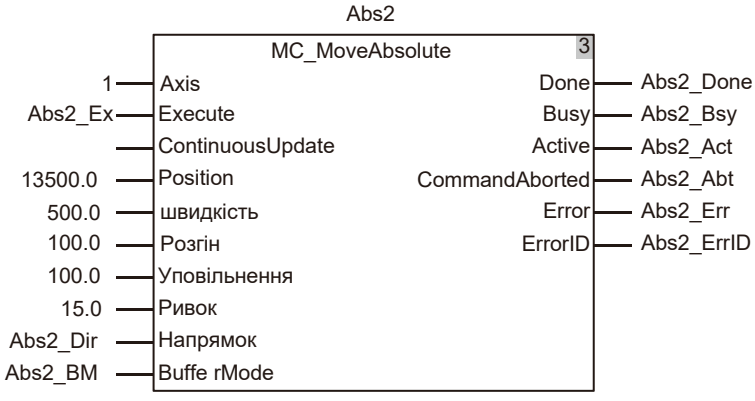
1. Змінні та програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Abs1	MC_Move Absolute	
Abs1_Ex	BOOL	FALSE
Abs1_Dir	MC_DIRECTION	0
Abs1_BM	MC_Buffer_Mode	0
Abs1_Done	BOOL	
Abs1_Bsy	BOOL	
Abs1_Act	BOOL	
Abs1_Abt	BOOL	
Abs1_Err	BOOL	

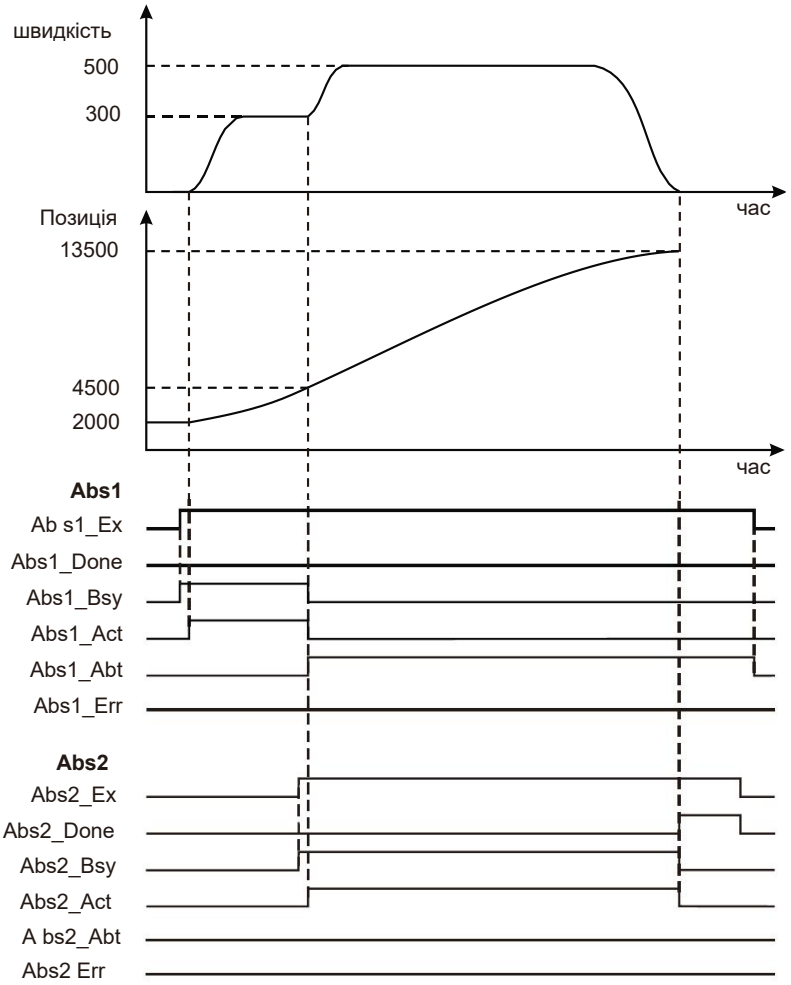
Ім'я змінної	Тип даних	Початкове значення
Abs1_ErrID	WORD	
Abs2	MC_Move Absolute	
Abs2_Ex	BOOL	FALSE
Abs2_Dir	MC_DIRECTION	0
Abs2_BM	MC_Buffer_Mode	0
Abs2_Done	BOOL	
Abs2_Bsy	BOOL	
Abs2_Act	BOOL	
Abs2_Abt	BOOL	
Abs2_Err	BOOL	
Abs2_ErrID	WORD	







2. Крива руху та часові діаграми



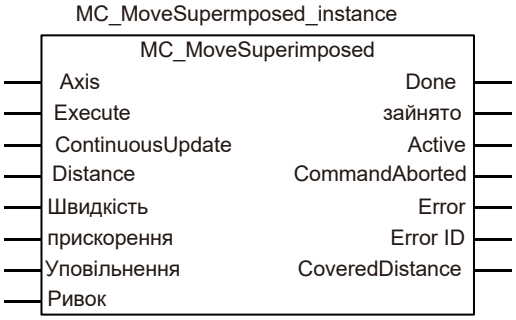
- Коли Abs1_Ex змінюється з FALSE на TRUE, починає виконуватися перша інструкція MC_MoveAbsolute, і на даний момент поточна позиція осі становить 2000, а цільова позиція – 7000.

- Коли вісь досягає 4500, Abs2_Ex змінюється з FALSE на TRUE; друга інструкція MC_MoveAbsolute починає виконуватися, а перша інструкція MC_MoveAbsolute переривається, а її вихідний параметр Abs1_Abt змінюється на TRUE.

- Коли вісь досягає 13500, виконання другої інструкції MC_MoveAbsolute завершується, а її вихідний параметр Abs2_Done змінюється на TRUE.

11.3.9 MC_MoveSuperimposed

FB/FC	Пояснення	Застосовна модель
FB	MC_MoveSuperimposed керує віссю, щоб накласти задану відстань на поточний стан руху відповідно до заданої швидкості, прискорення та уповільнення.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
Відстань	Відстань до накладання （ Одиниця: одиниця)	LREAL	Від'ємне число, додатне число і 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/секунда)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_MoveSuperimposed виконується, коли *Execute* змінюється з FALSE на TRUE. Немає жодного впливу на виконання інструкції, коли *Execute* of the instruction змінюється з TRUE на FALSE під час виконання інструкції.

2. Коли *Execute* знову змінюється з FALSE на TRUE під час виконання інструкції, це не впливає на виконання інструкції, і інструкція продовжить виконуватися в попередньому порядку. Коли *Execute* знову змінюється з FALSE на TRUE після завершення виконання інструкції, інструкцію можна виконати повторно.
3. Зверніться до розділу 10.2 щодо співвідношення між *швидкістю, прискоренням, уповільненням і ривком* .

● **Вихідні параметри**

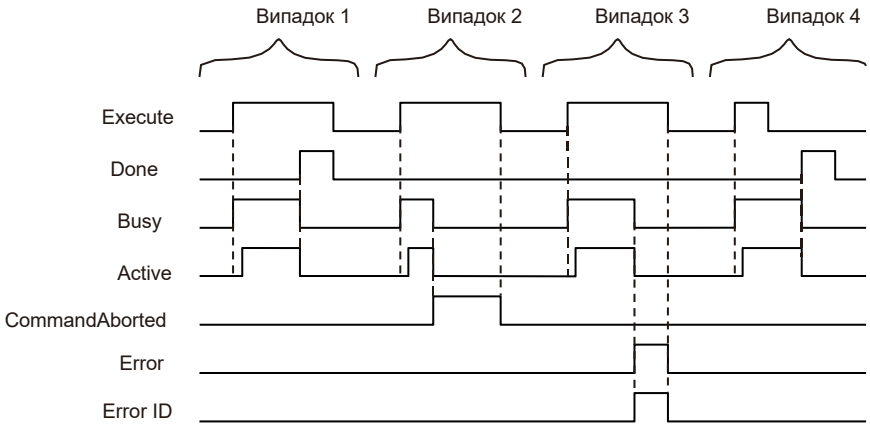
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	-
Подолана відстань	Повна накладена відстань з моменту запуску інструкції.	LREAL	Від'ємне число, додатне число і 0

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли накладене позиціонування завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один цикл *Active* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* та *Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється на TRUE, а інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. *CommandAborted* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : коли виникає помилка, наприклад, вимкнена вісь, оскільки *Execute* має значення TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, а значення *ErrorID* очищається на 0, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : *Зроблені* зміни змінюються на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через один цикл *«Готово»* змінюється на FALSE.

● функція

Інструкція MC_MoveSuperimposed керує віссю для незалежного накладання встановленої відстані на поточний стан руху відповідно до заданої швидкості, прискорення та уповільнення.

1. Коли виконується інструкція MC_MoveSuperimposed, виконання попередньої інструкції, за винятком інструкцій MC_MoveSuperimposed і MC_HaltSuperimposed, не переривається. Якщо обидві інструкції виконуються одночасно, їхні відстані, швидкості, прискорення та уповільнення відповідно складатимуться в реальному часі. Коли встановлена швидкість будь-якої з інструкцій буде досягнута, прискорення інструкції буде 0. Якщо виконання попередньої інструкції завершено, швидкості, прискорення та уповільнення більше не будуть складатися, а інструкція MC_MoveSuperimposed продовжує виконуватися незалежно.
2. Якщо інструкція MC_MoveSuperimposed виконується, коли вісь перебуває в стані зупинки, ефект виконання інструкції MC_MoveSuperimposed еквівалентний ефекту інструкції MC_MoveRelative.
3. Виконайте іншу інструкцію руху, за винятком інструкцій MC_MoveSuperimposed і MC_HaltSuperimposed, коли інструкція MC_MoveSuperimposed і одна інструкція руху спільно керують віссю. Якщо значення *Buffermode* останньої виконаної інструкції руху дорівнює 0, і інструкція MC_MoveSuperimposed, і попередньо виконана інструкція руху будуть перервані.

Якщо значення *Buffermode* останньої виконаної інструкції руху є іншим числом, крім 0, інструкція MC_MoveSuperimposed і раніше виконана інструкція руху не будуть перервані.

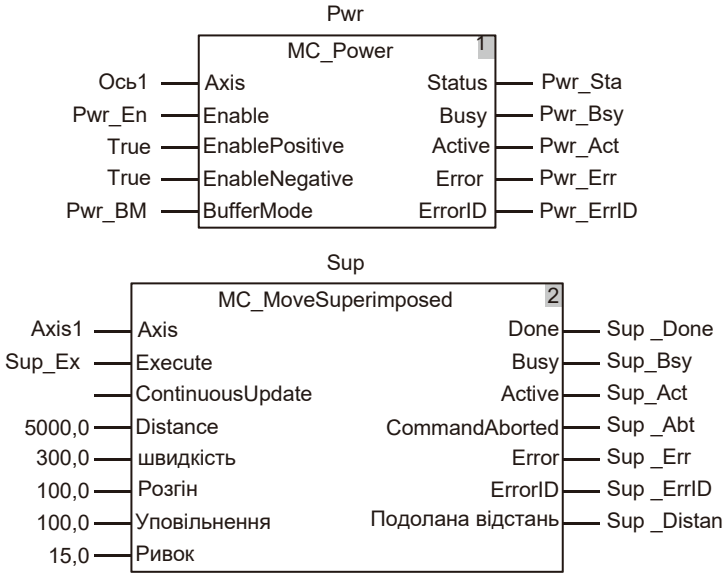
4. Якщо інша інструкція MC_MoveSuperimposed виконується, коли одна інструкція MC_MoveSuperimposed та інша інструкція руху спільно керують віссю, попередня інструкція MC_MoveSuperimposed буде перервана, але інша інструкція руху не вплине.
5. Якщо інша інструкція MC_MoveSuperimposed виконується, коли одна інструкція MC_MoveSuperimposed незалежно керує віссю, попередня інструкція MC_MoveSuperimposed буде перервана.
6. Якщо інструкція MC_HaltSuperimposed виконується під час виконання інструкції MC_MoveSuperimposed, інструкція MC_MoveSuperimposed буде перервана.
7. MC_MoveSuperimposed можна виконати на підлеглій осі, визначеній інструкцією MC_GearIn та інструкцією MC_ CamIn.

Приклад програмування 1

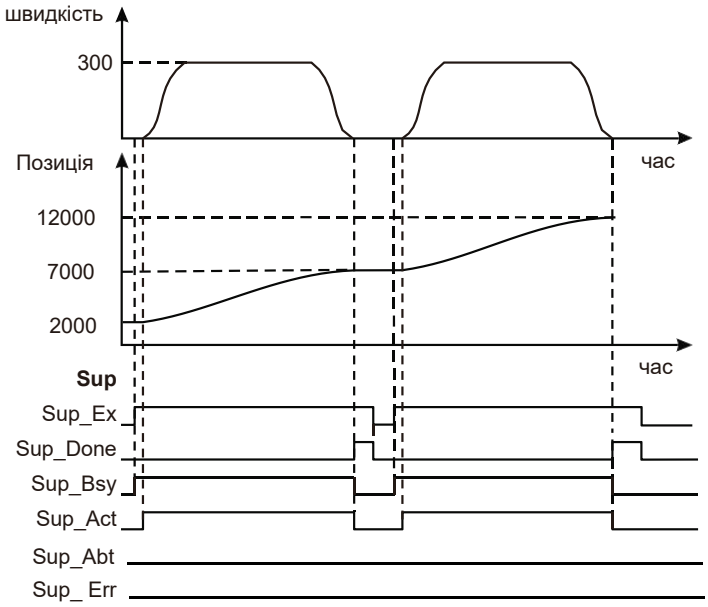
Приклад програмування такий, коли використовується одна інструкція MC_MoveSuperimposed.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	
Sup_ErrID	WORD	
Sup_Distan	LREAL	



2. Крива руху та часовий графік:



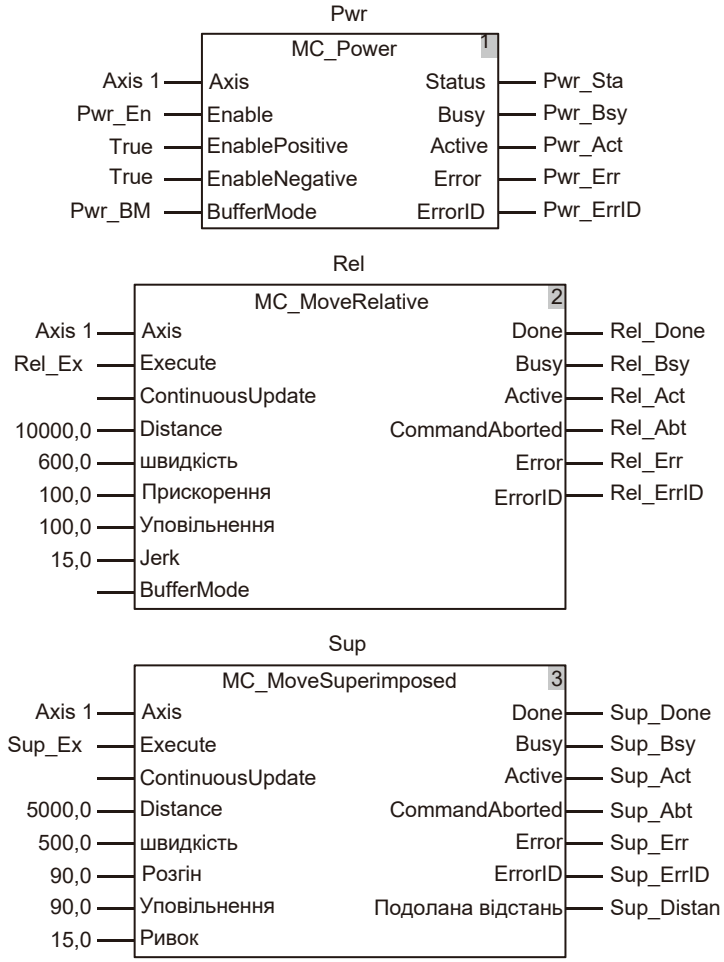
- Коли Sup_Ex змінюється на TRUE, Sup_Bsy змінюється на TRUE. Через один цикл Sup_Act змінюється на TRUE, і контролер руху керує роботою серводвигуна, використовуючи поточне положення як контрольну точку.
- Після того, як серводвигун завершить накладене позиціонування, Sup_Done змінюється на TRUE, а Sup_Bsy та Sup_Act тим часом змінюються на FALSE.
- Коли Sup_Ex змінюється на FALSE, Sup_Done змінюється на FALSE.
- Коли Sup_Ex знову змінюється на TRUE після того, як серводвигун пройде задану відстань, контролер руху керує роботою серводвигуна. Коли серводвигун завершує встановлену відстань, Sup_Done знову змінюється на TRUE.

Приклад програмування 2

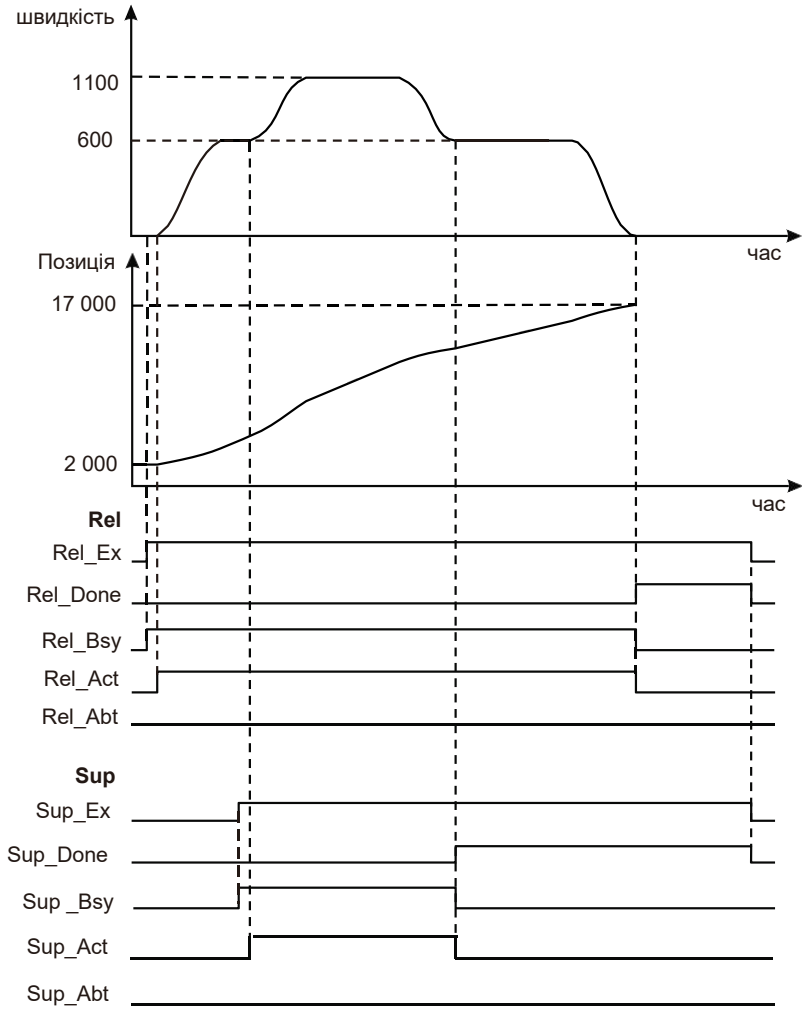
Нижче наведено приклад зіставлення інструкцій MC_MoveSuperimposed і MC_MoveRelative.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	
Sup_ErrID	WORD	
Sup_Distan	LREAL	


Pwr
MC_Power
1
Axis 1
Axis
Status
Pwr_Sta
Pwr_En
Enable
Busy
Pwr_Bsy
True
EnablePositive
Active
Pwr_Act
True
EnableNegative
Error
Pwr_Err
Pwr_BM
BufferMode
ErrorID
Pwr_ErrID
Rel
MC_MoveRelative
2
Axis 1
Axis
Done
Rel_Done
Rel_Ex
Execute
Busy
Rel_Bsy
ContinuousUpdate
Active
Rel_Act
10000,0
Distance
CommandAborted
Rel_Abt
600,0
швидкість
Error
Rel_Err
100,0
Прискорення
ErrorID
Rel_ErrID
100,0
Уповільнення
15,0
Jerk
BufferMode
Sup
MC_MoveSuperimposed
3
Axis 1
Axis
Done
Sup_Done
Sup_Ex
Execute
Busy
Sup_Bsy
ContinuousUpdate
Active
Sup_Act
5000,0
Distance
CommandAborted
Sup_Abt
500,0
швидкість
Error
Sup_Err
90,0
Розгін
ErrorID
Sup_ErrID
90,0
Уповільнення
Подолана відстань
Sup_Distan
15,0
Ривок

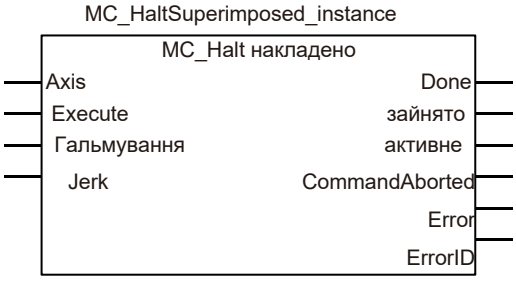
2. Крива руху та часовий графік:



- Коли Rel_Ex змінюється на TRUE, Rel_Bsy змінюється на TRUE. Через один період Rel_Act змінюється на TRUE, і контролер руху керує обертанням сервомотора, використовуючи поточне положення як контрольну точку.
- Коли Sup_Ex змінюється на TRUE, Sup_Bsy змінюється на TRUE. Через один цикл Sup_Act змінюється на TRUE, а інструкція MC_MoveSuperimposed починає керувати віссю. Швидкість і прискорення (0 на даний момент) для сервомотора є сумою швидкостей і прискорень двох інструкцій відповідно.
- Коли накладена відстань, визначена інструкцією MC_MoveSuperimposed, завершена, Sup_Done змінюється на TRUE, а Sup_Bsy і Sup_Act змінюються на FALSE.
- Коли відстань, визначену інструкцією MC_MoveRelative, завершена, Rel_Done змінюється на TRUE, а Rel_Bsy і Rel_Act змінюються на FALSE. Кінцеве положення осі є сумою відстаней двох інструкцій плюс початкове положення.
- Коли Rel_Ex змінюється на FALSE, Rel_Done змінюється на FALSE. Коли Sup_Ex змінюється на FALSE, Sup_Done змінюється на FALSE.

11.3.10 MC_HaltSuperimposed

FB/FC	Пояснення	Застосовна модель
FB	MC_HaltSuperimposed зупиняє виконання інструкції MC_MoveSuperimposed.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни цільового уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Інструкція MC_HaltSuperimposed виконується, коли *Execute* змінюється з FALSE на TRUE. Немає жодного впливу на виконання інструкції, коли *Execute* of the instruction змінюється з TRUE на FALSE під час виконання інструкції.
2. Зверніться до розділу 10.2 щодо зв'язку між *уповільненням* і *ривком* .

● **Вихідні параметри**

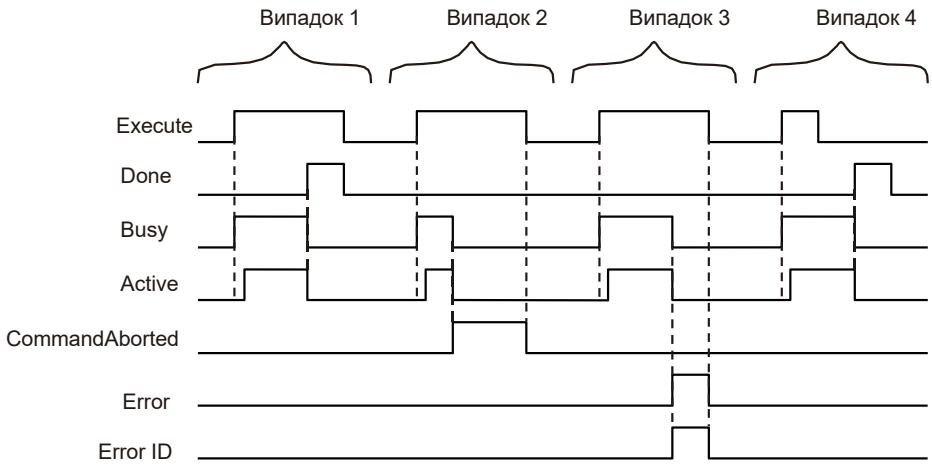
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	-

11

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один цикл *Active* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* і *Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється на TRUE, а інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. *CommandAborted* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : коли виникає помилка, наприклад, вісь вимкнено, оскільки *«Execute»* має значення «TRUE», *помилка* змінюється на «TRUE» і *ErrorID* показує відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : *Зроблені* зміни на TRUE, коли виконання інструкції завершено після зміни *Execute* з TRUE на FALSE під час виконання інструкції. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через один цикл *«Готово»* змінюється на FALSE.

● функція

Інструкція MC_HaltSuperimposed використовується для припинення виконання інструкції MC_MoveSuperimposed.

- Інструкція MC_HaltSuperimposed не може бути виконана окремо, і її можна використовувати лише разом з інструкцією MC_MoveSuperimposed.
- Якщо інструкція MC_HaltSuperimposed виконується, коли інструкція MC_MoveSuperimposed та інша інструкція руху спільно керують віссю, інструкція MC_HaltSuperimposed перерве інструкцію MC_MoveSuperimposed, але це не вплине на виконання інших інструкцій руху.
- Інструкція MC_HaltSuperimposed може призупинити виконання інструкції MC_HaltSuperimposed.

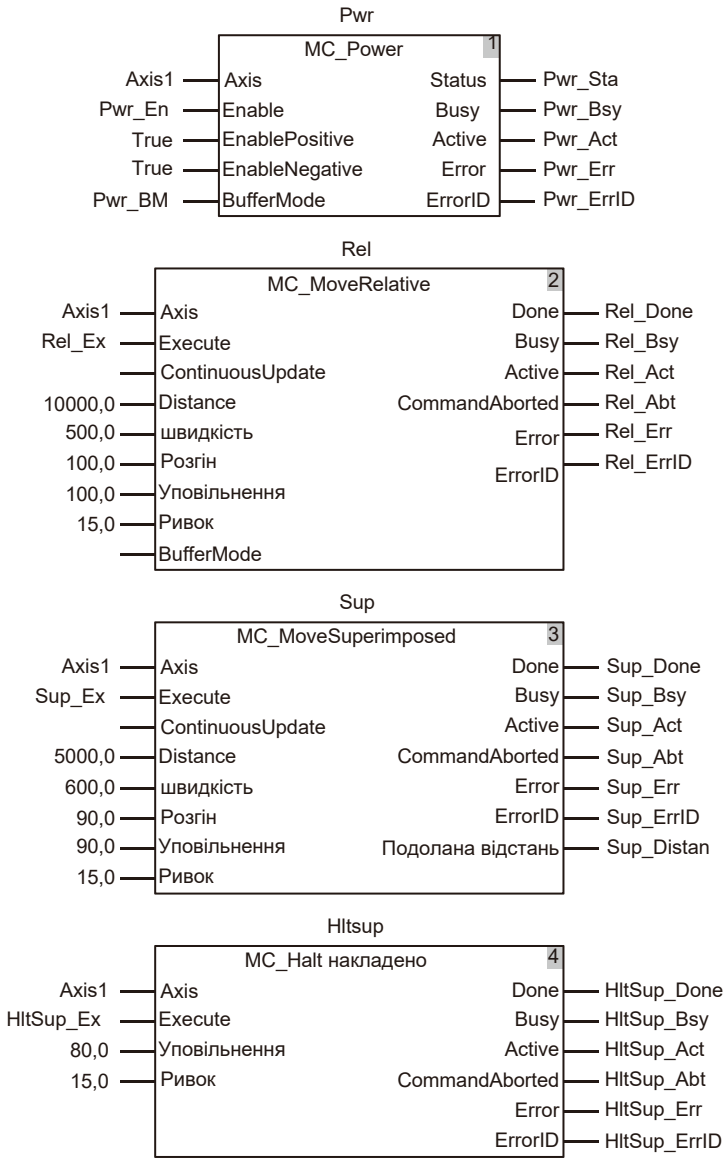
Приклад програмування

Приклад програмування такий, коли використовується одна інструкція MC_HaltSuperimposed.

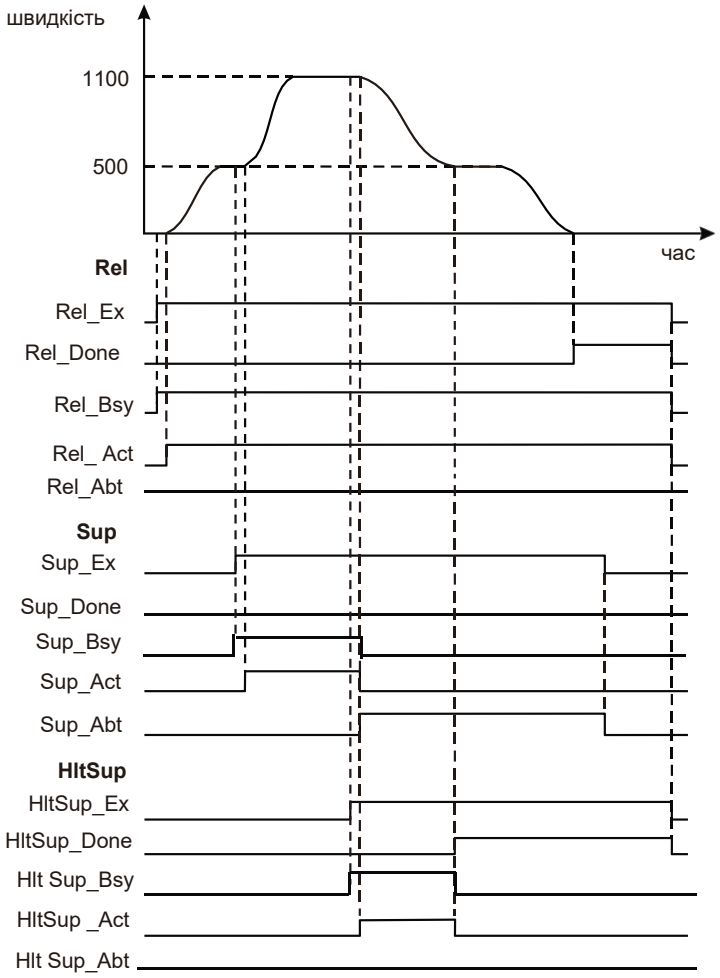
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Sup	MC_MoveSuperimposed	
Sup_Ex	BOOL	FALSE
Sup_Done	BOOL	
Sup_Bsy	BOOL	
Sup_Act	BOOL	
Sup_Abt	BOOL	
Sup_Err	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Sup_ErrID	WORD	
Sup_Distan	LREAL	
HltSup	MC_HaltSuperimposed	
HltSup_Ex	BOOL	FALSE
HltSup_Done	BOOL	
HltSup_Bsy	BOOL	
HltSup_Act	BOOL	
HltSup_Abt	BOOL	
HltSup_Err	BOOL	
HltSup_ErrID	WORD	



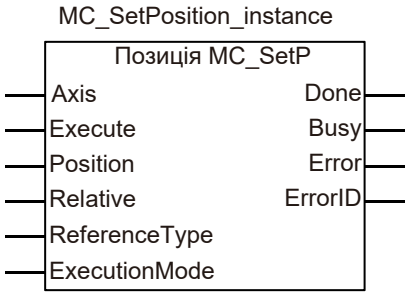
2. Крива руху та часовий графік



- Коли Rel_Ex змінюється на TRUE, Rel_Bsy змінюється на TRUE. Через один цикл Rel_Act змінюється на TRUE, і контролер руху керує обертанням серводвигуна, використовуючи поточне положення як контрольну точку. Коли Sup_Ex змінюється на TRUE, Sup_Bsy змінюється на TRUE. Через один цикл Sup_Act змінюється на TRUE, починається виконання інструкції MC_MoveSuperimposed, а швидкості та прискорення (0 на даний момент) для серводвигуна будуть додані відповідно.
- Коли Hltsup_Ex змінюється на TRUE, Hltsup_Bsy змінюється на TRUE. Через один цикл Hltsup_Act змінюється на TRUE, починається виконання інструкції MC_HaltSuperimposed, інструкція MC_MoveSuperimposed переривається, а Sup_Bsy та Sup_Act змінюються на FALSE, а Sup_Abt тим часом змінюється на TRUE. Виконання інструкції MC_MoveSuperimposed зупиняється інструкцією MC_HaltSuperimposed.
- Коли Hltsup_Done змінюється на TRUE, Hltsup_Bsy і Hltsup_Act змінюються на FALSE.
- Виконання інструкції MC_HaltSuperimposed не впливає на інструкцію MC_MoveRelative, що виконується.

11.3.11 MC_SetPosition

FB/FC	Пояснення	Застосовна модель
FB	MC_SetPosition використовується для встановлення позиції осі на задане значення, і фактичний рух осі не виконується відповідно.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
Позиція	Вкажіть цільову позицію. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число або 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
Відносна	Укажіть відносний режим або абсолютний режим для цільової позиції та поточної позиції.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
ReferenceType	Вкажіть тип позиції для довідки.	MC_ ReferenceType	0: позиція mcCommand 1: Фактична позиція mc (0)	Коли *Execute* змінюється з FALSE на TRUE
Режим виконання	Зарезервований			

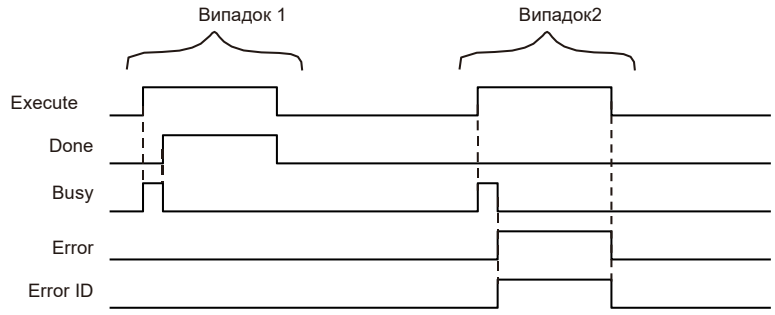
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли інструкція виконана.	BOOL	TRUE / FALSE
Busy	TRUE під час виконання інструкції.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли позиціонування завершено	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**

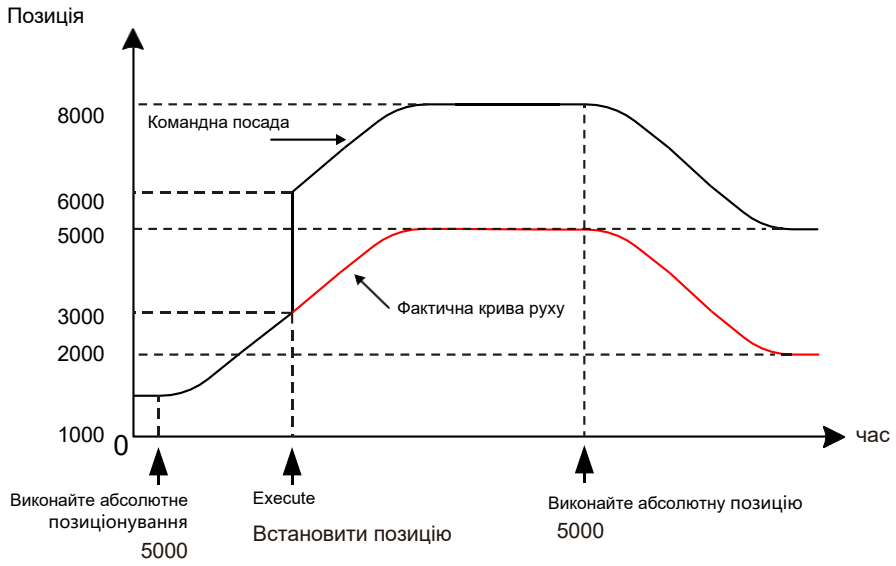


Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через одну точку *Done* змінюється на TRUE, а тим часом *Busy* змінюється на FALSE.

Випадок 2 : Коли виникає помилка, коли *Execute* має значення TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. А тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

● **функція**

MC_SetPosition використовується для встановлення позиції осі на задане значення, і фактичний рух осі не відбувається. Виконання MC_SetPosition не впливає на поточний рух. Однак це впливає на фактичний ефект виконання інструкції, яка виконується після завершення виконання MC_SetPosition.

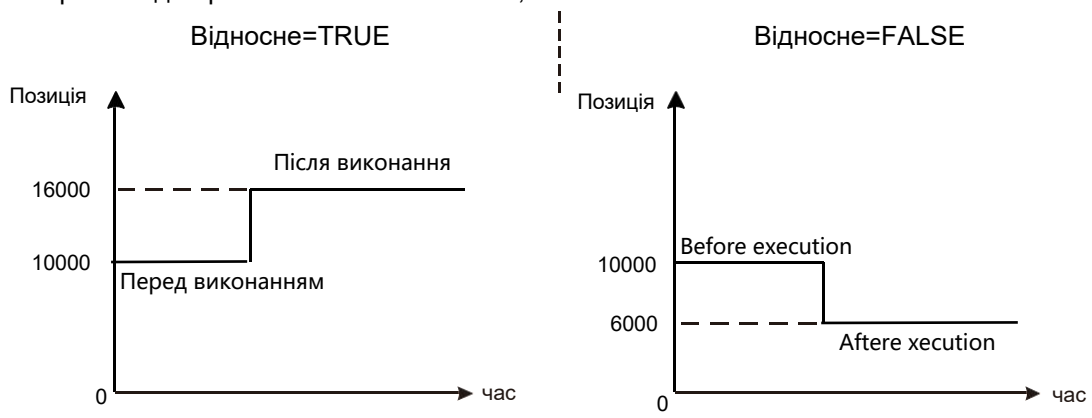


■ Зв'язок між *позицією* та *відносним*

Позиція, відносна та еталонна позиція, яка відповідає положенню осі в момент початку виконання інструкції, разом визначають значення налаштування позиції.

Відносне використовується для визначення зв'язку між *позицією* та опорною позицією. Якщо для *параметра Relative* встановлено значення TRUE, це відносний зв'язок між *позицією* та опорною позицією, а також значення параметра позиції = опорна позиція + *позиція.* Якщо *Relative* має значення FALSE, це абсолютний зв'язок між *Position* і опорним положенням, а значення налаштування положення дорівнює *Position.*

Як показано на наступних малюнках, опорна позиція встановлена на 10000, а значення *Position* становить 6000 для виконання інструкції. Відповідні результати виконання відповідно проілюстровано для різних *відносних* значень, як показано нижче.



■ ReferenceType

ReferenceType використовується для вибору командної позиції або фактичної позиції як контрольної позиції. Коли *ReferenceType* дорівнює 0, опорна позиція є командною позицією осі. Якщо *ReferenceType* дорівнює 1, опорна позиція є фактичною позицією осі.

Коли позиція команди береться як опорна позиція, інструкція обчислює цільову позицію команди на основі поточної позиції команди та значення *Position* і переглядає значення позиції команди на значення цільової позиції. Тим часом фактичне положення осі відповідно зміниться. Закон зміни полягає в тому, що величина зміни фактичної позиції є такою ж, як і командної позиції. Тобто відхилення між командним положенням і

фактична позиція залишається незмінною в момент виконання інструкції та завершення виконання інструкції.

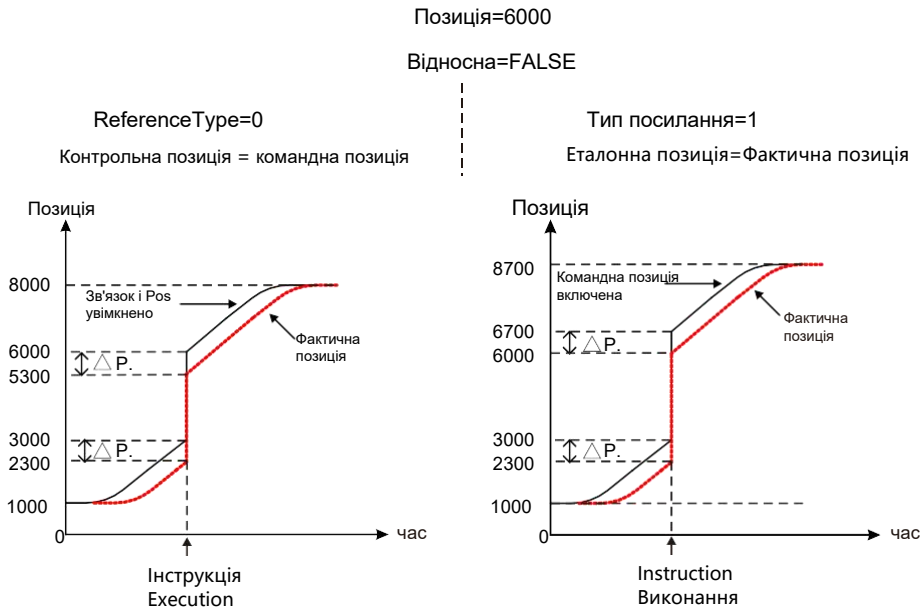
Рішення для фактичної позиції, яка береться за базову позицію, є таким самим, як і для командної позиції, яка береться за базову позицію.

Не буде ніякої різниці в ефекті виконання між положенням команди та фактичним положенням як опорним положенням, якщо вісь перебуває в стані зупинки, коли виконується MC_SetPosition. Це тому, що різниця між командним положенням і фактичним положенням дорівнює 0, оскільки вісь нерухома.

Відмінності в ефекті виконання між командною позицією та фактичною позицією як опорною позицією існують, як показано нижче, якщо вісь рухається під час виконання MC_SetPosition. Якщо не дорівнює нулю, різниця між командним положенням і фактичним положенням спричинена часом відповіді на команду.

Коли MC_SetPosition виконується в абсолютному режимі з *Position* , встановленим на 6000, тоді як вісь позиціонується з цільовою позицією 5000, положення команди та фактичне положення осі становлять 3000 і 2300 відповідно (різниця значення △ P =700). Позиція команди змінюється на 6000, а фактична позиція стає 5300 (5300=6000- △ P) після виконання інструкції, якщо контрольна позиція є командною позицією, як показано на малюнку зліва.

Фактична позиція осі змінюється на 6000, а позиція команди стає 6700 (6700=6000+ △ P) після виконання інструкції, якщо контрольна позиція є фактичною, як показано на малюнку справа.



■ **Зв’язок між типом осі та типом посилання**

Різні типи осей застосовуються до різних еталонних типів, як показано в наступній таблиці.

Тип осі	Тип посилання	
	Командна позиція	Фактична позиція
Реальна вісь	ТАК	ТАК
Вісь кодера	ТАК	ТАК
Віртуальна вісь	ТАК	ТАК

Виникне помилка у виконанні інструкції, якщо вісь, на якій виконується MC_SetPosition, не підтримує вибраний Reference Type.

■ **Пояснення до ситуації застосування інструкції**

Коли MC_SetPosition виконується на головній осі, яка знаходиться у вбудованому багатовісному відношенні, зміна положення головної осі, викликана інструкцією, не впливає на підлеглу вісь. Тобто підпорядкована вісь здійснюватиме будь-який рух відповідно, коли змінюється положення головної осі, викликане MC_SetPosition.

Коли MC_SetPosition виконується на підпорядкованій осі, положення підпорядкованої осі зміниться, але вихідне співвідношення між підпорядкованою віссю та головною віссю не вплине.

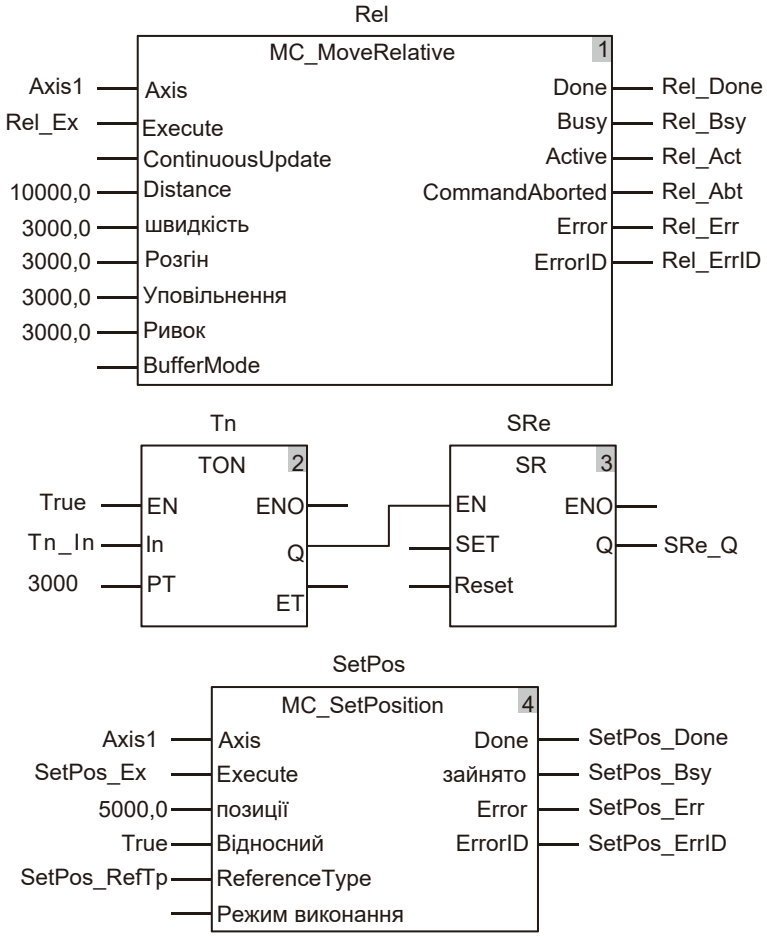
MC_SetPosition повідомить про помилку, коли він виконується в процесі виконання MC_Stop. Але MC_SetPosition можна виконати нормально після завершення виконання MC_Stop.

Приклад програмування 1

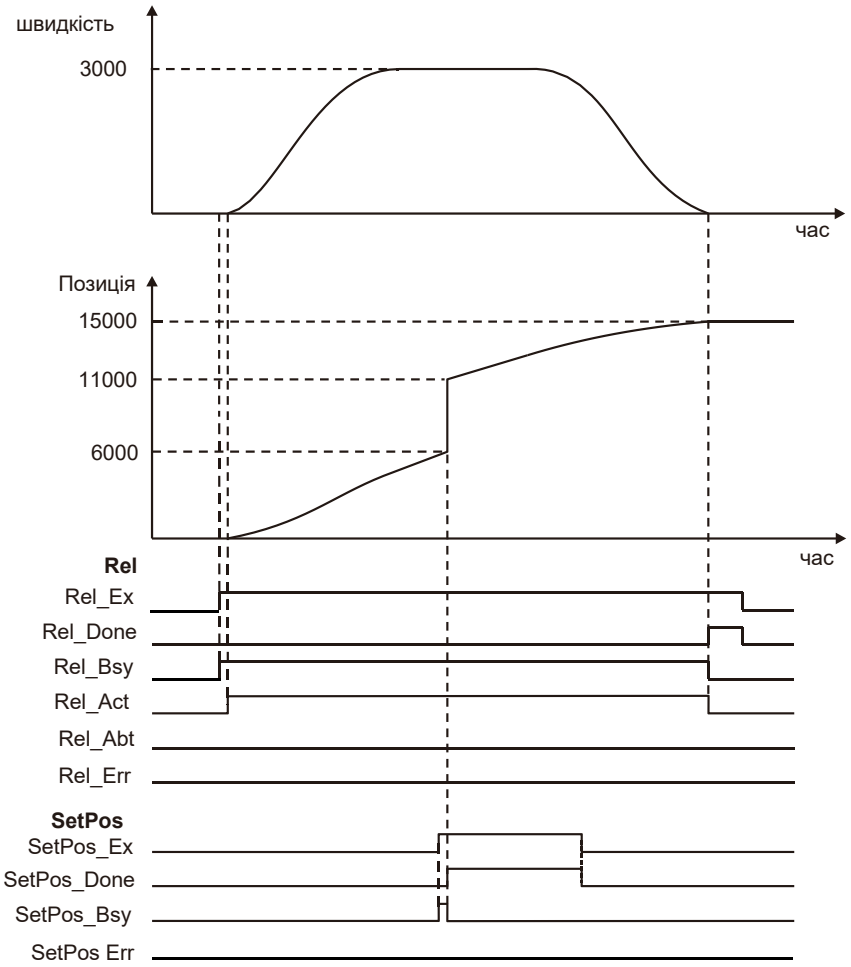
У наступному прикладі показано вплив виконання MC_SetPosition на інструкцію позиціонування, коли *Relative* of MC_SetPosition інструкції TRUE.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Rel	MC_MoveRelative	
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNE ТИП	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



2. Крива руху та часові діаграми:



- Коли Rel_Ex змінюється з FALSE на TRUE, починається виконання інструкції MC_MoveRelative, а MC_SetPosition виконується через 3 секунди після виконання MC_MoveRelative.
- Позиція команди становить 6000, коли MC_SetPosition починає виконуватися, і 11000 (11000=6000+5000) після завершення виконання інструкції. Позиція становить 15000, оскільки виконання MC_MoveRelative завершується.
- MC_SetPosition не впливає на рух, який виконується шляхом спостереження за наведеною вище кривою зміни швидкості.

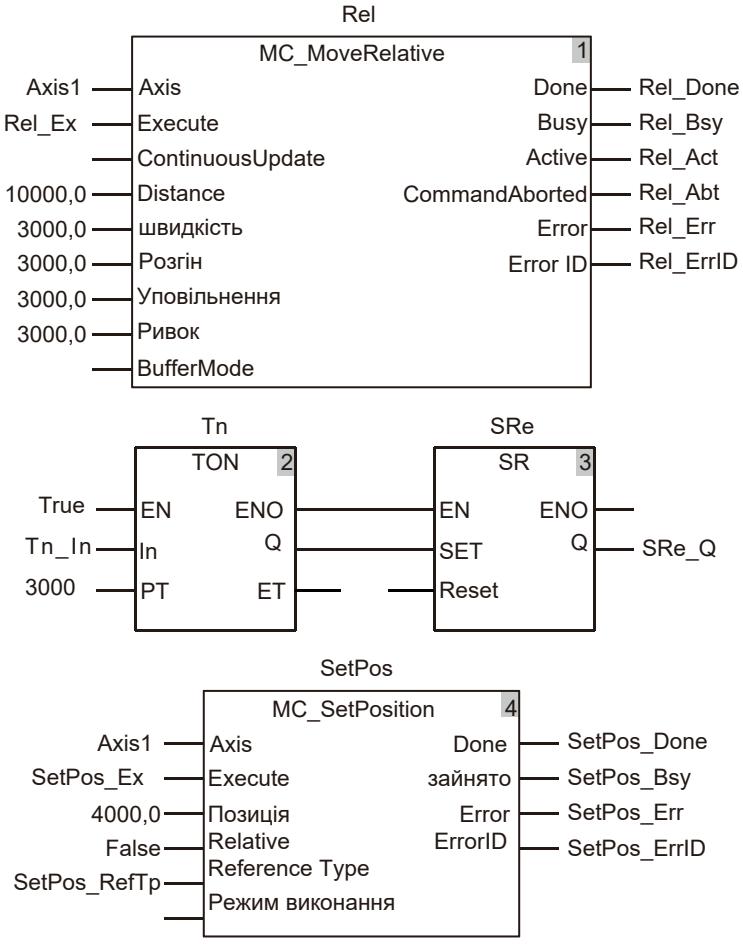
Приклад програмування 2

У наступному прикладі описано вплив виконання MC_SetPosition на положення осі, коли *Відносна* інструкції MC_SetPosition є FALSE (для MC_SetPosition вибрано абсолютний режим).

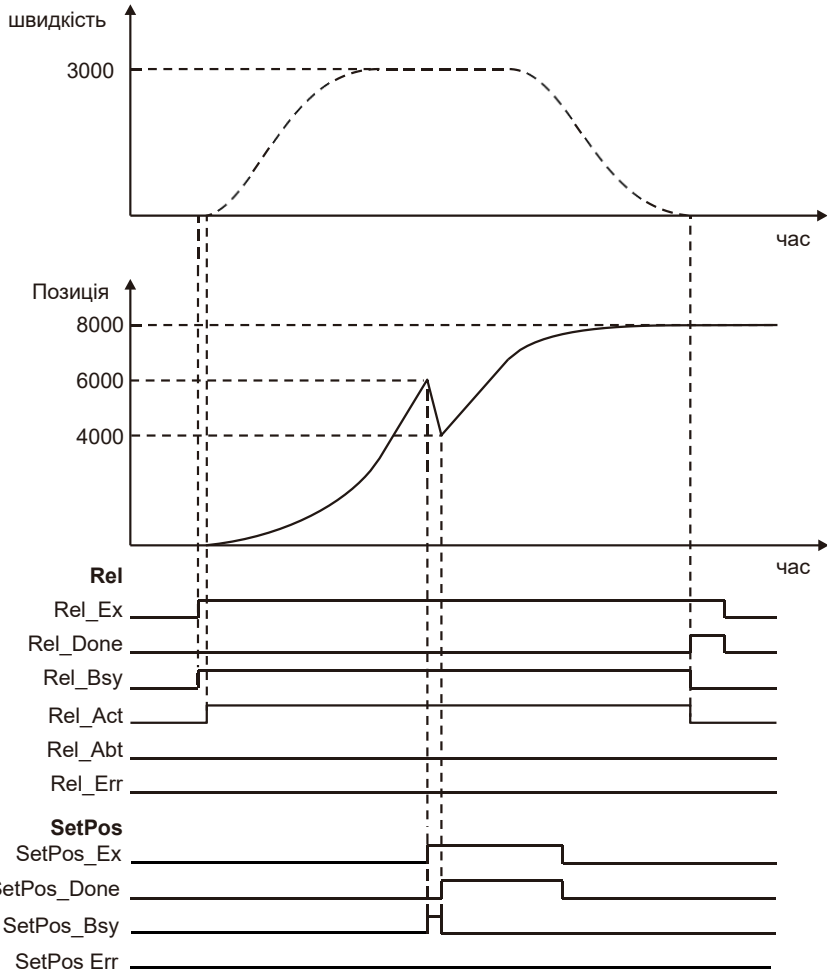
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Rel	MC_MoveRelative	
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	

Ім'я змінної	Тип даних	Початкове значення
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



2. Крива руху та часові діаграми:



- Коли Rel_Ex змінюється з FALSE на TRUE, починається виконання інструкції MC_MoveRelative, а MC_SetPosition виконується через 3 секунди після виконання MC_MoveRelative.
- Позиція команди становить 6000, коли MC_SetPosition починає виконуватися, і 4000 після завершення виконання інструкції. Позиція становить 8000, оскільки виконання MC_MoveRelative завершується.
- MC_SetPosition не впливає на рух, який виконується шляхом спостереження за наведеною вище кривою зміни швидкості.

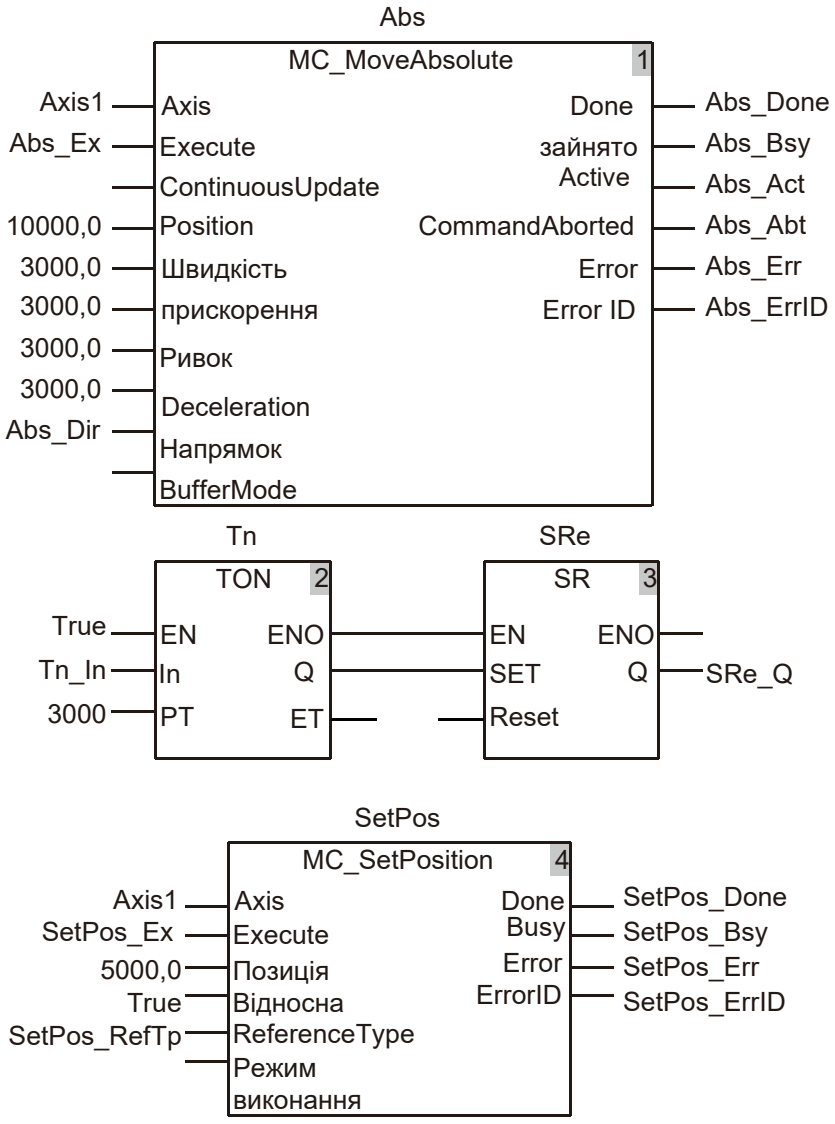
Приклад програмування 3

У наступному прикладі показано, як виконання MC_SetPosition впливає на інструкцію MC_MoveAbsolute, яка виконується. MC_SetPosition не впливає на фактичний ефект виконання MC_MoveAbsolute, який виконується.

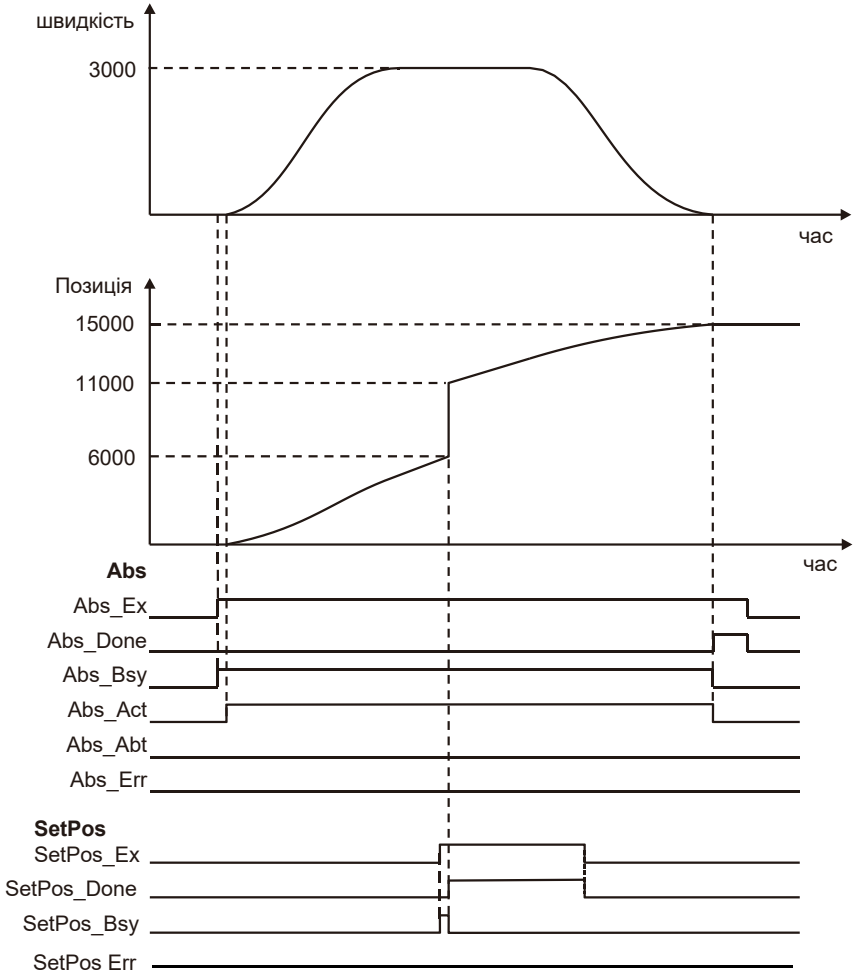
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Abs	MC_MoveAbsolute	
Axis1	USINT	1
Abs_Ex	BOOL	FALSE
Abs_Dir	MC_DIRECTION	1
Abs_Done	BOOL	
Abs_Bsy	BOOL	
Abs_Act	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Abs_Abt	BOOL	
Abs_Err	BOOL	
Abs_ErrID	WORD	
Tn	TON	
Tn_In	BOOL	FALSE
SRe	SR	
SRe_Q	BOOL	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	



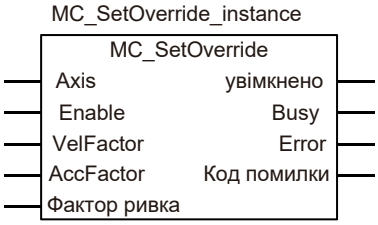
2. Крива руху та часові діаграми:



- Коли Abs_Ex змінюється з FALSE на TRUE, починається виконання інструкції MC_MoveAbsolute, а MC_SetPosition виконується через 3 секунди після виконання MC_MoveAbsolute.
- Позиція команди становить 6000, коли MC_SetPosition починає виконуватися, і 11000 після завершення виконання інструкції. Позиція становить 15000, оскільки виконання MC_MoveAbsolute завершується.
- Можна побачити, що MC_SetPosition не впливає на фактичний ефект виконання MC_MoveAbsolute, який виконується шляхом спостереження за наведеною вище кривою зміни швидкості.

11.3.12 MC_SetOverride

FB/FC	Пояснення	Застосовна модель
FB	MC_SetOverride змінює цільову швидкість для осі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ВелФактор	Коефіцієнт корекції швидкості (Одиниця: %)	LREAL	0~500 (100)	Коли *Enable* змінюється на TRUE
AccFactor	Зарезервований	-	-	-
JerkFactor	Зарезервований	-	-	-

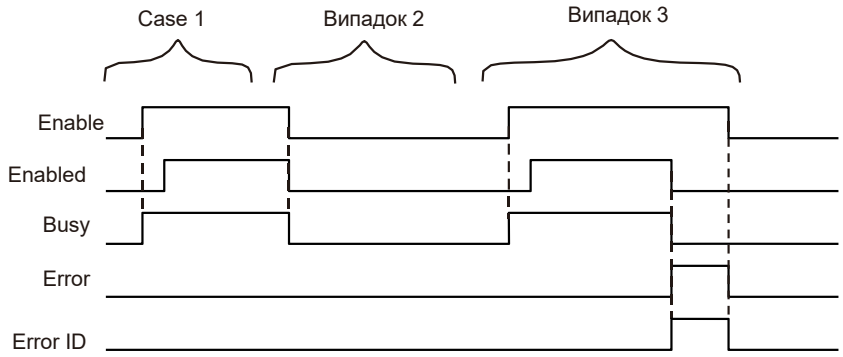
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Увімкнено	TRUE, коли інструкція керує віссю.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	-

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Увімкнено	◆ Коли починається інструктаж .	◆ Коли *Enable* змінюється на FALSE. ◆ Коли *помилка* змінюється на TRUE.
Busy	◆ Коли *Enable* має значення TRUE .	◆ Коли *Enable* змінюється на FALSE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. *Увімкнено* змінюється на TRUE, коли виконання інструкції завершено.

Випадок 2 : коли *Enable* змінюється з TRUE на FALSE, *Enabled* і *Busy* змінюються на FALSE.

Випадок 3 : коли виникає помилка після зміни *Enable* з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *параметри Enabled* і *Busy* змінюються на FALSE. *Error* змінюється на FALSE, коли *Enable* змінюється з TRUE на FALSE.

- **функція**

MC_SetOverride змінює цільову швидкість для осі.

1. Якщо цільові швидкості інструкцій руху потрібно змінити, використовуйте інструкцію MC_SetOverride. Тому інструкція не впливає на інструкції без цільових швидкостей. Однак *Enabled* залишається TRUE, навіть якщо *Enable* інструкції MC_SetOverride встановлено на TRUE для інструкцій, на які MC_SetOverride не впливає.
2. Інструкції, які цільові швидкості можна змінити за допомогою MC_SetOverride, наведено в наступній таблиці.

MC_MoveAbsolute （ Абсолютний позиціонування ）	MC_MoveRelative （ Відносний позиціонування ）
MC_MoveAdditive （ добавка позиціонування ）	MC_MoveVelocity （ швидкість інструкція ）
MC_MoveSuperimposed （ Накладене розташування ）	

3. Нова цільова швидкість розраховується, як показано нижче.

   Нова цільова швидкість після модифікації = цільова швидкість поточної виконаної інструкції x коефіцієнт перевизначення швидкості
4. Одиницею *VelFactor* є %. «100» означає «100%». Допустимий діапазон *VelFactor* становить від 0 до 500. Станеться помилка, якщо інструкція MC_SetOverride буде виконана, коли значення *VelFactor* перевищить допустимий діапазон.
5. Вісь буде прискорюватися вгору або вниз, доки не буде досягнуто цільової швидкості після модифікації *Прискорення* або *сповільнення* поточної інструкції.
6. Error виникне, коли цільова швидкість після модифікації перевищить максимальну швидкість у параметрах осі.
7. Якщо значення *VelFactor* встановлено на 0, цільова швидкість змінюється на 0, вісь сповільнюється, поки швидкість не буде 0. Якщо робочий стан осі потрібно зберегти, а роботу осі потрібно призупинити, встановіть значення *VelFactor* на 0. На даний момент стан осі не зміниться.
8. Коли інструкції руху виконуються або буферизуються, значення VelFactor можна змінити, щоб встановити нову цільову швидкість.
9. Якщо значення *VelFactor* змінено, коли *Enable* має значення TRUE, значення набуде чинності негайно без перезапуску інструкції MC_SetOverride.

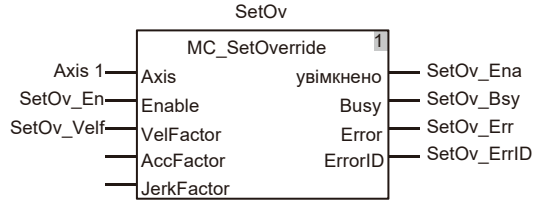
10. Якщо значення *VelFactor* змінено, коли *Enable* має значення TRUE, а значення *VelFactor* перевищує допустимий діапазон, у MC_SetOverride виникне помилка, і цільова швидкість повернеться до цього значення, оскільки значення *VelFactor* становить 100%.
11. Коли *Enable* змінюється на FALSE, вісь прискорюватиметься або гальмуватиметься, використовуючи VelFactor=100 як ціль.
12. Якщо інша інструкція MC_SetOverride запускається під час виконання однієї інструкції MC_SetOverride на осі, результат виконання пізніше виконаної інструкції MC_SetOverride буде розглядатися як еталонний результат. Значення *Enabled* для двох інструкцій має значення TRUE.
13. Якщо під час виконання інструкції MC_MoveVelocity використовується інструкція MC_SetOverride, *InVelocity* залишається TRUE, навіть якщо MC_SetOverride виконується після того, як *Invelocity* MC_MoveVelocity змінюється на TRUE.

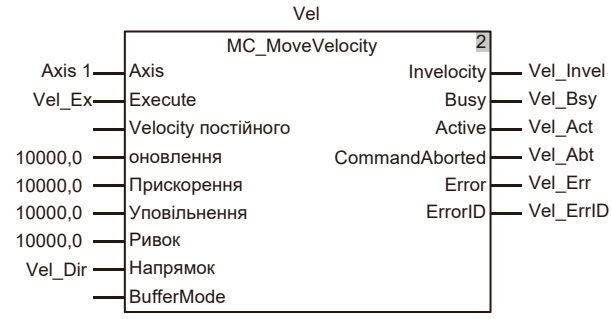
Приклад програмування

Приклад того, як на MC_MoveVelocity впливає виконання інструкції MC_SetOverride, описано нижче.

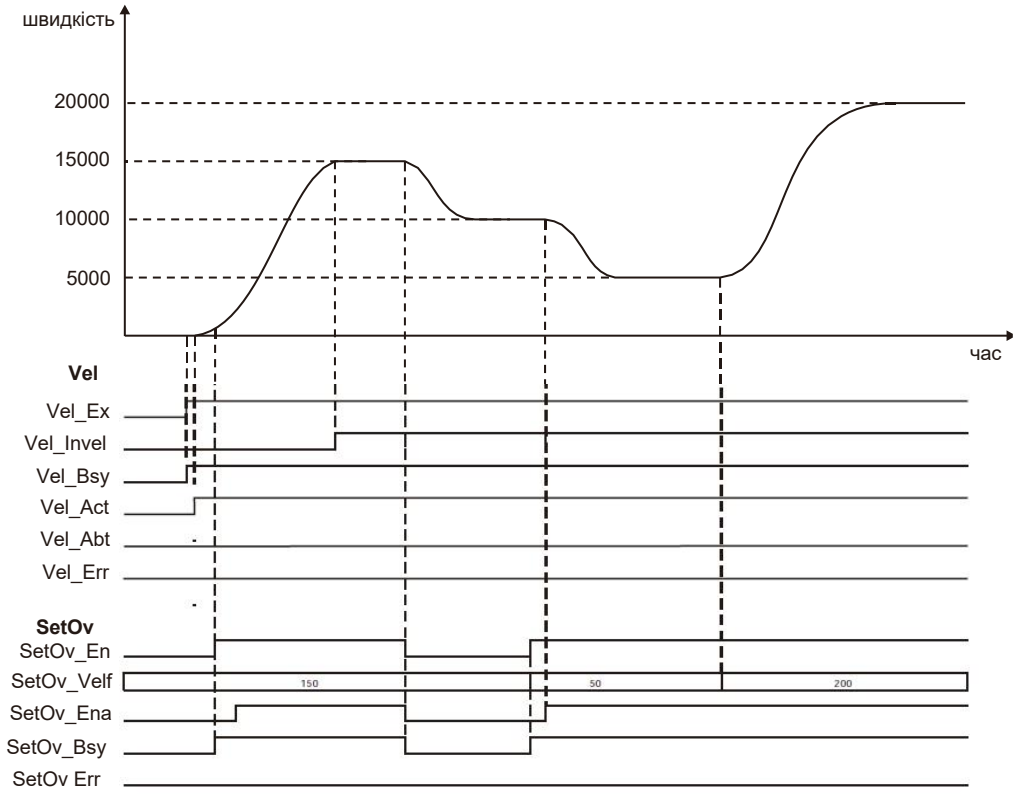
1. **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
SetOv	MC_SetOverride	
Axis1	USINT	1
SetOv_En	BOOL	FALSE
SetOv_Velt	LREAL	0.0
SetOv_Ena	BOOL	
SetOv_Bsy	BOOL	
SetOv_Err	BOOL	
SetOv_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	





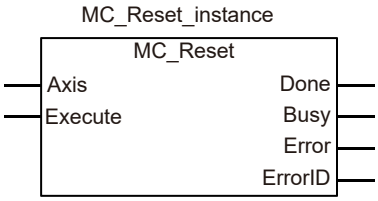
2. **Крива руху та часовий графік**



- Коли Vel_Ex змінюється на TRUE, Vel_Bsy змінюється на TRUE. Через один цикл Vel_Act змінюється на TRUE, і вісь починає рухатися вперед. Коли цільова швидкість не досягнута (Vel_Invel не TRUE), SetOv_En встановлено на TRUE, MC_SetOverride діє, а цільова швидкість MC_MoveVelocity змінюється на нову цільову швидкість. Коли досягається нова цільова швидкість MC_MoveVelocity, Vel_Invel змінюється на TRUE. Після того як Vel_Invel змінюється на TRUE, Vel_Invel залишається TRUE, навіть якщо значення VelFactor (SetOv_Velf) змінено.
- Коли SetOv_En змінюється на FALSE, це означає, що вісь починає сповільнюватися зі швидкістю, коли значення Vel_Invel дорівнює 100 як цільова швидкість.
- Значення SetOv_Velf набуде чинності негайно, якщо значення SetOv_Velf буде змінено під час виконання MC_SetOverride. Цільова швидкість MC_MoveVelocity зміниться відповідно.

11.3.13 MC_Reset

FB/FC	Пояснення	Застосовна модель
FB	MC_Reset очищає стани помилок і інформацію про тривогу осі всередині контролера руху.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. Функції в розділі 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

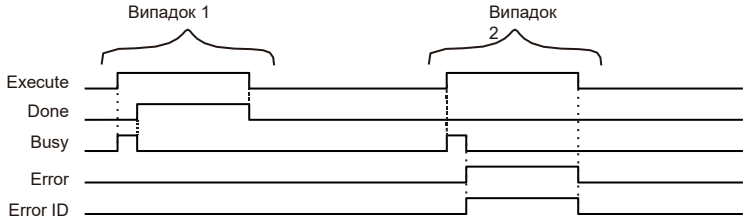
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли значення вхідних параметрів інструкції є недопустимими або помилку не можна виправити.	◆ Коли *Execute* змінюється з TRUE на FALSE

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* скидається на 0.

● функція

MC_Reset очищає стан помилки та інформацію про тривогу осі щодо реальної осі або віртуальної осі всередині контролера руху. Стан осі можна спостерігати за допомогою MC_ReadStatus. Інструкцію MC_Reset можна виконати для очищення помилок, коли вісь, налаштована в контролері руху, переходить у стан ErrorStop. Інструкція може бути виконана незалежно від того, переходить вісь у стан ErrorStop чи ні. Коли виникають такі помилки, як тривоги осі, вісь в автономному режимі або проблеми з перемикачем кінцевого автомата, вісь переходить у стан ErrorStop, а інструкції руху, які виконуються, припиняються. Коли вісь нагадує, виконання інструкції може очистити інформацію аварійного сигналу осі. Після завершення виконання інструкції MC_Reset стан осі буде визначено інструкцією MC_Power, і вісь буде в стані Disabled або Standstill.

Зверніться до розділу 9 для пояснення станів осі.

Після того, як виникає аварійний сигнал осі, за винятком аварійного сигналу, який виникає, коли вісь зустрічає кінцевий перемикач під час наведення, вісь сигналізації переходить у стан ErrorStop всередині контролера руху. Сигналізацію осі можна усунути, якщо *Done* має значення TRUE після виконання інструкції. Якщо *помилка* має значення TRUE, аварійний сигнал осі неможливо усунути, і користувачі повинні перевірити, чи причина помилки все ще існує.

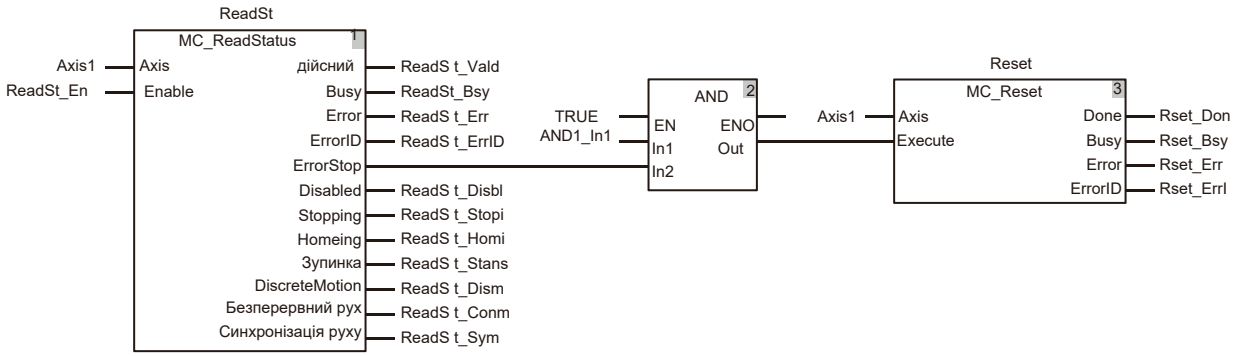
Приклад програмування

Коли ReadSt_En має значення TRUE, інструкція MC_ReadStatus визначить статус осі 1. Коли вісь 1 переходить у стан ErrorStop через вісь у режимі офлайн або аварійного сигналу, *ErrorStop* інструкції MC_ReadStatus зміниться на TRUE, а інструкція MC_Reset буде виконана.

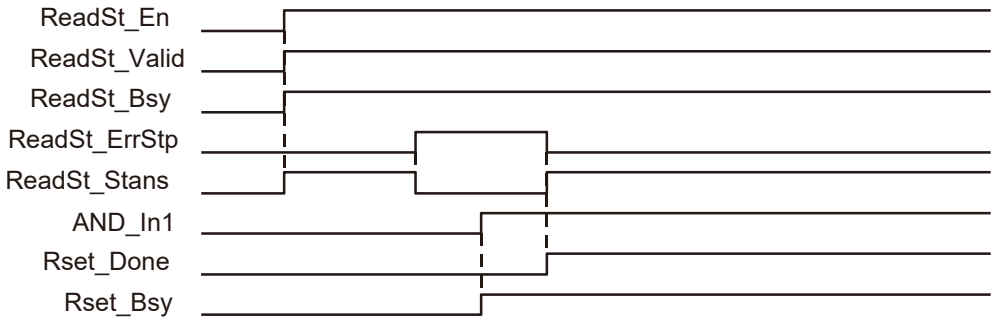
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ReadSt	MC_ReadStatus	
Axis1	USINT	1
ReadSt_En	BOOL	FALSE
ReadSt_Vald	BOOL	
ReadSt_Bsy	BOOL	
ReadSt_Err	BOOL	
ReadSt_ErrID	WORD	
ReadSt_Disbl	BOOL	
ReadSt_Stpin	BOOL	
ReadSt_Homi	BOOL	
ReadSt_Stans	BOOL	

Ім'я змінної	Тип даних	Початкове значення
ReadSt_Dism	BOOL	
ReadSt_Conm	BOOL	
ReadSt_Sym	BOOL	
AND1_In1	BOOL	FALSE
Reset	MC_Reset	
Rset_Done	BOOL	
Rset_Bsy	BOOL	
Rset_Err	BOOL	
Rset_ErrID	WORD	



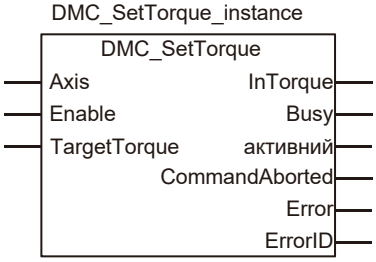
2. Графік часу



- Коли ReadSt_En змінюється з FALSE на TRUE після ввімкнення осі сервоприводу, ResdSt_Vald і ResdSt_Bsy змінюються на TRUE, а вісь перебуває в стані зупинки.
- AND_In1 встановлюється з FALSE на TRUE, коли вісь переходить у стан ErrorStop і виконується MC_Reset. Rset_Busy має значення TRUE у першому циклі, а Rset_Done має значення TRUE у другому циклі. Тим часом вісь із стану ErrorStop переходить у стан Standstill.

11.3.14 DMC_SetTorque

FB/FC	Пояснення	Застосовна модель
FB	DMC_SetTorque встановлює крутний момент осі сервоприводу. Під час виконання інструкції сервовісь працюватиме в режимі крутного моменту.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
TargetTorque	Вкажіть значення цільового крутного моменту. Крутний момент виражається промілом номінального крутного моменту осі сервоприводу. Наприклад, значення налаштування 30 вказує на те, що встановлений крутний момент становить 30‰ від номінального крутного моменту осі сервоприводу. Якщо *Enable* має значення TRUE, зміна значення параметра призведе до безпосереднього зміни крутного моменту.	INT	Від'ємне число, додатне число і 0 (0)	Коли *Enable* змінюється на TRUE

Примітки:

1. Якщо значення крутного моменту додатне число, ефект, який виробляє сервопривід, працює в позитивному напрямку. Якщо значення крутного моменту є від'ємним числом, ефект, який виробляє сервопривід, працює у від'ємному напрямку.
2. Якщо *Enable* має значення TRUE, інструкція завжди дійсна, а крутний момент змінюється відповідно до зміни значення крутного моменту. Інструкція не може бути перервана іншими інструкціями, крім MC_Stop. Коли *Enable* інструкції скидається на FALSE, виконання інструкції припиняється, і можна виконувати іншу інструкцію.

● **Вихідні параметри**

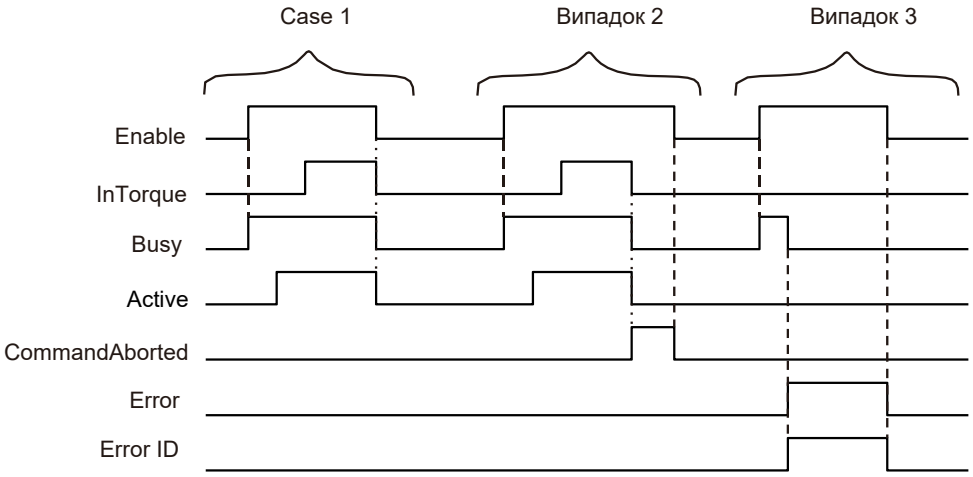
Назва параметра	функція	Тип даних	Допустимий діапазон
InTorque	TRUE, коли досягається цільовий крутний момент.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
InTorque	◆ Коли досягнуто цільового крутного моменту.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється з TRUE на FALSE
Busy	◆ Коли *Enable* змінюється на TRUE	◆ Коли *InTorque* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю	◆ Коли *InTorque* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Enable* змінює значення TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE в тому самому циклі. *Active* змінюється на TRUE у наступному циклі, а *InTorque* змінюється на TRUE у 3-му циклі . Коли *Enable* змінюється з TRUE на FALSE, *Busy* , *Active* та *InTorque* змінюються на FALSE в тому самому циклі.

Випадок 2 : коли інструкція DMC_SetTorque переривається MC_Stop після зміни *Enable* з FALSE на TRUE, CommandAborted змінюється на TRUE, а тим часом *InTorque* , *Busy* та *Active* змінюються на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *CommandAborted*
змінюється на FALSE.

Випадок 3 : значення вхідного параметра є недопустимим, наприклад, номер осі: 0 перед виконанням інструкції DMC_SetTorque. *Значення Busy* змінюється на TRUE, коли *Enable* змінюється з FALSE на TRUE. Через один цикл *помилка* змінюється на TRUE, *Busy* змінюється на FALSE, а *ErrorID* відображає відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а вміст *ErrorID* очищається до 0.

11

● функція

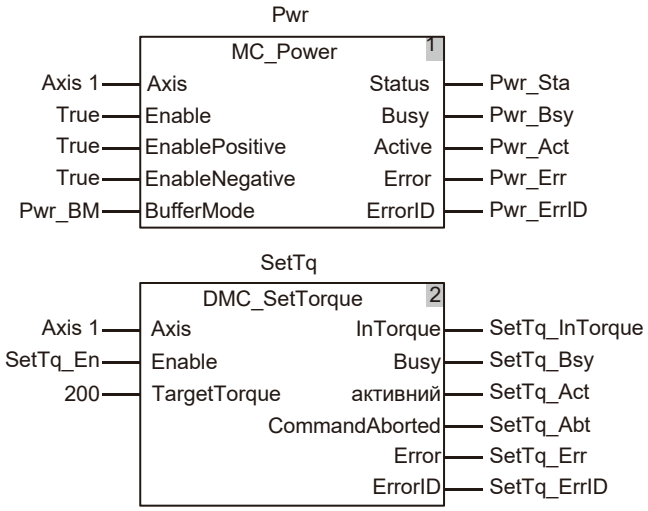
DMC_SetTorque встановлює крутний момент осі сервоприводу. Під час виконання інструкції сервовісь працюватиме в режимі крутного моменту.

Приклад програмування

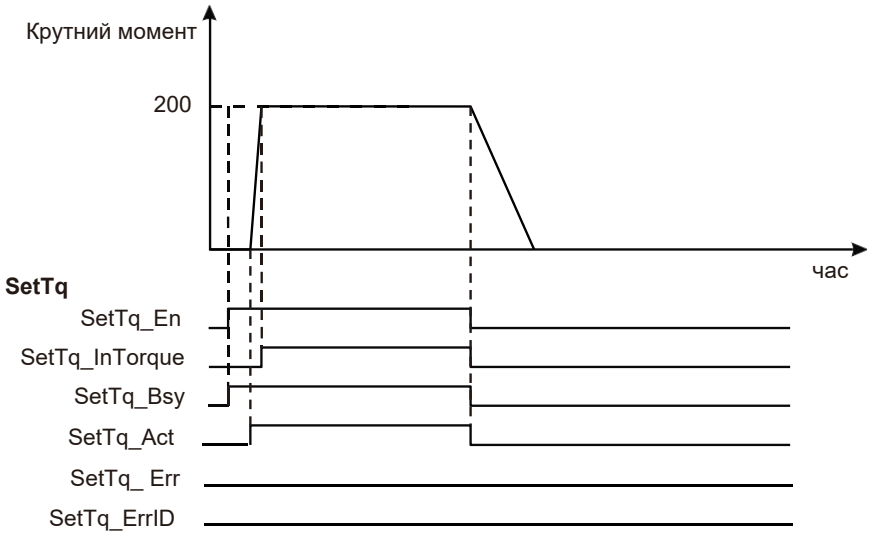
Приклад виконання інструкції DMC_SetTorque описується наступним чином.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
SetTq	DMC_SetTorque	
SetTq_En	BOOL	FALSE
SetTq_InTorque	BOOL	
SetTq_Bsy	BOOL	
SetTq_Act	BOOL	
SetTq_Abt	BOOL	
SetTq_Err	BOOL	
SetTq_ErrID	WORD	



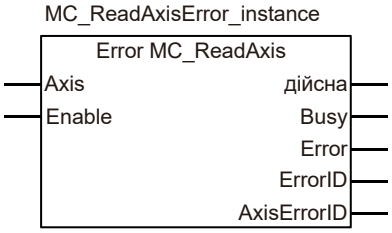
2. Крива руху та часовий графік



- Коли SetTq_En змінюється з FALSE на TRUE після ввімкнення осі сервоприводу, SetTq_Bsy змінюється на TRUE. Через один цикл SetTq_Act змінюється на TRUE і запускається інструкція DMC_SetTorque. Коли крутний момент досягається, SetTq_InTorque змінюється на TRUE, а SetTq_Bsy та SetTq_Act залишаються TRUE.
- SetTq_InTorque, SetTq_Bsy і SetTq_Act змінюються на FALSE, коли SetTq_En змінюється з FALSE на TRUE.

11.3.15 MC_ReadAxisError

FB/FC	Пояснення	Застосовна модель
FB	MC_ReadAxisError використовується для читання інформації про помилку осі сервоприводу.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

● **Вихідні параметри**

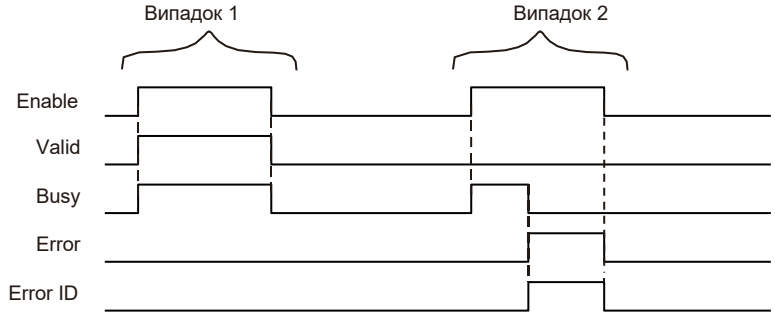
Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE під час виконання інструкції.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
AxisErrorID	Якщо *значення Valid* дорівнює TRUE, значення *ErrorID* , xxx (шістнадцяткове) вказує на те, що сервопривод видає сигнал тривоги, а xxx є кодом тривоги, який повідомляє сервопривод. Наприклад, AL303 сервоприводу означає, що значення *ErrorID* дорівнює 303 (шістнадцяткове).	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли зчитується помилка осі	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли помилка змінюється з FALSE на TRUE
Busy	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли помилка змінюється з FALSE на TRUE

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Enable* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE. Коли *Enable* змінюється на FALSE, *Valid* і *Busy* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки.

Тим часом *Busy* змінюється на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *виникає помилка*

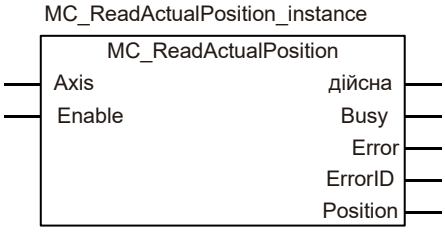
змінюється на FALSE, і значення *ErrorID* очищається.

● **функція**

MC_ReadAxisError використовується для читання інформації про помилку осі сервоприводу, наприклад коду тривоги, який відображатиметься на панелі сервоприводу та осі сервоприводу в автономному режимі. Інструкція запускається високим рівнем. Помилки осі зчитуються, якщо *для параметра Дійсно встановлено* значення TRUE.

11.3.16 MC_ReadActualPosition

FB/FC	Пояснення	Застосовна модель
FB	MC_ReadActualPosition використовується для читання фактичного положення осі, включаючи реальні осі, віртуальні осі та осі кодера.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

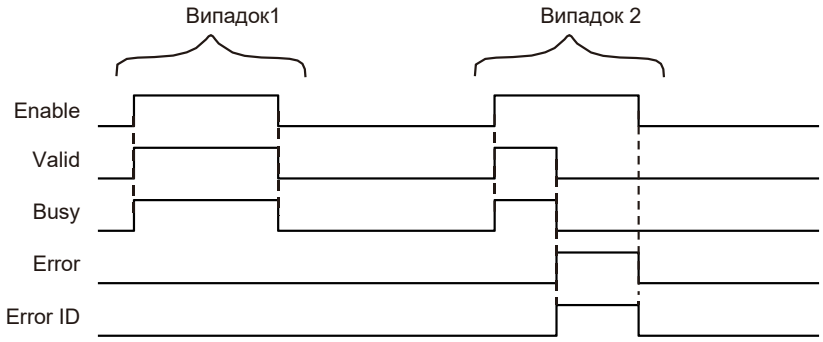
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE під час виконання інструкції.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Позиція	Фактичне положення осі.	LREAL	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли фактична позиція була прочитана.	◆ Коли *Enable* змінюється з TRUE на FALSE
Busy	◆ Коли *Enable* змінюється на TRUE.	◆ Коли «Готово» змінюється на TRUE ◆ Коли помилка змінюється на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Enable* змінюється з TRUE на FALSE

● Діаграма часу оновлення вихідних даних



Випадок 1 : Коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE одночасно. Коли *Enable* змінюється на FALSE, *Valid* і *Busy* змінюються на FALSE.

Випадок 2 : Коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *Busy* і Valid змінюються на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається.

● функція

MC_ReadActualPosition використовується для читання фактичного положення осі, включаючи реальну вісь, віртуальну вісь і вісь кодера.

■ Фактична позиція

Одиницею фактичної позиції, яку зчитує MC_ReadActualPosition, є одиниця, а одиницею зворотного зв'язку позиції, яку сервопривод передає контролеру, є імпульс. Таким чином, фактичне положення отримується шляхом перетворення кількості імпульсів зворотного зв'язку позиції сервоприводу. Передаточне число сервоприводу, механічне передаточне число та одиниці обертання вихідного сигналу серед параметрів осі необхідні для перетворення.

Формула перетворення показана нижче.

$$\text{ActualPosition} = \frac{\text{Одиниці обертання на виході}}{\text{(кількість імпульсів/обертання)} \\ \text{*механічне передавальне число}} * \text{Кількість імпульсів зворотного зв'язку положення сервоприводу}$$

Якщо вісь є лінійною віссю, її вихідне *положення* дорівнює фактичному положенню вище під час виконання інструкції.

Якщо вісь є віссю обертання, її вихідне *положення Position* дорівнює ActualPosition % за модулем під час виконання інструкції. (*Позиція* - це залишок, отриманий шляхом ділення Фактичної Позиції на встановлений модуль між параметрами осей) . Отже, значення *Position* змінюється від 0 до модуля.

■ Час для оновлення фактичної позиції

Час для оновлення фактичного положення пов'язаний із часом циклу зв'язку між контролером і сервоприводом, оскільки фактичне положення виходить із кількості імпульсів зворотного зв'язку позиції, які дає сервопривод. За один цикл зв'язку сервопривід посилає кількість імпульсів позиції зворотного зв'язку до контролера лише один раз. Таким чином, зчитана фактична позиція залишається незмінною протягом одного циклу зв'язку.

Z причин, зазначених вище, будь ласка, використовуйте функцію захоплення положення, щоб отримати більш точне положення в реальному часі, оскільки інструкція зчитує фактичне положення осі в меншому реальному часі, ніж функція захоплення положення.

11

■ Вплив MC_SetPosition на фактичну позицію

Фактична позиція, яку зчитує MC_ReadActualPosition, повинна також включати зміщення позиції, викликане MC_SetPosition після виконання MC_SetPosition.
Формула перетворення показана нижче.

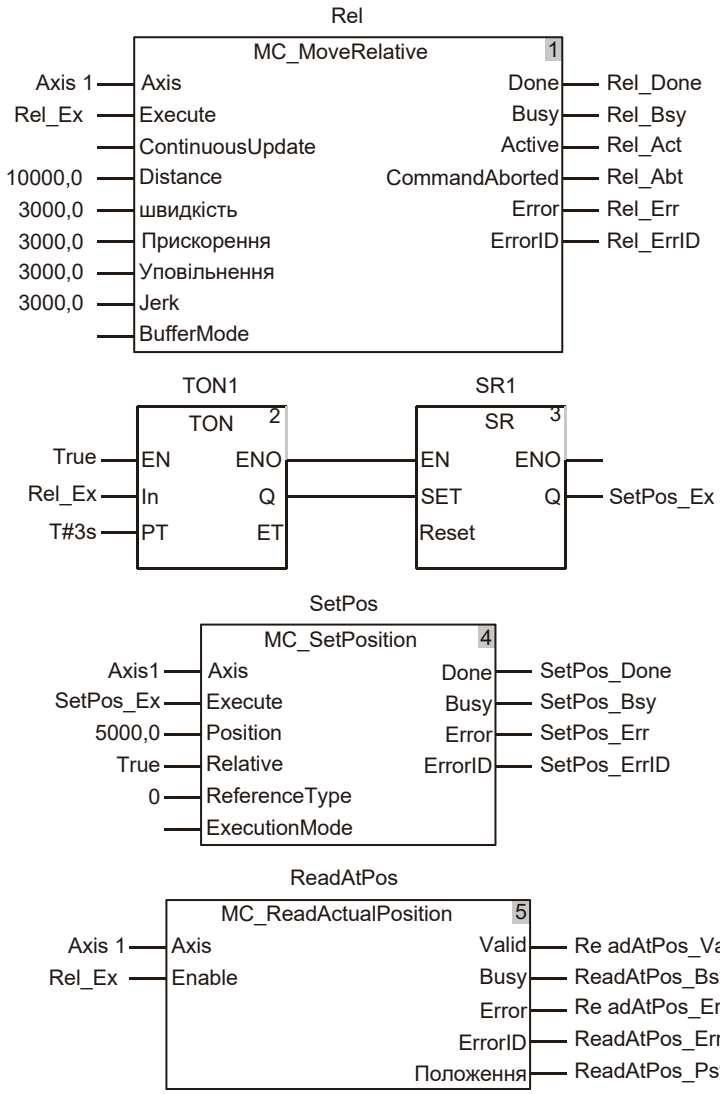
$$\text{ActualPosition} = \text{Зміщення позиції викликано MC_SetPosition} + \frac{\text{Одиниці обертання на виході}}{\text{(кількість імпульсів / обертання)} * \text{механічне передавальне число}} * \text{Кількість імпульсів зворотного зв'язку положення сервоприводу}$$

Приклад програмування

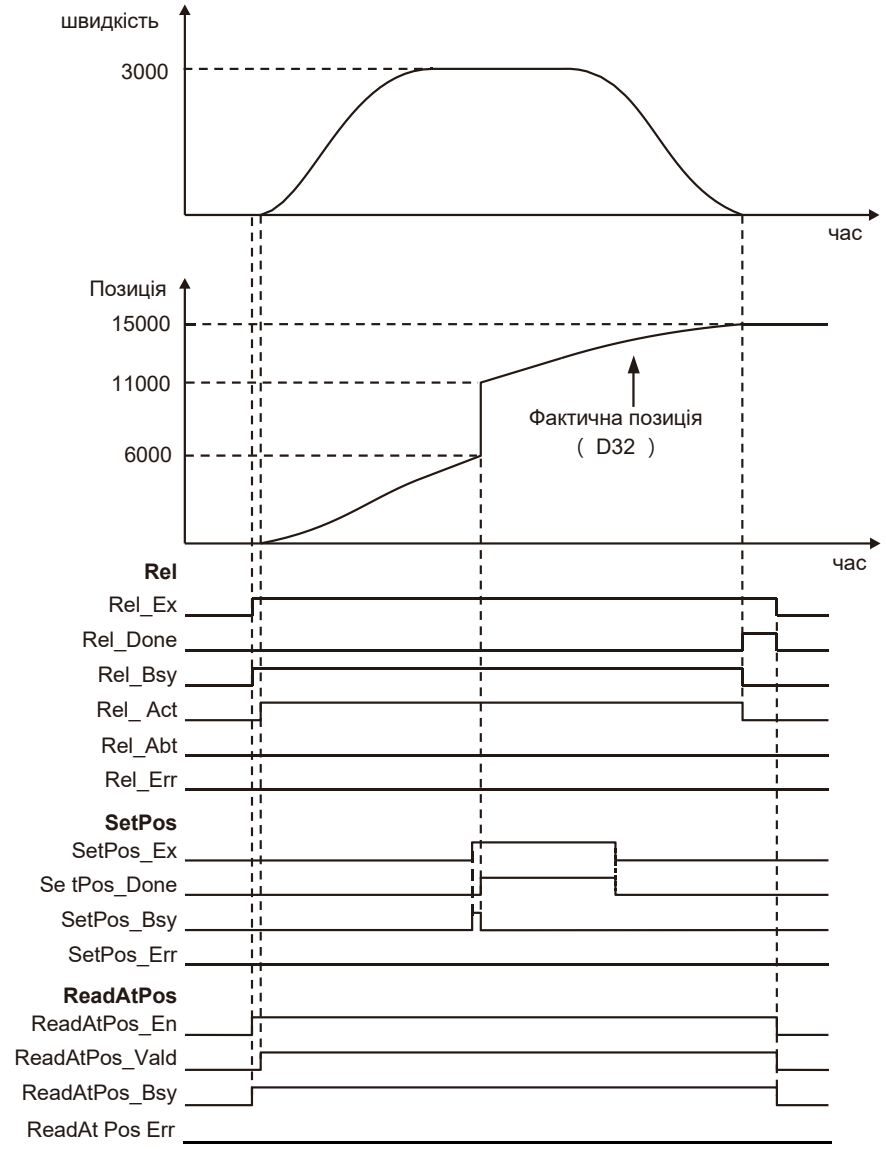
У цьому прикладі показано вплив MC_SetPosition на виконання MC_ReadActualPosition.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Rel	MC_MoveRelative	
Axis1	USINT	1
Rel_Ex	BOOL	FALSE
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	
Rel_Err	BOOL	
Rel_ErrID	WORD	
TON1	TON	
SR1	SR	
SetPos	SetPosition	
SetPos_Ex	BOOL	FALSE
SetPos_RefTp	MC_REFERECNETYPE	0
SetPos_Done	BOOL	
SetPos_Bsy	BOOL	
SetPos_Err	BOOL	
SetPos_ErrID	WORD	
ReadAtPos	ReadActualPosition	
ReadAtPos_Vald	BOOL	
ReadAtPos_Bsy	BOOL	
ReadAtPos_Err	BOOL	
ReadAtPos_ErrID	WORD	
ReadAtPos_Pst	LREAL	


Rel
MC_MoveRelative
1
Axis 1
Axis
Rel_Ex
Execute
ContinuousUpdate
10000,0
Distance
3000,0
швидкість
3000,0
Прискорення
3000,0
Уповільнення
3000,0
Jerk
BufferMode
Done
Rel_Done
Busy
Rel_Bsy
Active
Rel_Act
CommandAborted
Rel_Abt
Error
Rel_Err
ErrorID
Rel_ErrID
TON1
TON
2
True
EN
ENO
Rel_Ex
In
Q
T#3s
PT
ET
SR1
SR
3
EN
ENO
SET
Q
SetPos_Ex
Reset
SetPos
MC_SetPosition
4
Axis1
Axis
SetPos_Ex
Execute
5000,0
Position
True
Relative
0
ReferenceType
ExecutionMode
Done
SetPos_Done
Busy
SetPos_Bsy
Error
SetPos_Err
ErrorID
SetPos_ErrID
ReadAtPos
MC_ReadActualPosition
5
Axis 1
Axis
Rel_Ex
Enable
Valid
Re adAtPos_Vald
Busy
ReadAtPos_Bsy
Error
Re adAtPos_Err
ErrorID
ReadAtPos_ErrID
Положення
ReadAtPos_Pst

2. Крива руху та часові діаграми:



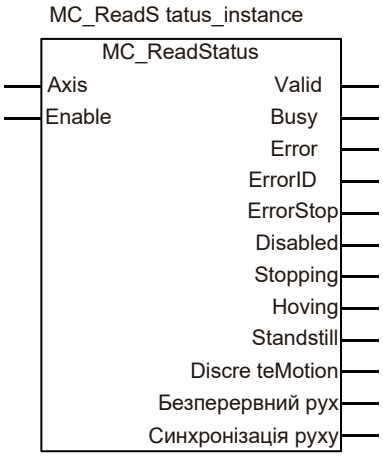
- Коли Rel_Ex змінюється з FALSE на TRUE, виконання MC_MoveRelative і MC_ReadActualPosition починається одночасно. MC_SetPosition виконується через 3 секунди після виконання MC_MoveRelative.

- Фактична позиція становить 6000, коли MC_SetPosition починає виконуватися, і 11000 (11000=6000+5000) після завершення виконання. Фактична позиція становить 15000 після завершення виконання MC_MoveRelative.

- З наведеної вище діаграми кривої швидкості видно, що MC_SetPosition не впливає на поточний рух. Але діаграма кривої ActualPosition відображає, що на фактичну позицію, яку зчитує MC_ReadActualPosition, впливає MC_SetPosition.

11.3.17 MC_ReadStatus

FB/FC	Пояснення	Застосовна модель
FB	MC_ReadStatus використовується для читання стану осі сервоприводу в контролері.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли Enable змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE під час виконання інструкції.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	
ErrorStop	Див. розділ 10.4.	BOOL	TRUE / FALSE
Вимкнено		BOOL	TRUE / FALSE
Зупинка		BOOL	TRUE / FALSE
Наведення		BOOL	TRUE / FALSE
Застій		BOOL	TRUE / FALSE
Дискретний рух		BOOL	TRUE / FALSE
Безперервний рух		BOOL	TRUE / FALSE
SyncMotion		BOOL	TRUE / FALSE

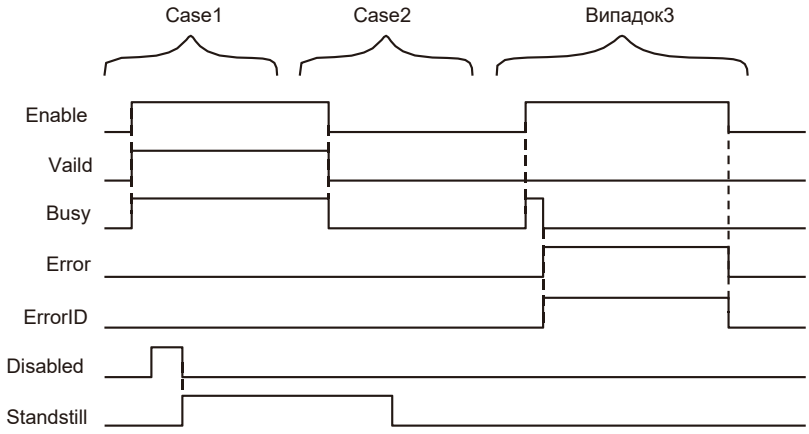
Примітки:

1. Коли *Enable* змінюється з FALSE на TRUE, починається виконання MC_ReadStatus і зчитується стан осі.
2. Коли *Enable* змінюється з TRUE на FALSE, *Valid, Busy* і *Error* змінюються на FALSE, тим часом *ErrorID* змінюється на 0, а виходи *ErrorStop, Disabled, Stopping, Homeing, Standstill, DiscreteMotion, ContinuousMotion* і *SyncMotion* зберігають статус *Enable як* TRUE.

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється з FALSE на TRUE
Busy	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється з FALSE на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Enable* змінюється з TRUE на FALSE
ErrorStop	◆ Коли вісь переходить у стан ErrorStop	◆ Коли вісь не в стані ErrorStop
Вимкнено	◆ Коли вісь переходить у вимкнений стан	◆ Коли вісь не вимкнуто
Зупинка	◆ Коли вісь переходить у стан зупинки	◆ Коли вісь не в стані зупинки
Наведення	◆ Коли вісь переходить у початковий стан	◆ Коли вісь не в початковому стані
Застій	◆ Коли вісь переходить у стан зупинки	◆ Коли вісь не стоїть на місці
Дискретний рух	◆ Коли вісь переходить у стан DiscreteMotion	◆ Коли вісь не в стані DiscreteMotion
Безперервний рух	◆ Коли вісь переходить у стан ContinuousMotion	◆ Коли вісь не в стані ContinuousMotion
SyncMotion	◆ Коли вісь переходить у стан SyncMotion	◆ Коли вісь не в стані SyncMotion

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE одночасно, а *ErrorStop, Disabled, Stopping, Homing, Standstill, DiscreteMotion, ContinuousMotion* і *SyncMotion* зміняться на TRUE або FALSE відповідно до стану осі.

Випадок 2 : коли *Enable* змінюється з TRUE на FALSE, *Valid* і *Busy* змінюються на FALSE одночасно, а виходи *ErrorStop, Disabled, Stopping, Homing, Standstill, DiscreteMotion, ContinuousMotion* і *SyncMotion* зберігатимуть той самий стан, що й *Enable* має значення TRUE.

Випадок 3 : коли значення вхідного параметра Axis виходить за межі допустимого діапазону та *Enable* змінюється з FALSE на TRUE, *Busy* змінюється з FALSE на TRUE, через один цикл, *Error* змінюється з FALSE на TRUE, а *ErrorID* показує відповідні коди помилок і *Busy* змінюється з TRUE на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а *ErrorID тим часом* змінюється на 0.

- **функція**

MC_ReadStatus використовується для читання стану осі сервоприводу в контролері.
Докладніше про стан осі див. у розділі 10.4.

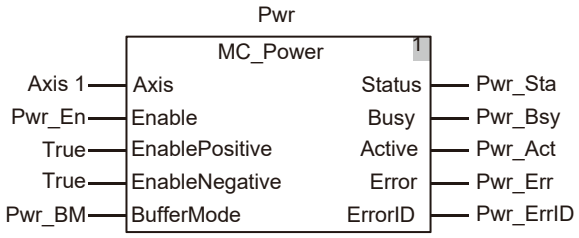
Приклад програмування

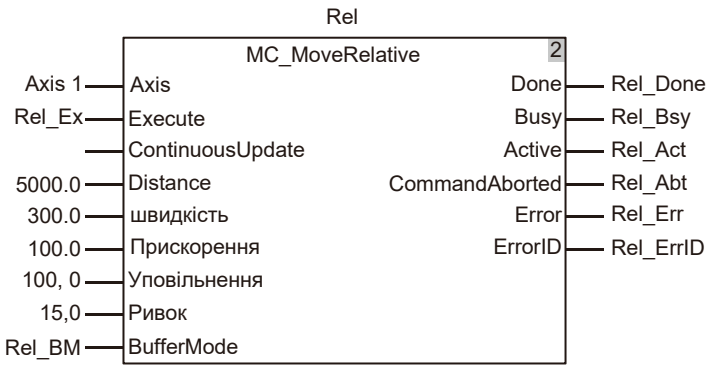
Цей приклад виконання MC_ReadStatus показаний нижче.

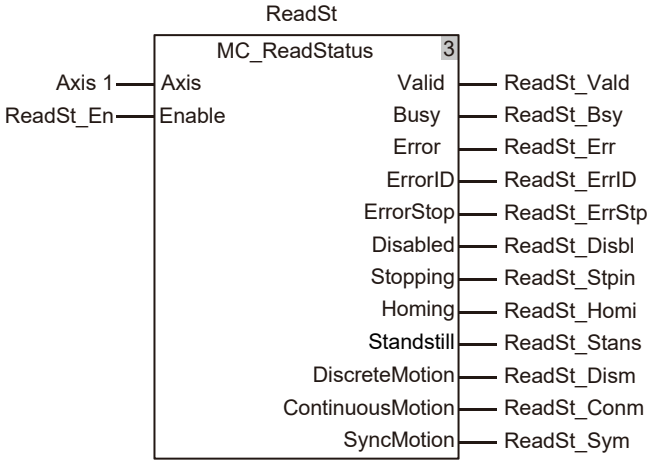
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel	MC_MoveRelative	
Rel_Ex	BOOL	FALSE
Rel_BM	MC_Buffer_Mode	0
Rel_Done	BOOL	
Rel_Bsy	BOOL	
Rel_Act	BOOL	
Rel_Abt	BOOL	

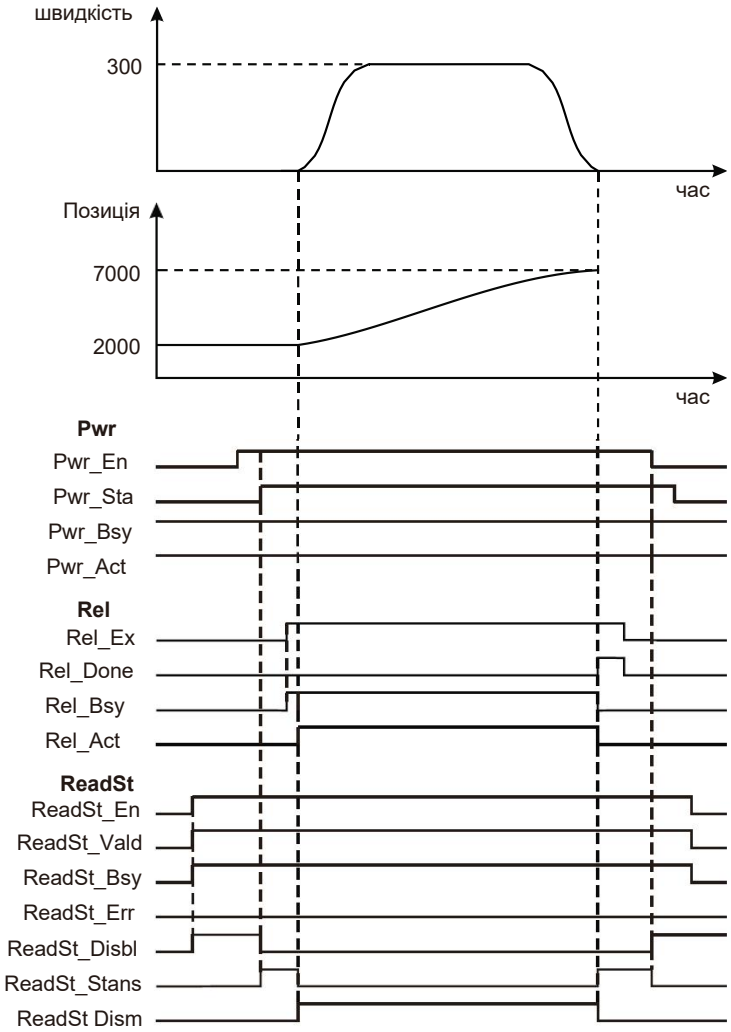
Ім'я змінної	Тип даних	Початкове значення
Rel_Err	BOOL	
Rel_ErrID	WORD	
ReadSt	MC_ReadStatus	
ReadSt_En	BOOL	FALSE
ReadSt_Vald	BOOL	
ReadSt_Bsy	BOOL	
ReadSt_Err	BOOL	
ReadSt_ErrID	WORD	
ReadSt_ErrStp	BOOL	
ReadSt_Disbl	BOOL	
ReadSt_Stpin	BOOL	
ReadSt_Homi	BOOL	
ReadSt_Stans	BOOL	
ReadSt_Dism	BOOL	
ReadSt_Conm	BOOL	
ReadSt_Sym	BOOL	







2. Крива руху та часові діаграми:



- ReadSt_Vald, ReadSt_Bsy і ReadSt_Disbl змінюються на TRUE, оскільки ReadSt_En змінюється з FALSE на TRUE.

- Коли Pwr_Sta змінюється з FALSE на TRUE, ReadSt_Stans змінюється на TRUE, ReadSt_Disbl змінюється на FALSE, а стан осі змінюється з Disabled на Standstill.

- Контролер руху контролює рух серводвигуна, починаючи з поточної позиції, коли Rel_Act змінюється з FALSE на TRUE. Тим часом ReadSt_Stans змінюється на FALSE, а ReadSt_Dism змінюється на TRUE. Коли серводвигун переміщує цільову відстань, Rel_Done і ReadSt_Stans змінюються на TRUE; Rel_Bsy, Rel_Act і ReadSt_Dism змінюються на FALSE.

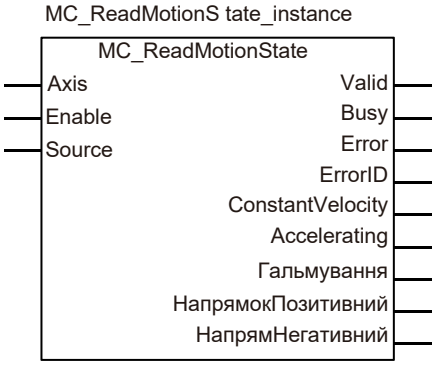
- Rel_Done також змінюється на FALSE, оскільки Rel_Ex змінюється на FALSE.

- Коли Pwr_En змінюється на FALSE, ReadSt_Disbl змінюється на TRUE, ReadSt_Stans змінюється на FALSE, а через кілька циклів Pwr_Sta також змінюється на FALSE.

- Коли ReadSt_En змінюється на FALSE, ReadSt_Vald і ReadSt_Bsy змінюються на FALSE, а ReadSt_Disbl залишається TRUE.

11.3.18 MC_ReadMotionState

FB/FC	Пояснення	Застосовна модель
FB	MC_ReadMotionState використовується для читання поточного стану руху осі сервоприводу.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
Джерело	Зарезервований	-	-	-

Примітки:

1. Коли *Enable* змінюється з FALSE на TRUE, починається виконання MC_ReadStatus.
2. Коли виконується MC_ReadStatus і *Enable* змінюється з TRUE на FALSE, виконання інструкції припиняється, а виходи *ConstantVelocity* , *Accelerating* , *Decelerating* , *DirectionPositive* і *DirectionNegative* зберігають статус *Enable як* TRUE.

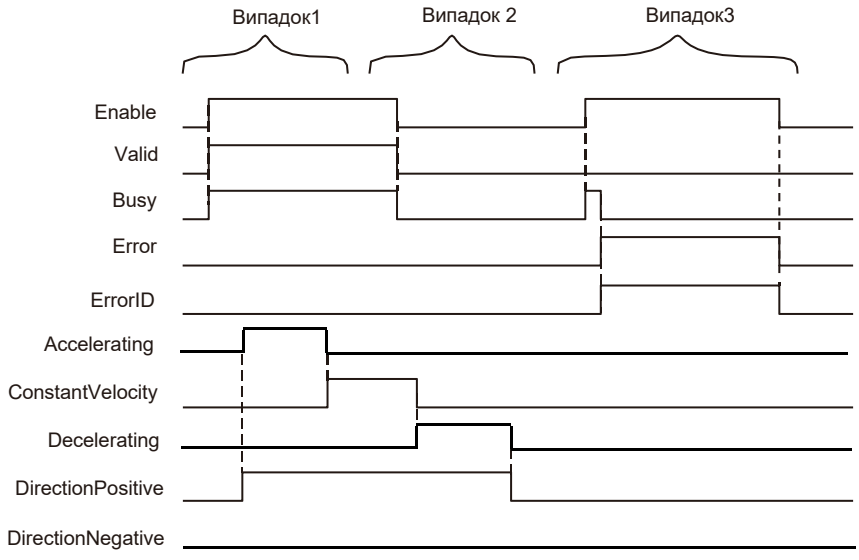
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE під час виконання інструкції.	BOOL	TRUE / FALSE
Error	TRUE, поки є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	
Постійна швидкість	TRUE, коли вісь рухається з постійною швидкістю	BOOL	TRUE / FALSE
Прискорення	TRUE, коли абсолютне значення осьової швидкості збільшується.	BOOL	TRUE / FALSE
Гальмування	TRUE, коли абсолютне значення осьової швидкості зменшується.	BOOL	TRUE / FALSE
НапрямокПозитивний	TRUE, коли поточне значення позиції збільшується.	BOOL	TRUE / FALSE
DirectionNegative	TRUE, коли поточне значення позиції зменшено.	BOOL	TRUE / FALSE

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли зчитується фактична швидкість осі	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється з FALSE на TRUE
Busy	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється з FALSE на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Enable* змінюється з TRUE на FALSE
ErrorID		
Постійна швидкість	◆ Коли осьова швидкість не змінюється	◆ Коли швидкість осі змінюється, а *параметр «Увімкнути»* залишається TRUE
Прискорення	◆ При цьому абсолютна величина осьової швидкості збільшується	◆ Коли швидкість осі більше не збільшується, а *Enable* все ще має значення TRUE
Гальмування	◆ При цьому абсолютна величина осьової швидкості зменшується	◆ Коли швидкість осі більше не зменшується, а *Enable* все ще має значення TRUE
НапрямокПозитивний	◆ Коли поточне значення позиції збільшується	◆ Коли значення поточної позиції більше не збільшується, а *Enable* все ще має значення TRUE
DirectionNegative	◆ Коли значення поточної позиції зменшено	◆ Коли значення поточної позиції більше не зменшується, а *Enable* все ще має значення TRUE

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE, а ConstantVelocity, Accelerating, Decelerating, DirectionPositive і DirectionNegative змінюються на TRUE або FALSE відповідно до стану осі.

Випадок 2 : коли *Enable* змінюється з TRUE на FALSE, *Valid* і *Busy* змінюються на FALSE, а *ConstantVelocity, Accelerating, Decelerating, DirectionPositive* і *DirectionNegative* залишаються в стані, коли *Enable* має значення TRUE.

Випадок 3 : коли значення Axis виходить за межі допустимого діапазону та *Enable* змінюється з FALSE на TRUE, *Busy* змінюється з FALSE на TRUE, через один період, *Error* змінюється з FALSE на TRUE, а *ErrorID* показує відповідні коди помилок. Тим часом *Busy* змінюється з TRUE на FALSE. *Error* змінюється з TRUE на FALSE, а значення *ErrorID* стає 0, коли *Enable* змінюється з TRUE на FALSE.

- **функція**

MC_ReadMotionState використовується для читання поточного стану руху осі сервоприводу. Стан руху осі сервоприводу включає постійний рух, прискорення або уповільнення, позитивне обертання та негативне обертання.

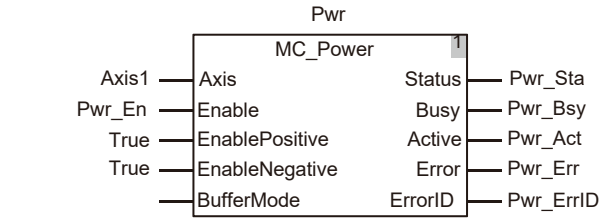
Приклад програмування

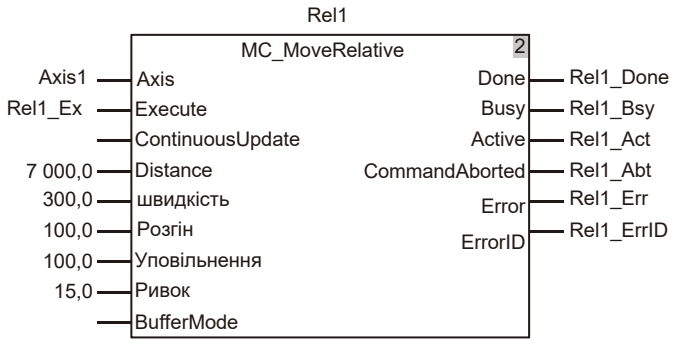
Цей приклад виконання MC_ ReadMotionState показаний нижче.

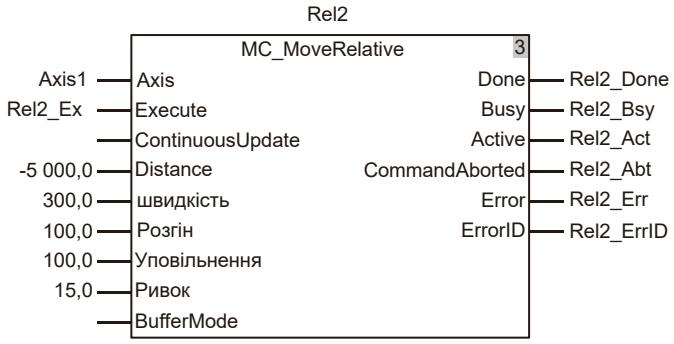
1. Таблиця змінних і програма

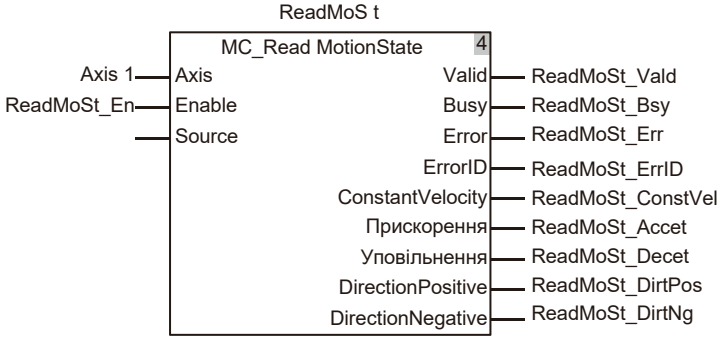
Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	
ReadMoSt	MC_ReadMotionState	
ReadMoSt_En	BOOL	FALSE
ReadMoSt_Vald	BOOL	

Ім'я змінної	Тип даних	Початкове значення
ReadMoSt_Bsy	BOOL	
ReadMoSt_Err	BOOL	
ReadMoSt_ErrID	WORD	
ReadMoSt_ConstVel	BOOL	
ReadMoSt_Accet	BOOL	
ReadMoSt_Decet	BOOL	
ReadMoSt_DirtPos	BOOL	
ReadMoSt_DirtNg	BOOL	

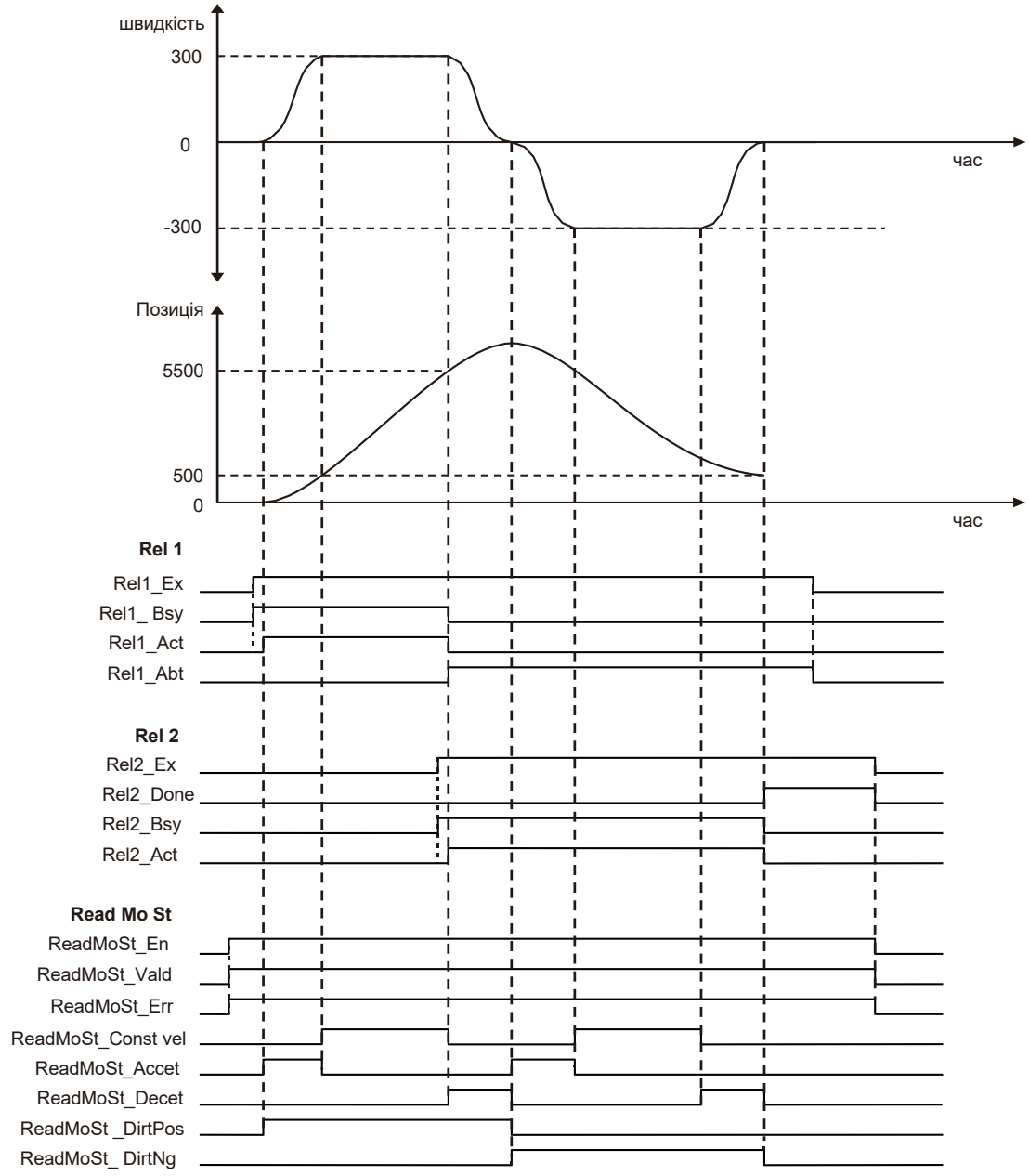








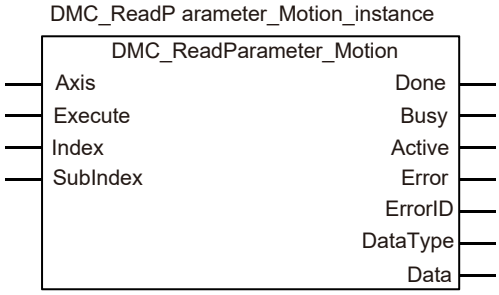
2. Крива руху та часові діаграми:



- ReadMoSt_Vald і ReadMoSt_Bsy змінюються з FALSE на TRUE, оскільки ReadMoSt_En змінюється з FALSE на TRUE.
- Коли Rel1_Act змінюється з FALSE на TRUE, вісь починає прискорюватися в позитивному напрямку, а ReadMoSt_Accet і ReadMoSt_DirtPos тим часом змінюються на TRUE.
- Коли ReadMoSt_Constvel змінюється з FALSE на TRUE, ReadMoSt_Accet змінюється з TRUE на FALSE, і вісь переходить у стан руху з постійною швидкістю в позитивному напрямку.
- Коли Rel2_Act змінюється з FALSE на TRUE, ReadMoSt_Decet змінюється з FALSE на TRUE, і вісь починає сповільнюватися в позитивному напрямку.
- Коли ReadMoSt_Accet і ReadMoSt_DirtNg змінюються з FALSE на TRUE, ReadMoSt_Decet і ReadMoSt_DirtPos змінюються на FALSE одночасно, і вісь починає прискорюватися в негативному напрямку.
- Коли Rel2_Done змінюється з FALSE на TRUE, вісь перестає рухатися, а ReadMoSt_Decet і ReadMoSt_DirtNg змінюються на FALSE.

11.3.19 DMC_ReadParameter_Motion

FB/FC	Пояснення	Застосовна модель
FB	DMC_ReadParameter_Motion зчитує значення підлеглого параметра.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть адресу станції веденого пристрою для керування.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Index	Індекс параметра для читання	UINT	0	Коли *Execute* змінюється з FALSE на TRUE
SubIndex	Субіндекс параметра для читання	USINT	0	Коли *Execute* змінюється з FALSE на TRUE

- **Вихідні параметри**

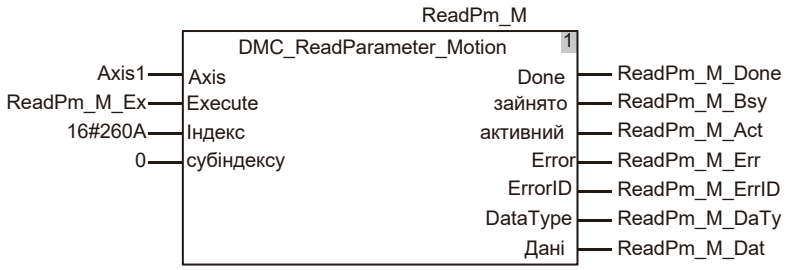
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	
DataType	Тип даних параметра читання 1 : байт, 2 : Слово, 4 : Подвійне слово.	USINT	
Data	Значення параметра читання	UDINT	

Примітка: відповідний індекс і підіндекс параметра збереження

1. Визначений користувачем параметр - це параметр сервоприводу, який потрібно зчитати. Довжину вказують користувачі відповідно до типу даних параметра для читання. Довжина байтового параметра дорівнює 1. Довжина параметра слова дорівнює 2. Довжина параметра подвійного слова дорівнює 4. Обчислення індексу та субіндексу параметра сервоприводу показано наступним чином.

Індекс = Параметр сервоприводу (Hex) + 2000 (Hex)
Субіндекс = 0.

Приклад: Розрахунок індексу параметра сервоприводу P6-10: 2000 + 060A (шістнадцятковий вираз P6-10) = 260A, субіндекс = 0.

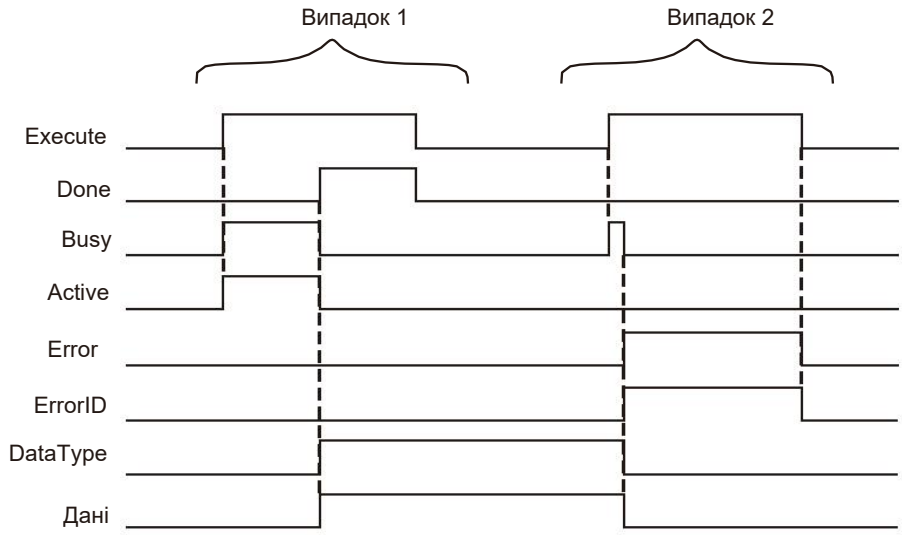


2. Для індексу та підіндексу інших підлеглих параметрів зверніться до посібника продукту, пов'язаного з функцією CANopen.

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли читання закінчено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *«Execute»* змінюється з FALSE на TRUE, *«Зайнятий»* і *«Активний»* змінюється на TRUE. Через один цикл *«Готово»* змінюється на TRUE, а *DataType* і *Data* показують відповідні значення даних. Після того, як *Done* змінюється на TRUE, *Busy* і *Active* змінюються на FALSE у тому самому циклі. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE, а *DataType* і *Data* зберігають вихідні значення. Якщо *помилка* змінюється на TRUE, значення *DataType* і *Data* буде скинуто до 0.

Випадок 2 : значення вхідного параметра є недопустимим, наприклад, номер осі: 0 перед виконанням інструкції DMC_ReadParameter_Motion. Коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється з FALSE на TRUE, а *ErrorID* відображає відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а вміст *ErrorID* очищається до 0.

- **функція**

DMC_ReadParameter_Motion зчитує значення підлеглого параметра. Користувачі можуть вказати індекс та підіндекс параметра, який потрібно прочитати.

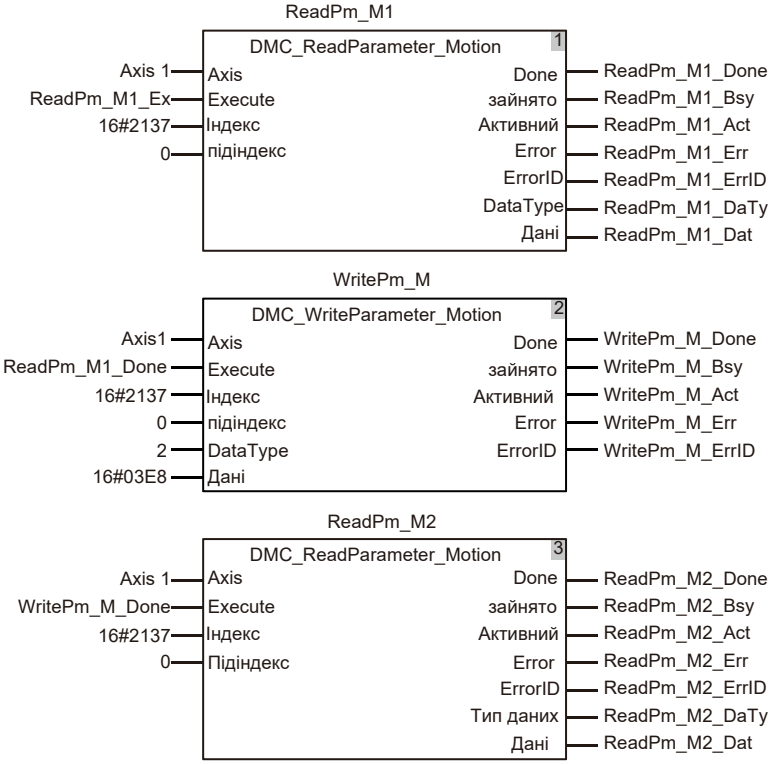
Приклад програмування

Приклад виконання інструкції DMC_ReadParameter_ Motion описується наступним чином.

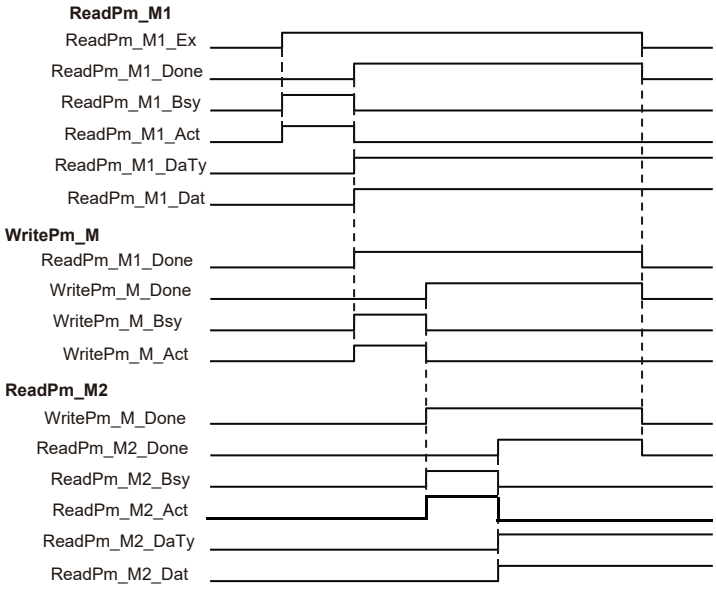
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ReadPm_M1	DMC_ReadParameter_ Motion	
Axis1	USINT	1
ReadPm_M1_Ex	BOOL	TRUE
ReadPm_M1_Done	BOOL	TRUE
ReadPm_M1_Bsy	BOOL	FALSE
ReadPm_M1_Act	BOOL	FALSE
ReadPm_M1_Err	BOOL	FALSE
ReadPm_M1_ErrID	WORD	FALSE
ReadPm_M1_DaTy	USINT	2
ReadPm_M1_Dat	UDINT	5000
WritePm_M	DMC_WriteParameter_Motion	
WritePm_M_Done	BOOL	TRUE
WritePm_M_Bsy	BOOL	FALSE
WritePm_M_Act	BOOL	FALSE

Ім'я змінної	Тип даних	Початкове значення
WritePm_M_Err	BOOL	FALSE
WritePm_M_ErrID	WORD	FALSE
ReadPm_M2	DMC_ReadParameter_ Motion	
ReadPm_M2_Done	BOOL	TRUE
ReadPm_M2_Bsy	BOOL	FALSE
ReadPm_M2_Act	BOOL	FALSE
ReadPm_M2_Err	BOOL	FALSE
ReadPm_M2_ErrID	WORD	FALSE
ReadPm_M2_DaTy	USINT	2
ReadPm_M2_Dat	UDINT	1000



2. Графік часу

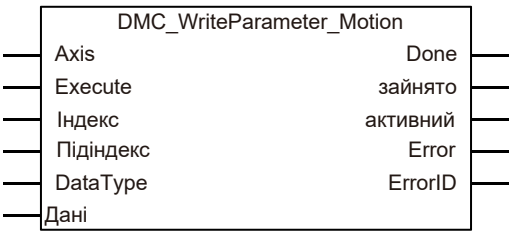


- Коли ReadPm_M1_Ex змінюється з FALSE на TRUE, починається виконання першої інструкції DMC_ReadParameter_Motion. Після завершення виконання інструкції ReadPm_M1_Done змінюється на TRUE, значення ReadPm_M1_DaTy дорівнює 2, а ReadPm_M1_Dat дорівнює 5000. Тобто вміст параметра сервоприводу P1-55, який зчитується, становить 5000 (максимальна швидкість сервоприводу обмежена 5000 об/хв.)
- Коли ReadPm_M1_Done змінюється з FALSE на TRUE, починається виконання інструкції DMC_WriteParameter_Motion. Після завершення виконання інструкції WritePm_M_Done змінюється на TRUE. Це означає, що запис 1000 до параметра веденого сервоприводу P1-55 успішний. (Максимальна швидкість сервоприводу обмежена 1000 об/хв)
- Коли WritePm_M_Done змінюється з FALSE на TRUE, починається виконання другої інструкції DMC_ReadParameter_Motion. Після завершення виконання інструкції ReadPm_M2_Done змінюється на TRUE, ReadPm_M2_DaTy дорівнює 2, а ReadPm_M2_Dat дорівнює 1000. Тобто вміст параметра підлеглого сервоприводу P1-55, який зчитується, дорівнює 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв. .)

11.3.20 DMC_WriteParameter_Motion

FB/FC	Пояснення	Застосовна модель
FB	DMC_WriteParameter_Motion встановлює значення підлеглого параметра.	AS516E-B AS532EST-B AS564EST-B

Позиція DMC_WriteParameter_Motion_in



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Axis	Укажіть підлеглий пристрій для керування.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
Індекс	Індекс параметра для запису.	UINT		
Підіндекс	Підіндекс параметра для запису.	USINT		
Тип даних	Тип даних параметра для запису 1 : байт, 2 : Слово, 4 : Подвійне слово.	USINT		
Дані	Значення параметра для запису	UDINT		

Примітки:

1. DataType має бути типом даних параметра для запису. Якщо заповнене значення неправильне, виникне помилка.
2. Зверніться до глави 9 для розрахунку індексу та субіндексу параметрів сервоприводу.

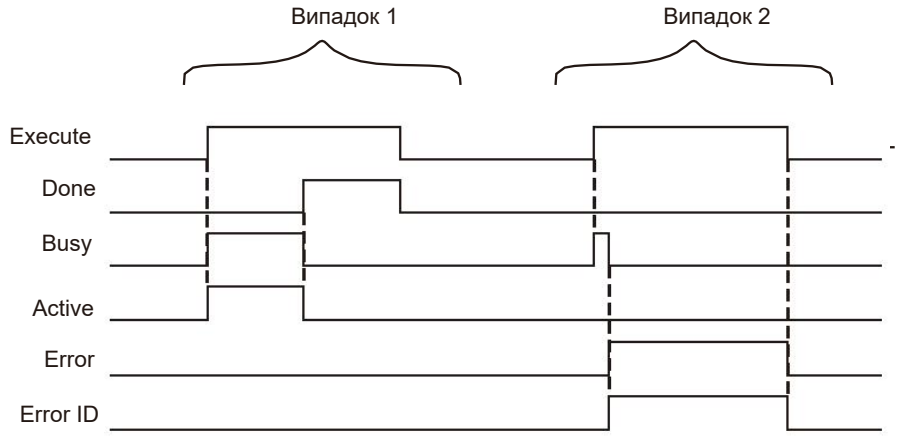
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли написання буде завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Done* змінюється з FALSE на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *«Execute»* змінюється з FALSE на TRUE, *«Зайнятий»* і *«Активний»* змінюється на TRUE. Через один цикл *«Готово»* змінюється на «TRUE». Після того, як *Done* змінюється на TRUE, *Busy* і *Active* змінюються на FALSE у тому самому циклі. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється з TRUE на FALSE.

Випадок 2 : значення вхідного параметра є недопустимим, наприклад, номер осі: 0 перед виконанням інструкції DMC_WriteParameter_Motion. Коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється з FALSE на TRUE, а *ErrorID* відображає відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а вміст *ErrorID* очищається до 0.

● **функція**

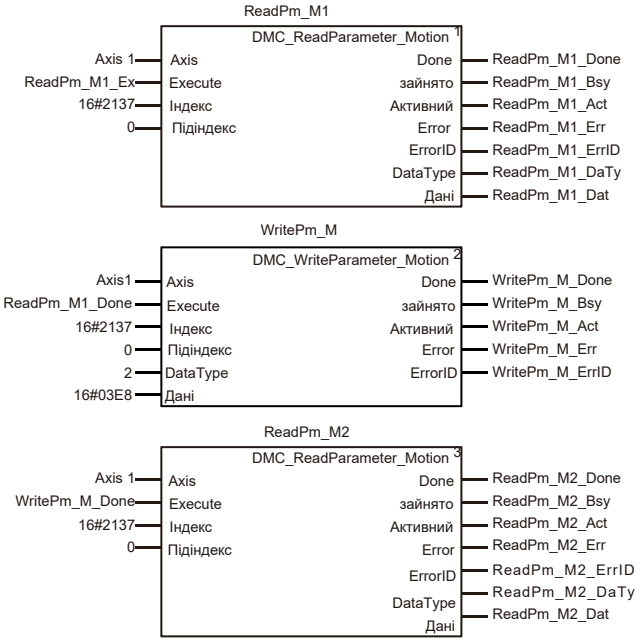
DMC_WriteParameter_Motion встановлює значення підлеглого параметра. Користувачі можуть вказати індекс та підіндекс параметра, який потрібно встановити.

Приклад програмування

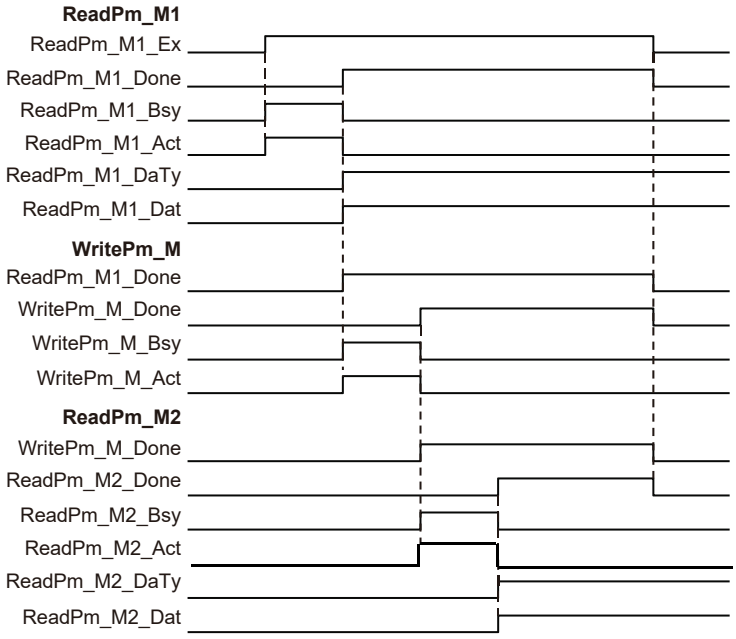
Приклад виконання інструкції DMC_WriteParameter_ Motion описується наступним чином.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ReadPm_M1	DMC_ReadParameter_Motion	
Axis1	USINT	1
ReadPm_M1_Ex	BOOL	TRUE
ReadPm_M1_Done	BOOL	TRUE
ReadPm_M1_Bsy	BOOL	FALSE
ReadPm_M1_Act	BOOL	FALSE
ReadPm_M1_Err	BOOL	FALSE
ReadPm_M1_ErrID	WORD	FALSE
ReadPm_M1_DaTy	USINT	2
ReadPm_M1_Dat	UDINT	5000
WritePm_M	DMC_WriteParameter_Motion	
WritePm_M_Done	BOOL	TRUE
WritePm_M_Bsy	BOOL	FALSE
WritePm_M_Act	BOOL	FALSE
WritePm_M_Err	BOOL	FALSE
WritePm_M_ErrID	WORD	FALSE
ReadPm_M2	DMC_ReadParameter_Motion	
ReadPm_M2_Done	BOOL	TRUE
ReadPm_M2_Bsy	BOOL	FALSE
ReadPm_M2_Act	BOOL	FALSE
ReadPm_M2_Err	BOOL	FALSE
ReadPm_M2_ErrID	WORD	FALSE
ReadPm_M2_DaTy	USINT	2
ReadPm_M2_Dat	UDINT	1000



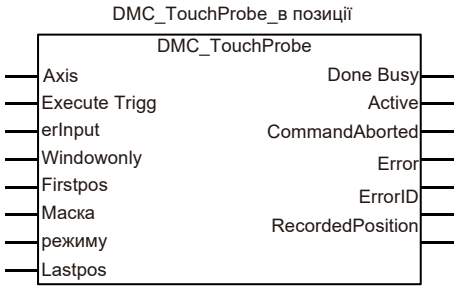
2. Графік часу



- Коли ReadPm_M1_Ex змінюється з FALSE на TRUE, починається виконання першої інструкції DMC_ReadParameter_Motion. Після завершення виконання інструкції ReadPm_M1_Done змінюється на TRUE, ReadPm_M1_DaTy дорівнює 2, а ReadPm_M1_Dat дорівнює 5000. Тобто вміст веденого параметра сервоприводу P1-55, який зчитується, дорівнює 5000. (Максимальна швидкість сервоприводу обмежена 5000 об/хв. .)
- Коли ReadPm_M1_Done змінюється з FALSE на TRUE, починається виконання DMC_WriteParameter_Motion. Коли виконання інструкції завершено, WritePm_M_Done змінюється на TRUE. Це означає, що встановлений параметр сервоприводу P1-55 становить 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв.)
- Коли WritePm_M_Done змінюється з FALSE на TRUE, починається виконання другої інструкції DMC_ReadParameter_Motion. Після завершення виконання інструкції ReadPm_M2_Done змінюється на TRUE, ReadPm_M2_DaTy дорівнює 2, а ReadPm_M2_Dat дорівнює 1000. Тобто вміст параметра підлеглого сервоприводу P1-55, який зчитується, дорівнює 1000. (Максимальна швидкість сервоприводу обмежена 1000 об/хв. .)

11.3.21 DMC_TouchProbe

FB/FC	Пояснення	Застосовна модель
FB	DMC_TouchProbe використовується для захоплення положення осі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
TriggerInput	Вкажіть одну з вхідних точок I0~I7, I10~I17 контролера руху як біт для запуску захоплення позиції. Значення параметра 0~15 відповідають вхідним точкам I0~I7 та I10~I17. Параметр дійсний, коли *Mode* дорівнює 0 і 1, і недійсний, коли *Mode* дорівнює 5, 6, 7 і 8.	MC_Triggerinput	0:mcTriggerinputI0 ... 7: mcTriggerinputI7 8:mcTriggerinputI10 ... 15: mcTriggerinputI17 (0)	
Тільки вікно	Зарезервований	-	-	-
Firststops	Зарезервований	-	-	-

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Ластопи	Зарезервований	-	-	-
Режим	Режим 0: тригерний сигнал надходить від наростаючого фронту вхідних точок: I0~I7 та I10~I17 контролера руху. Точка введення, яка використовується, визначається *TriggerInput* . Положення фіксується через наростаючий фронт біта тригера. Зафіксоване положення перетворюється з кількості імпульсів, які отримує порт зовнішнього кодера контролера через параметри осі. Режим 1: сигнал тригера надходить від спадного фронту однієї з точок входу: I0~I7 та I10~I17 контролера руху, який визначається *TriggerInput* . Зафіксоване положення перетворюється з кількості імпульсів, які отримує порт зовнішнього кодера контролера через параметри осі. Режим 5: тригерний сигнал надходить від наростаючого фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 6: тригерний сигнал надходить від спадного фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 7: Сигнал запуску надходить від наростаючого фронту та спадного фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 8: тригерний сигнал надходить від наростаючого фронту фази Z сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі.	INT		

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
	Для захоплення, яке виконується за допомогою точки входу сервоприводу, різні моделі сервоприводу відповідають різним точкам входу, як показано в таблиці нижче. Модель сервоприводу: A2-E | B3-E | A3-E Вхідна точка: DI13 | DI1 | DI1			
Маска	Зарезервований	-	-	-

Примітки:

1. У режимі 0 і режимі 1 одна і та ж точка введення не може використовуватися для запису позиції одночасно.
2. У режимі 5, режимі 6, режимі 7 і режимі 8 фіксація позиції не може виконуватися для однієї осі одночасно.

- **Вихідні параметри**

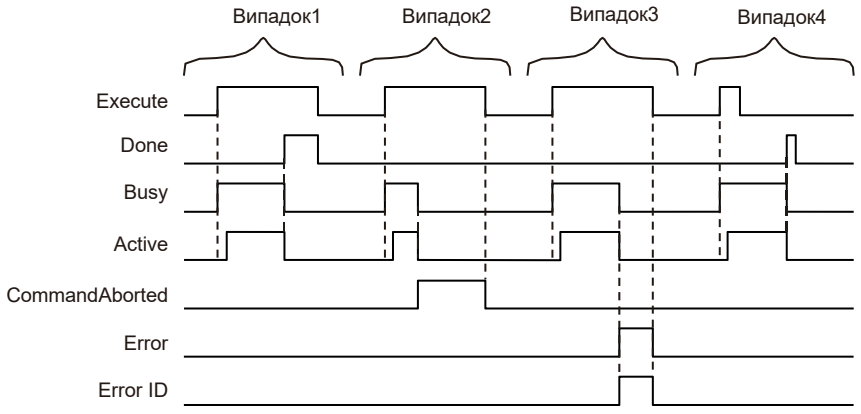
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	
RecordedPosition	Зафіксована позиція після завершення виконання інструкції. Додаткову інформацію див. у наступній функції.	LREAL	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли Execute змінюється з TRUE на FALSE
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Active	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли виконання інструкції переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і інструкція переривається іншою інструкцією, *Commandaborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : після того, як *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, коли виникає помилка, наприклад аварійний сигнал осі або Offline. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : під час виконання інструкції *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE. Тим часом *«Зайнятий»* і *«Активний»* змінюється на FALSE, а через період «Готово» змінюється на FALSE.

- **функція**
  - (*RecordedPosition*) положення, яке фіксує DMC_TouchProbe, перетворюється з іншого значення на основі параметрів осі. Джерела даних для перетворення перераховані в наведеній нижче таблиці.

Режим	Джерело даних
Режим 0 і режим 1	Кількість імпульсів, які отримує порт зовнішнього кодера контролера руху
Режим 5	Кількість імпульсів, які серводвигун повертає на сервопривод
Режим 6	Кількість імпульсів, які серводвигун повертає на сервопривод
Режим 7	Кількість імпульсів, які серводвигун повертає на сервопривод
Режим 8	Кількість імпульсів, які серводвигун повертає на сервопривод

  - Діапазон значень джерела даних -2147483648~2147483647. Коли значення джерела даних перевищує 2147483647, воно стане -2147483648. Зі зміною знака «+» або «–» значення джерела даних знак «+» або «-» значення *RecordedPosition* не зміниться, але значення *RecordedPosition* продовжуватиме збільшуватися.

  - Позиція, зафіксована інструкцією DMC_TouchProbe, обчислюється відповідно до параметрів осі. Для різних режимів джерела даних різні. «Налаштування передавального числа сервоприводу» та «Налаштування передавального числа механізму» в параметрах осі показано в наступній таблиці. Коли значення *режиму* інструкції дорівнює 5 (що можна послати до введення режиму 5 нижче), кількість імпульсів, які серводвигун подає назад до сервоприводу, становить 435, а позиція зафіксована

    за інструкцією 65,25. Формула розрахунку: 435× (3×1000) ÷ (2×10000) =65,25. 10000, 2, 3 і 1000 у формулі відповідають 10000, 2, 3 і 1000 у наступній таблиці відповідно. Для іншого режиму метод розрахунку для позиції, зафіксованої інструкцією, такий самий, як описано вище, але відрізняється лише джерело даних.

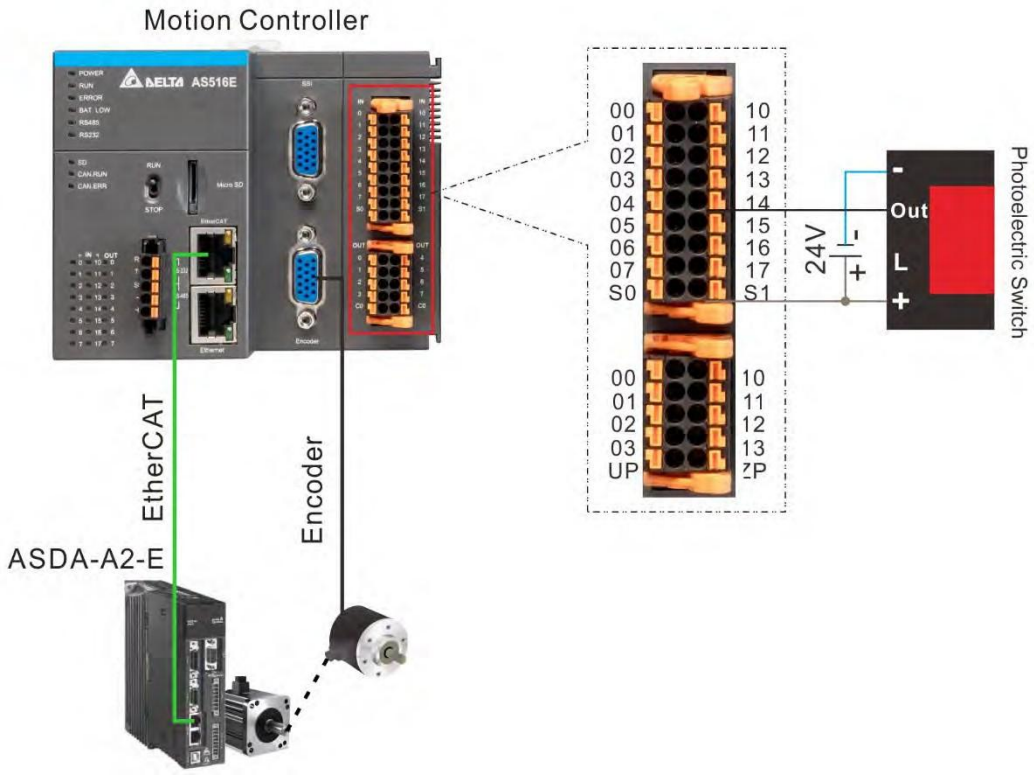
Налаштування передавального числа сервоприводу	Установка передавального числа механізму
Нумератор одиниці: 128	Вихідні оберти шестерні: 3
Знаменник одиниць: 1	Вхідні оберти шестерні: 2
Імпульсів на оберт: 10000	Одиниць на вихідний оберт: 1000 одиниць/оберт

  - Коли *Mode* =0 або 1 у DMC_TouchProbe, зафіксоване положення також може бути обчислено відповідно до методу, згаданого вище. У фактичному застосуванні фіксація положення зазвичай виконується шляхом побудови зовнішньої осі кодера. Коли кількість імпульсів, отриманих на зовнішньому інтерфейсі кодера контролера руху, становить 638, позиція, зафіксована DMC_TouchProbe, становить 95,4. Формула розрахунку: 638× (3×1000) ÷ (2×10000) =95,4. У формулі 1000 – це *одиниці на вихідний оберт* , 2 – *вхідний оберт шестерні* ; 3 — *вихідна кількість обертів шестерні* , а 10000 — *кількість імпульсів на один оберт*). Коли I0 один раз змінюється з FALSE на TRUE, запис позиції виконується один раз.

- **Рисунок проводки**
  - **Режим 0 і режим 1**

Режим 0: Зовнішній сигнал запускає точку I контролера руху, а захоплення позиції здійснюється через наростаючий фронт точки введення, визначеної *TriggerInput* . Зафіксоване положення перетворюється з кількості імпульсів, які порт зовнішнього кодера контролера отримує через параметри осі.

Режим 1: Зовнішній сигнал запускає точку I контролера руху, а захоплення позиції здійснюється через спадаючий фронт точки введення, визначеної *TriggerInput* . Зафіксоване положення перетворюється з кількості імпульсів, які порт зовнішнього кодера контролера отримує через параметри осі.



■ **Режим 5, 6 і 7**

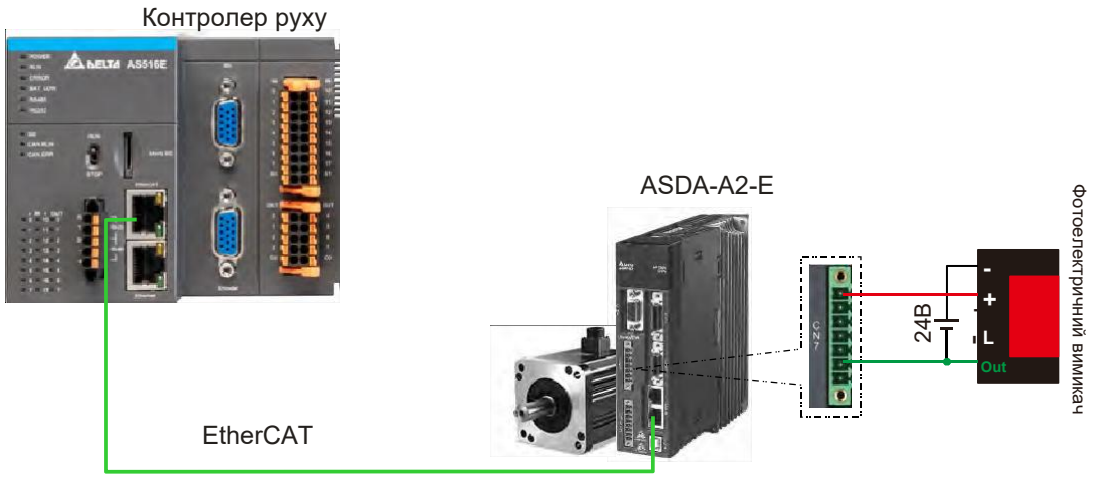
Режим 5 : сигнал запуску надходить від наростаючого фронту DI13 розширювальних DI сервоприводу CN7.

Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі.

Режим 6 : Сигнал тригера надходить від спадного фронту DI13 розширювальних DI сервоприводу CN7.

Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі.

Режим 7 : тригерний сигнал надходить від наростаючого та спадаючого фронтів DI13 сервоприводу Розширення CN7 DI. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Позиція, зафіксована на спаді, отримується через змінну *FallingPosition* імені екземпляра. Форма «Назва екземпляра. FallingPosition”.



■ **Режим 8**

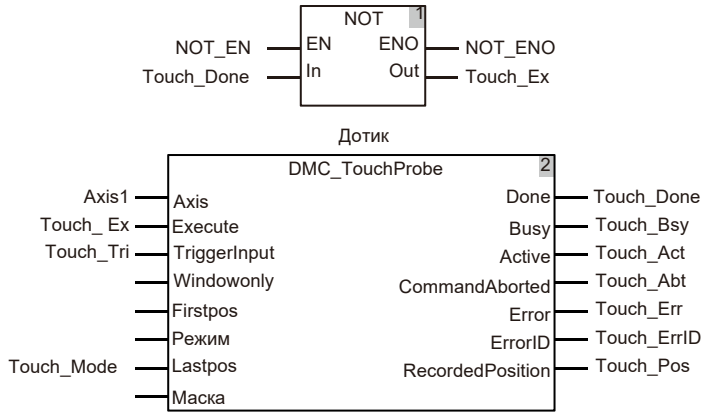
Тригерний сигнал надходить від наростаючого фронту фази Z сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі.

Приклад програмування 1

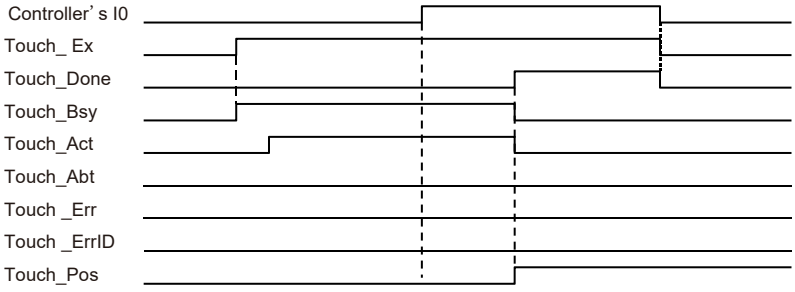
Зафіксуйте положення осі зовнішнього датчика за допомогою наростаючого фронту I0 у режимі 0.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
NOT_EN	BOOL	FALSE
NOT_ENO	BOOL	
Touch	DMC_TouchProbe	
Axis1	USINT	3
Touch_Ex	BOOL	FALSE
Touch_Tri	MC_Triggerinput	0
Touch_Mode	INT	0
Touch_Done	BOOL	
Touch_Bsy	BOOL	
Touch_Act	BOOL	
Touch_Abt	BOOL	
Touch_Err	BOOL	
Touch_ErrID	UINT	
Touch_Pos	LREAL	



2. Графік часу



- ❖ Коли Touch_Ex змінюється з FALSE на TRUE, Touch_Bsy змінюється з FALSE на TRUE у першому циклі, а Touch_Act змінюється з FALSE на TRUE у другому циклі.
- ❖ Коли зовнішній сигнал запускає I0 контролера, DMC_TouchProbe починає виконуватися. Коли Touch_Done змінюється з FALSE на TRUE, вихідна позиція Touch_Pos перетворюється з кількості імпульсів, які порт зовнішнього кодера контролера отримує через параметри осі. Тим часом Touch_Bsy і Touch_Act змінюються з TRUE на FALSE. Коли Touch_Ex змінюється з TRUE на FALSE, Touch_Done змінюється з TRUE на FALSE, а позиція, яку фіксує Touch_Pos, не буде скинута до 0.

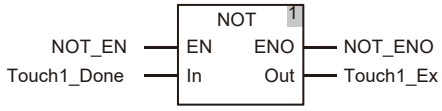
Приклад програмування 2

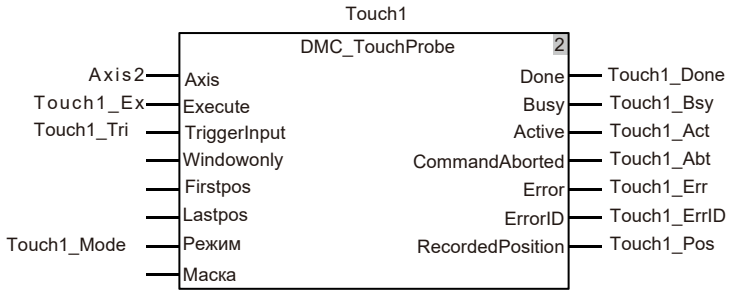
Зовнішній сигнал запускає DI13 розширювальних DI сервоприводу CN7 у режимі 5. Зафіксуйте положення, яке перетворюється з кількості імпульсів, які серводвигун подає назад до сервоприводу через параметри осі.

1. Таблиця змінних і програма

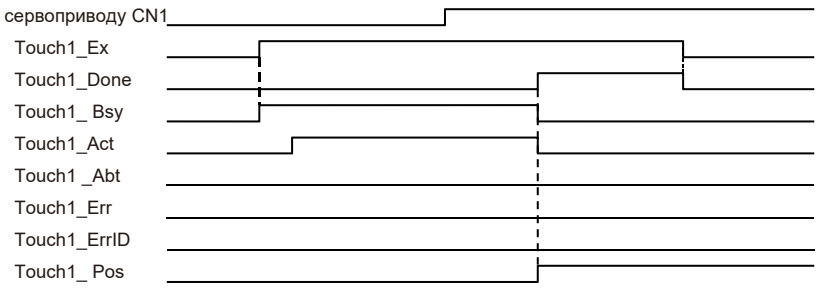
Ім'я змінної	Тип даних	Початкове значення
NOT_EN	BOOL	FALSE
NOT_ENO	BOOL	
Touch1	DMC_TouchProbe	
Axis2	USINT	1
Touch1_Ex	BOOL	FALSE
Touch1_Tri	MC_Triggerinput	
Touch1_Mode	INT	5
Touch1_Done	BOOL	
Touch1_Bsy	BOOL	
Touch1_Act	BOOL	
Touch1_Abt	BOOL	
Touch1_Err	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Touch1_ErrID	UINT	
Touch1_Pos	LREAL	





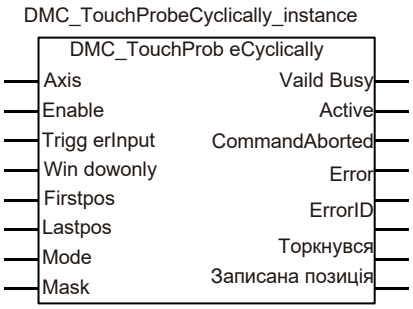
2. Графік часу



- ❖ Коли Touch1_Ex змінюється з FALSE на TRUE, Touch1_Bsy змінюється з FALSE на TRUE у першому циклі, а Touch1_Act змінюється з FALSE на TRUE у другому циклі.
- ❖ Коли виконання DMC_TouchProbe завершується після того, як зовнішній сигнал запускає DI13 розширення CN7 сервоприводу, Touch1_Done змінюється з FALSE на TRUE, а Touch1_Pos виводить положення, перетворене з кількості імпульсів, які серводвигун повертає на сервопривід відповідно до осі параметри. Тим часом Touch1_Bsy і Touch1_Act змінюються з TRUE на FALSE. Коли Touch1_Ex змінюється з TRUE на FALSE, Touch1_Done змінюється з TRUE на FALSE, а позиція, яку фіксує Touch1_Pos, не скидається на 0.

11.3.22 DMC_TouchProbeCyclically

FB/FC	Пояснення	Застосовна модель
FB	DMC_TouchProbeCyclically використовується для циклічного захоплення положення осі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Enable	Інструкція вмикається, якщо *ввімкнути* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
TriggerInput	Вкажіть одну з вхідних точок I0~I7, I10~I17 контролера руху як біт для запуску захоплення позиції. Значення параметра 0~15 відповідають вхідним точкам I0~I7 та I10~I17. Параметр дійсний, коли *Mode* дорівнює 0 і 1, і недійсний, якщо *Mode* дорівнює 5, 6, 7 і 8.	MC_Triggerinput	0: mcTriggerinputI0 ... 7: mcTriggerinputI7 8: mcTriggerinputI10 ... 15: mcTriggerinputI17 (0)	
Тільки вікно	Зарезервований	-	-	-
Firststops	Зарезервований	-	-	-
Ластопи	Зарезервований	-	-	-

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Режим	Режим 0: тригерний сигнал надходить від наростаючого фронту вхідних точок: I0~I7 та I10~I17 контролера руху. Точка введення, яка використовується, визначається *TriggerInput* . Положення фіксується через наростаючий фронт біта тригера. Зафіксоване положення перетворюється з кількості імпульсів, які отримує порт зовнішнього кодера контролера через параметри осі. Режим 1: сигнал тригера надходить від спадного фронту однієї з точок входу: I0~I7 та I10~I17 контролера руху, який визначається *TriggerInput* . Зафіксоване положення перетворюється з кількості імпульсів, які отримує порт зовнішнього кодера контролера через параметри осі. Режим 5: тригерний сигнал надходить від наростаючого фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 6: тригерний сигнал надходить від спадного фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 7: Сигнал запуску надходить від наростаючого фронту та спадного фронту вхідної точки сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Режим 8: тригерний сигнал надходить від наростаючого фронту фази Z сервоприводу. Зафіксоване положення перетворюється з кількості імпульсів, які серводвигун повертає на сервопривід через параметри осі. Для циклічного захоплення, яке виконується за допомогою точки введення а	INT		

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
	сервопривод, різні моделі сервоприводів відповідають різним вхідним точкам, як показано в таблиці нижче.			
	Модель сервоприводу: A2-E / B3-E / A3-E			
	Вхідна точка: DI13 / DI1 / DI1			
Маска	Зарезервований	-	-	-

Примітки:

1. У режимі 0 і режимі 1 одна і та ж точка введення не може використовуватися для запису позиції одночасно.
2. У режимі 5, режимі 6, режимі 7 і режимі 8 фіксація позиції не може виконуватися для однієї осі одночасно.

● **Вихідні параметри**

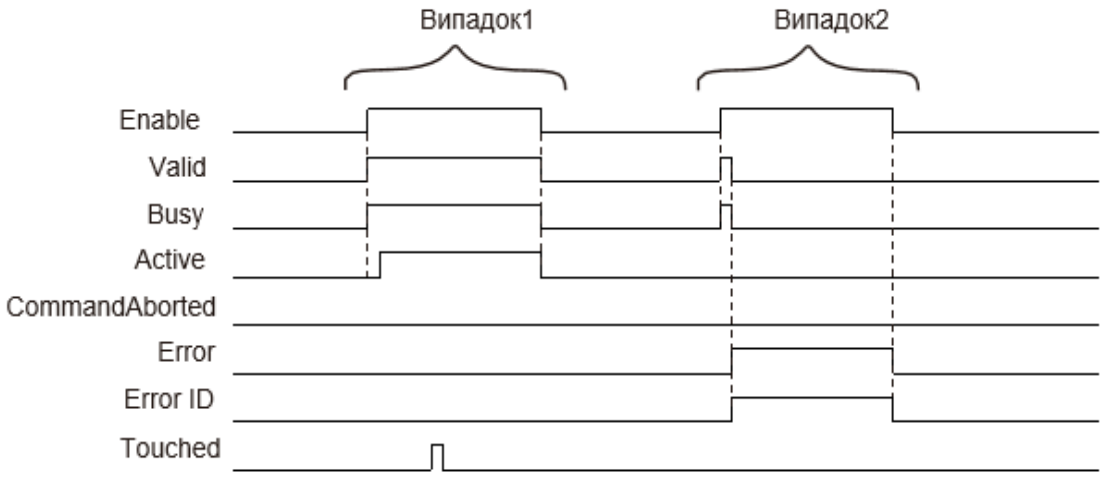
Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли отримане значення є дійсним.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	
Зворушений	TRUE, коли одне захоплення завершується інструкцією.		
RecordedPosition	Зафіксована позиція після завершення виконання інструкції.	LREAL	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли інструкція виконується нормально, коли *Enable* має значення TRUE.	◆ Коли Execute змінюється з TRUE на FALSE
Busy	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Active	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли виконання інструкції переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE. Через один період *активність* змінюється на TRUE. Після завершення одного захоплення *Touched* змінюється на TRUE, а стан TRUE триватиме один період. Коли *Enable* змінюється з TRUE на FALSE, *Valid* , *Busy* і *Active* змінюються на FALSE.

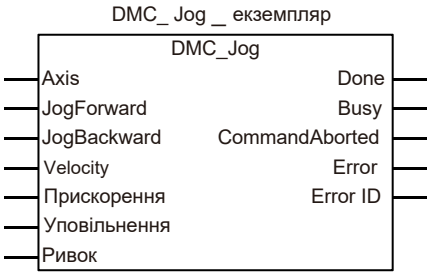
Випадок 2 : коли існує помилка у вхідних параметрах і *Enable* змінюється з FALSE на TRUE, *Busy* і *Valid* змінюються на TRUE. Через один період *помилка* змінюється на TRUE, а *ErrorID* відображає відповідні коди помилок, а тим часом *Busy* і *Valid* змінюються на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* буде видалено.

● Пояснення функції

DMC_TouchProbeCyclically використовується для циклічного захоплення положення осі. Функція схожа на функцію DMC_TouchProbe. Різниця полягає в тому, що положення осі може циклічно фіксуватися інструкцією, коли *Enable* змінюється на TRUE. Тобто, як тільки зовнішній сигнал виявлено, інструкція негайно виконує захоплення позиції без необхідності повторного запуску.
Зверніться до DMC_TouchProbe , щоб дізнатися більше про функції.

11.3.23 DMC_Jog

FB/FC	Пояснення	Застосовна модель
FB	DMC_Jog рухається вісь.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *JogForward* або *JogBackward* має значення TRUE
JogForward	Інструкція виконується, коли *JogForward* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
JogBackward	Інструкція виконується, коли *JogBackward* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/секунда)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *JogForward* або *JogBackward* має значення TRUE
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *JogForward* або *JogBackward* має значення TRUE
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *JogForward* або *JogBackward* має значення TRUE
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *JogForward* або *JogBackward* має значення TRUE

- **Вихідні параметри**

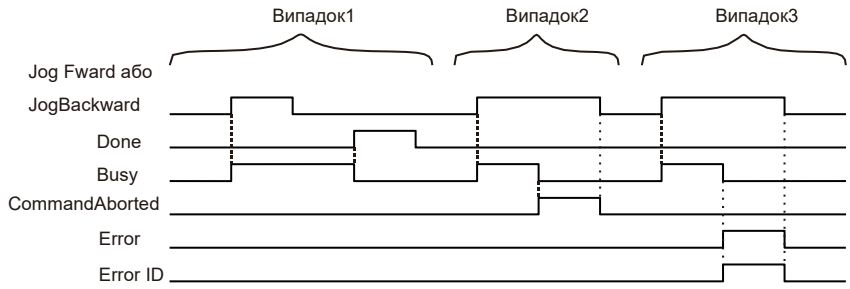
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли біг підтюпцем закінчено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
CommandAborted	TRUE, коли інструкцію перервано. .	BOOL	TRUE/FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE/FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

● **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли біг припиняється.	◆ Через один період після припинення бігу.
Busy	◆ Коли *JogForward* або *JogBackward* змінюється на TRUE.	◆ Коли *Done* змінюється на TRUE, ◆ Коли *помилка* змінюється на TRUE, ◆ Коли *CommandAborted* змінюється на TRUE.
CommandAborted	◆ TRUE, коли інструкція переривається іншою інструкцією руху.	◆ Коли *JogForward* або *JogBackward* змінюється з TRUE на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *JogForward* або *JogBackward* змінюється з TRUE на FALSE.

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *JogForward* або *JogBackward* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли біг припиняється, швидкість осі зменшується до 0, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE.

Випадок 2 : коли *JogForward* або *JogBackward* змінюється з FALSE на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *Busy* змінюється на FALSE після того, як інструкцію перервано іншою інструкцією руху. *CommandAborted* змінюється на FALSE, коли *JogForward* або *JogBackward* змінюється з TRUE на FALSE.

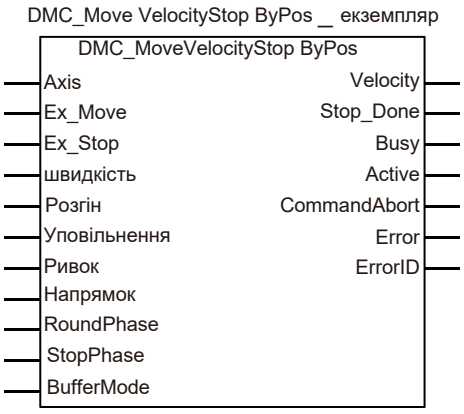
Випадок 3 : після того, як *JogForward* або *JogBackward* зміниться з FALSE на TRUE, *Error* зміниться на TRUE, а *ErrorID* відобразить відповідні коди помилок, якщо станеться будь-яка помилка, наприклад сигнал тривоги осі або Offline. Тим часом *Busy* зміниться на FALSE. *Error* зміниться на FALSE, а значення *ErrorID* буде скинуто до 0, коли *JogForward* або *JogBackward* зміниться з TRUE на FALSE.

- **функція**

  DMC_Jog рухається вісь. Параметр *JogForward* керує рухом осі вперед, а параметр *JogBackward* керує рухом осі назад. Коли біг припиняється, *Done* змінюється на TRUE, а через один цикл змінюється на FALSE. Тим часом *Busy* змінюється на FALSE.

11.3.24 DMC_MoveVelocityStopByPos

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveVelocityStopByPos використовується для зупинки осі на вказаній фазі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Ex_Move	Інструкція керує віссю, яка запускається, коли *Ex_Move* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Ex_Stop	Інструкція керує віссю, яка запускається, коли *Ex_Stop* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/секунда)	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Напрямок	Вкажіть напрямок обертання 1: Позитивний напрямок 3: Негативний напрямок	MC_Direction	1: mcPositiveDirection, 3: mcNegativeDirection (1)	Коли *Ex_Move* змінюється на TRUE.
RoundPhase	Укажіть модуль, якому відповідає крок (***UnitsPerRotation).***	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
StopPhase	Укажіть фазу в модулі.	LREAL	0 ~ значення налаштування *RoundPhase* (0)	Коли *Ex_Move* змінюється на TRUE.
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered 2 : mcBlending Низький 3 : mcBlending _Попередній 4 : mcBlending _Наступний 5 : mcBlending _Високий (0)	Коли *Ex_Move* змінюється на TRUE.

● **Вихідні параметри**

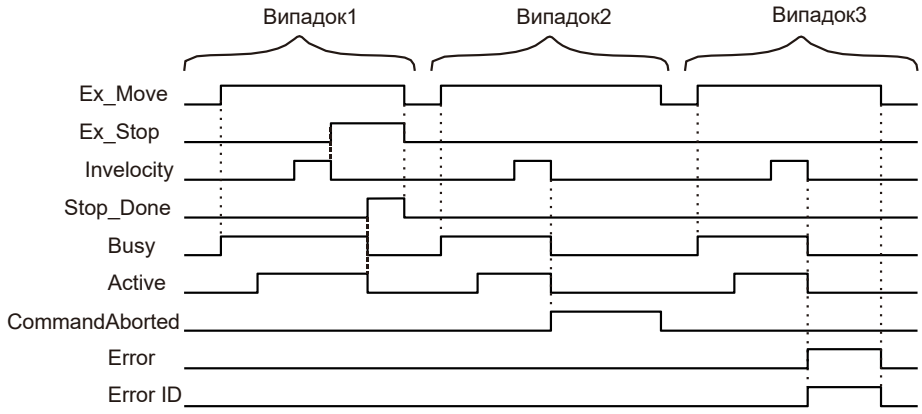
Назва параметра	функція	Тип даних	Допустимий діапазон
Invelocity	TRUE, коли досягнуто цільової швидкості.	BOOL	TRUE / FALSE
Stop_Done	TRUE, коли досягнуто положення зупинки.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується інструкцією.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Invelocity	◆ При досягненні цільової швидкості.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ *Invelocity* змінюється на FALSE негайно, коли *Ex_Move* знову змінюється з FALSE на TRUE, якщо значення вхідних параметрів переглядаються після досягнення цільової швидкості. Якщо значення вхідних параметрів не змінюються після завершення виконання інструкції, *Invelocity* змінюється на FALSE негайно, коли *Ex_Move* знову змінюється з FALSE на TRUE. *Invelocity* зміниться на TRUE у наступному циклі.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Stop_Done	◆ Коли досягнуто положення зупинки	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Ex_Stop* змінюється на FALSE.
Busy	◆ Коли *Ex_Move* змінюється на TRUE.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли вісь керується інструкцією.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ TRUE, коли інструкція переривається іншою інструкцією руху.	◆ Коли Ex_Move змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається після зміни *Ex_Move* та *Ex_Stop* з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Ex_Move* і *Ex_Stop* змінюються з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Ex_Move* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли вісь починає керуватися інструкцією, *Active* змінюється на TRUE. Коли *Ex_Stop* змінюється з FALSE на TRUE, *Invelocity* змінюється на FALSE. Коли позиція досягнута, *Stop_Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Ex_Move* та *Ex_Stop* змінюються з TRUE на FALSE, *Stop_Done* змінюється на FALSE.

Випадок 2 : коли *Ex_Move* змінюється на TRUE, інструкція переривається іншою інструкцією, а *CommandAborted* змінюється на TRUE. Тим часом *Invelocity* , *Busy* і *Active* змінюються на FALSE . Коли *Ex_Move* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли виникає помилка, коли *Ex_Move* має значення TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Invelocity* , *Busy* і *Active* змінюються на FALSE. *Error* змінюється на FALSE, коли *Ex_Move* змінюється з TRUE на FALSE.

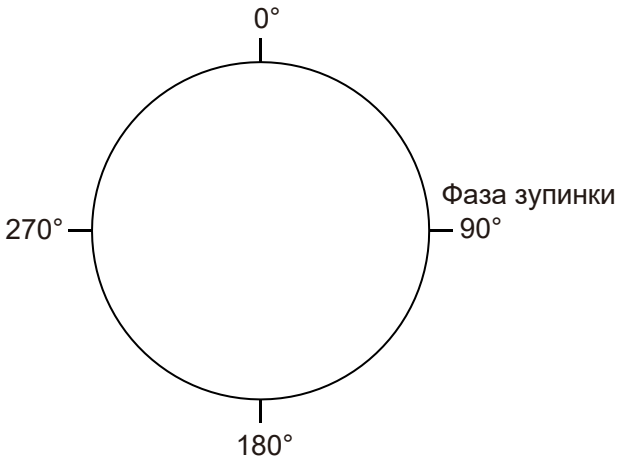
- **функція**

DMC_MoveVelocityStopByPos використовується для зупинки осі на визначеній фазі. ***RoundPhase*** — це модуль, якому відповідає крок (***UnitsPerRotation). StopPhase*** вказує фазу в модулі. Значення ***StopPhase*** має бути меншим за значення ***RoundPhase*** .

Коли ***Ex_Move*** змінюється з FALSE на TRUE, вісь керується для запуску. Коли ***Ex_Stop*** змінюється з FALSE на TRUE, вісь керується для зупинки на фазі, визначеній ***StopPhase*** . Кінцева позиція, де зупиняється вісь, дорівнює цілому кратному кроку (***UnitsPerRotation***) + значення ***StopPhase*** / ***RoundPhase*** значення* значення кроку (***UnitsPerRotation***).

Крок (***UnitsPerRotation***) становить 10000, ***RoundPhase*** встановлено на 360, а ***StopPhase*** встановлено на 90, як показано на малюнку нижче. Крім того, вісь може зупинятися на фазі, визначеній ***StopPhase,*** за допомогою інструкції DMC_MoveVelocityStopByPos.

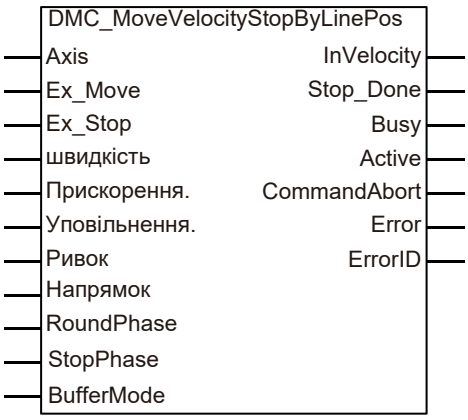
Термінальний привод може зупинитися в положенні 12500 одиниць, 22500 одиниць, 32500 одиниць або 42500 одиниць.



11.3.25 DMC_MoveVelocityStopByLinePos

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveVelocityStopByLinePos використовується для зупинки осі у вказаній позиції.	AS516E-B AS532EST-B AS564EST-B

DMC_MoveVelocity StopBy LinePos _ екземпляр



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Ex_Move	Інструкція керує віссю, яка запускається, коли *Ex_Move* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Ex_Stop	Ця інструкція керує віссю, яка запускається, коли *Ex_Stop* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
швидкість	Вкажіть цільову швидкість. (Одиниця: одиниця/секунда)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
ривок	Вкажіть швидкість зміни цільового прискорення або уповільнення. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Напрямок	Вкажіть напрямок обертання 1: Позитивний напрямок 3: Негативний напрямок	MC_Direction	1: mcPositiveDirection, 3: mcNegativeDirection (1)	Коли *Ex_Move* змінюється на TRUE.
RoundPhase	Укажіть модуль.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Ex_Move* змінюється на TRUE.
StopPhase	Укажіть позицію в модулі.	LREAL	0 ~ значення налаштування *RoundPhase* (0)	Коли *Ex_Move* змінюється на TRUE.
BufferMode	Укажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано 2: BlendingLow 3: BlendingPrevious 4: BlendingNext 5: BlendingHigh	MC_Buffer_Mode	0 : mcAborting 1 : mcBuffered 2 : mcBlendingLow 3 : mcBlending_Попередній 4 : mcBlending_Далі 5 : mcBlending_Високий (0)	Коли *Ex_Move* змінюється на TRUE.

- **Вихідні параметри**

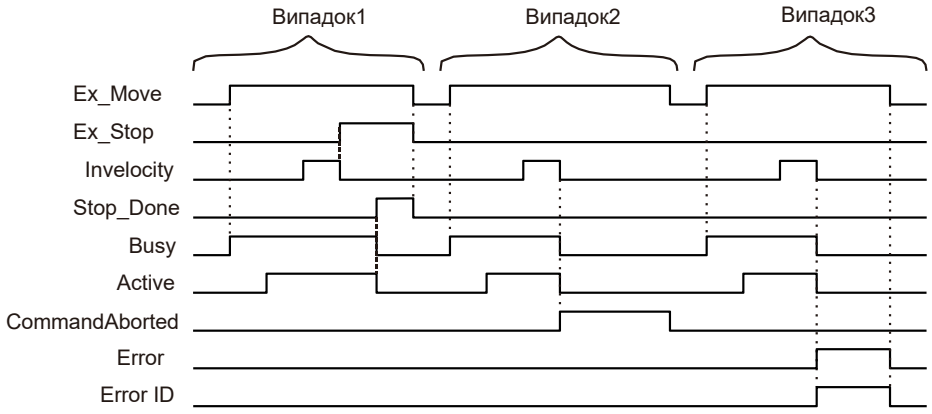
Назва параметра	функція	Тип даних	Допустимий діапазон
Invelocity	TRUE, коли досягнуто цільової швидкості.	BOOL	TRUE / FALSE
Stop_Done	TRUE, коли досягнуто положення зупинки.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується інструкцією.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Invelocity	◆ При досягненні цільової швидкості.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ *Invelocity* змінюється на FALSE негайно, коли *Ex_Move* знову змінюється з FALSE на TRUE, якщо значення вхідних параметрів переглядаються після досягнення цільової швидкості. Якщо значення вхідних параметрів не змінюються після завершення виконання інструкції, *Invelocity* змінюється на FALSE негайно, коли *Ex_Move* знову змінюється з FALSE на TRUE. *Invelocity* зміниться на TRUE у наступному циклі.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Stop_Done	◆ Коли досягнуто положення зупинки	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Ex_Stop* змінюється на FALSE.
Busy	◆ Коли *Ex_Move* змінюється на TRUE.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли вісь керується інструкцією.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ TRUE, коли інструкція переривається іншою інструкцією руху.	◆ Коли Ex_Move змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлюється на TRUE, коли інструкцію переривається після зміни *Ex_Move* та *Ex_Stop* з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Ex_Move* і *Ex_Stop* змінюються з TRUE на FALSE.

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Ex_Move* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли вісь починає керуватися інструкцією, *Active* змінюється на TRUE. Коли *Ex_Stop* змінюється з FALSE на TRUE, *Invelocity* змінюється на FALSE. Коли позиція досягнута, *Stop_Done* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Ex_Move* та *Ex_Stop* змінюються з TRUE на FALSE, *Stop_Done* змінюється на FALSE.

Випадок 2 : коли *Ex_Move* змінюється на TRUE, інструкція переривається іншою інструкцією, а *CommandAborted* змінюється на TRUE. Тим часом *Invelocity* , *Busy* і *Active* змінюються на FALSE . Коли *Ex_Move* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли виникає помилка, коли *Ex_Move* має значення TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Invelocity* , *Busy* і *Active* змінюються на FALSE. *Error* змінюється на FALSE, коли *Ex_Move* змінюється з TRUE на FALSE.

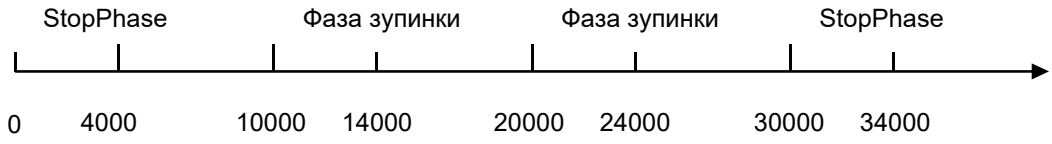
- **функція**

DMC_MoveVelocityStopByLinePos використовується для зупинки осі в заданому положенні. ***RoundPhase*** встановлює вказаний модуль. ***StopPhase*** встановлює позицію за вказаним модулем. Значення ***StopPhase*** має бути меншим за значення ***RoundPhase.*** Їхні одиниці є такими ж, як і для кроку (***UnitsPerRotation***).

Коли ***Ex_Move*** змінюється з FALSE на TRUE, вісь керується інструкцією. Коли ***Ex_Stop*** змінюється з FALSE на TRUE, вісь контролюється для зупинки в положенні, визначеному ***StopPhase*** . Кінцева позиція, де зупиняється вісь, є позицією цілого кратного значення ***RoundPhase*** + значення ***StopPhase*** .

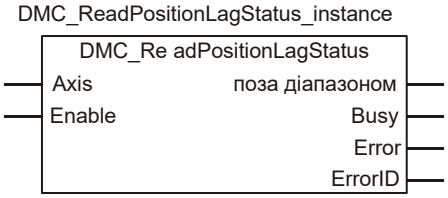
Для RoundPhase встановлено значення 10000, а ***для StopPhase*** – 4000, як показано на малюнку нижче. Приводом терміналу можна керувати так, щоб він зупинявся в положенні, визначеному ***StopPhase ,*** за допомогою інструкції DMC_MoveVelocityStopByLinePos.

Термінальний привід може зупинитися в положенні 4000 одиниць, 14000 одиниць, 24000 одиниць або 34000 одиниць.



11.3.26 DMC_ReadPositionLagStatus

FB/FC	Пояснення	Застосовна модель
FB	DMC_ReadPositionLagStatus використовується для виявлення затримки позиції.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* має значення TRUE.

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
За межою	TRUE, коли затримка позиції виходить за допустимий діапазон.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
За межою	◆ Коли абсолютне значення різниці фактичного положення вказаної осі перевищує встановлене значення *Lag* протягом періоду *HoldTime , оскільки воно* вперше перевищило значення *Lag* .	◆ Абсолютне значення різниці фактичного положення вказаної осі менше або дорівнює значенню *Lag* . ◆ Коли *Enable* має значення FALSE. ◆ Коли *помилка* змінюється на TRUE.
Busy	◆ Коли інструкція виконується.	◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли проблема вирішена.

● функція

DMC_ReadPositionLagStatus використовується для виявлення того, чи перевищує абсолютне значення різниці фактичних позицій зазначеної осі (яка є різницею між командною позицією та позицією зворотного зв'язку) значення налаштування. Дозволене значення різниці позицій встановлюється в інструкції DMC_WritePositionLagSetting.

OutOfRange зміниться на TRUE, якщо фактична різниця положення вказаної осі перевищувала значення *Lag* протягом періоду, визначеного *HoldTime ,* після того, як значення *Lag* було перевищено вперше. (*Lag* і *HoldTime* є входами DMC_WritePositionLagSetting.) Вихід *OutOfRange* зміниться на FALSE, коли абсолютне значення різниці фактичного положення вказаної осі менше або дорівнює значенню *Lag* .

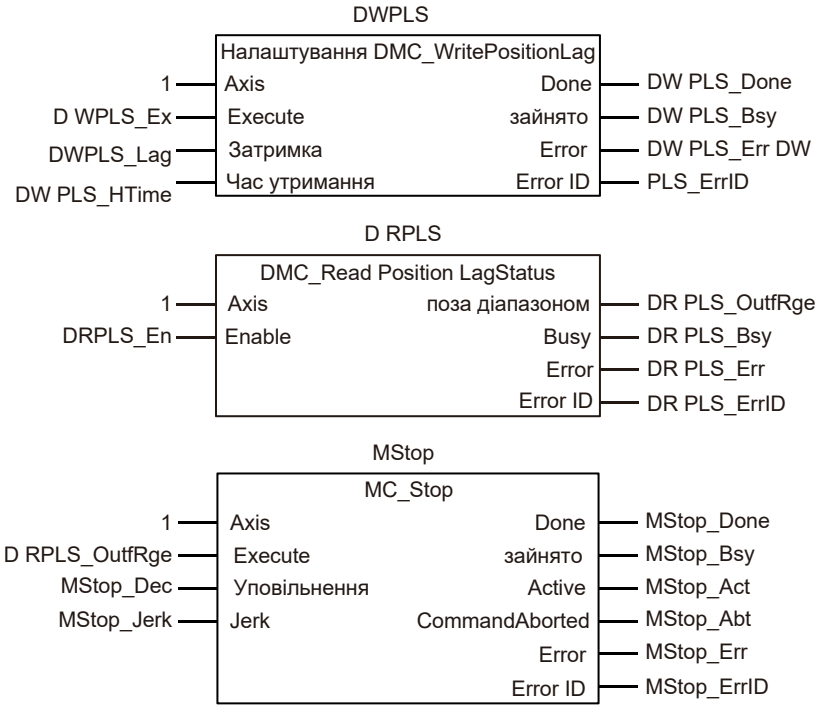
Коли значення параметра *HoldTime* параметра DMC_WritePositionLagSetting дорівнює 0, вихід *OutOfRange* зміниться на TRUE, як тільки інструкція виявить, що різниця фактичного положення вказаної осі перевищує значення *Lag* , а вихід *OutOfRange* зміниться на FALSE, як тільки інструкція виявить, що абсолютна значення різниці фактичного положення вказаної осі менше або дорівнює значенню *Lag* .

Вісь не зупиниться, якщо інструкція виявить, що фактична різниця позицій зазначеної осі перевищує значення *Lag* . Рух осі можна зупинити, запустивши виконання інструкції MC_Stop або MC_Halt через вихід *OutOfRange* .

Приклад програмування

1. **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
DWPLS	DMC_WritePositionLagSetting	
DWPLS_Ex	BOOL	
DWPLS_Lag	LREAL	
DWPLS_H_Time	LREAL	
DWPLS_Done	BOOL	
DWPLS_Bsy	BOOL	
DWPLS_Err	BOOL	
DWPLS_ErrID	WORD	
DRPLS	DMC_ReadPositionLagStatus	
DRPLS_OutfRge	BOOL	
DRPLS_Bsy	BOOL	
DRPLS_Err	BOOL	
DRPLS_ErrID	WORD	
MStop	MC_Stop	
Mstop_DEC	LREAL	
Mstop_Jerk	LREAL	
Mstop_Done	BOOL	
Mstop_Bsy	BOOL	
Mstop_Act	BOOL	
Mstop_Abt	BOOL	
Mstop_Err	BOOL	
Mstop_ErrID	WORD	

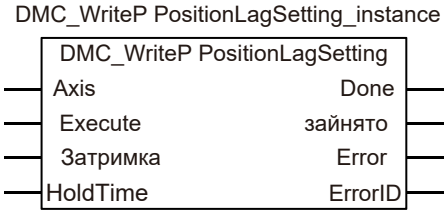


❖ Коли DWPLS_Ex змінюється з FALSE на TRUE, установіть значення різниці позицій, визначене *Lag* , і значення тривалості, визначене *HoldTime* для вказаної осі. Коли DWPLS_Done змінюється на TRUE, це означає, що запис параметрів завершено.

❖ Коли DRPLS_En змінюється на TRUE, інструкція DMC_ReadPositionLagStatus починає визначати, чи перевищує фактична різниця позицій зазначеної осі дозволене значення, встановлене в інструкції DMC_WritePositionLagSetting.
DRPLS_OutfRge змінюється на TRUE, коли інструкція DMC_ReadPositionLagStatus виявляє, що фактична різниця позицій зазначеної осі перевищує значення налаштування, а потім продовжує перевищувати значення налаштування протягом встановленого періоду часу.
Вісь може зупинитися, ініціювавши виконання інструкції MC_Stop через вихід DRPLS_OutfRge.
DRPLS_OutfRge змінюється на FALSE, коли інструкція DMC_ReadPositionLagStatus виявляє, що фактична різниця положення вказаної осі менша за значення параметра.

11.3.27 DMC_WritePositionLagSetting

FB/FC	Пояснення	Застосовна модель
FB	DMC_WritePositionLagSetting використовується для встановлення затримки позиції.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
відставання	Встановіть допустиме значення різниці між командною позицією та позицією зворотного зв'язку.	LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Час утримання	Період часу, протягом якого набір *затримки* перевищено. (Одиниця: секунда)	LREAL	0 або додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE/FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE. ◆ Коли *помилка* змінюється на TRUE.
Busy	◆ Коли інструкція виконується.	◆ Коли *помилка* змінюється на TRUE.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли помилка очищена.

● функція

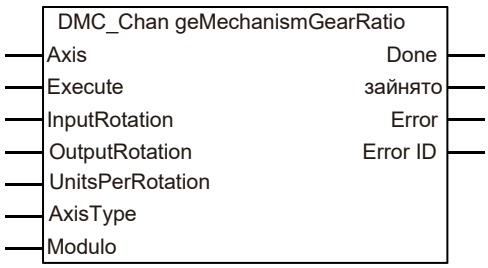
DMC_WritePositionLagSetting використовується для встановлення дозволеного значення затримки позиції (яке є різницею між позицією команди та позицією зворотного зв'язку) і дозволеного періоду часу, визначеного *HoldTime , протягом якого* перевищено значення налаштування *Lag* . Інструкція DMC_ReadPositionLagStatus визначає, чи перевищує фактична різниця положення вказаної осі значення параметра *Lag чи ні.*

Щоб отримати додаткові пояснення та приклад, зверніться до розділу інструкції DMC_ReadPositionLagStatus .

11.3.28 DMC_ChangeMechanismGearRatio

FB/FC	Пояснення	Застосовна модель
FB	DMC_ChangeMechanismGearRatio використовується для зміни значень параметрів осі.	AS516E-B AS532EST-B AS564EST-B

DMC_ChangeMechanismGearRatio _ екземпляр



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
InputRotation	Цей параметр і *OutputRotation* визначають механічне передавальне число.	LREAL	Позитивне ціле число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
OutputRotation	*InputRotation* і цей параметр визначають механічне передавальне число.	LREAL	Позитивне ціле число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Одиниці за обертання	Кількість одиниць, на яку переміщується виконавчий механізм терміналу, коли вихідний кінець коробки передач обертається по колу. (Одиниця: одиниці/оберт)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
AxisType	Тип осі 0: вісь обертання 1: лінійна вісь	USINT	0, 1 (0)	Коли *Execute* змінюється з FALSE на TRUE.
Модуло	Цикл, який використовується для рівного розподілу фактичного положення виконавчого механізму терміналу.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

- **Вихідні параметри**

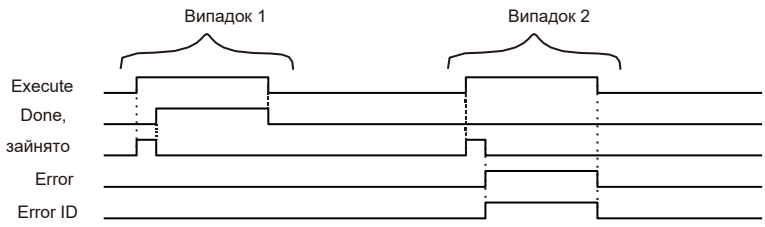
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

● функція

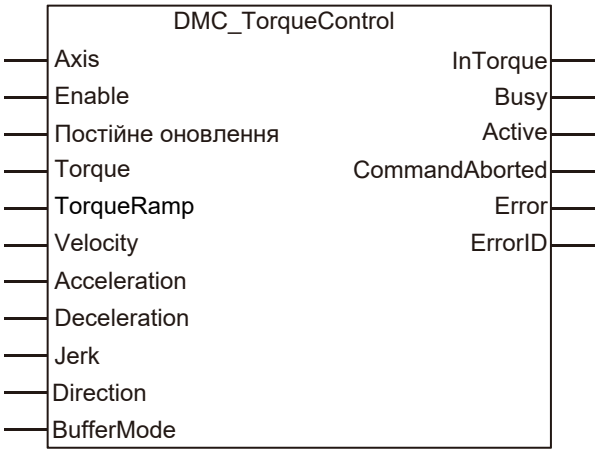
DMC_ChangeMechanismGearRatio використовується для зміни значень параметрів для термінального приводу. Інструкція може змінювати значення параметрів осі таким чином, щоб вони узгоджувалися з фактичними значеннями параметрів, що зручніше для користувачів.

Інструкція може бути виконана лише тоді, коли вісь знаходиться в стані Вимкнено або в стані зупинки.

11.3.29 DMC_TorqueControl

FB/FC	Пояснення	Застосовна модель
FB	DMC_TorqueControl керує віссю для роботи в режимі крутного моменту та виконання виведення крутного моменту.	AS516E-B AS532EST-B AS564EST-B

DMC_TorqueControl_прилад



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE .
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE .
ContinuousUpdate	Зарезервований	-	-	-
Крутний момент	Вкажіть цільовий крутний момент. Крутний момент виражається проміллом номінального крутного моменту осі сервоприводу. Наприклад, значення налаштування 30 вказує на те, що встановлений крутний момент становить 30‰ номінального крутного моменту осі сервоприводу. Якщо *Enable* змінює значення TRUE, зміна значення параметра призведе до негайної зміни крутного моменту.	INT	Від'ємне число, додатне число, 0 (0)	Коли *Enable* змінюється на TRUE .
TorqueRamp	Укажіть швидкість зміни крутного моменту від поточного до цільового. (Одиниця: ‰/с)	LREAL	Від'ємне число, додатне число (потрібно встановити значення змінної)	Коли *Enable* змінюється з FALSE на TRUE .
швидкість	Коли інструкція керування крутним моментом керує віссю, швидкість осі не може перевищувати встановлене значення. (Одиниця: одиниця/с)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Enable* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Прискорення	Зарезервований			
Уповільнення	Зарезервований			
ривок	Зарезервований			
Напрямок	Вкажіть напрямок обертання для осі. 1: Позитивний напрямок 3: Негативний напрямок	MC_Directi on	1: mcPositiveDirection, 3: mcNegativeDirection (Потрібно встановити значення змінної)	Коли *Enable* змінюється з FALSE на TRUE.
BufferMode	Зарезервований			

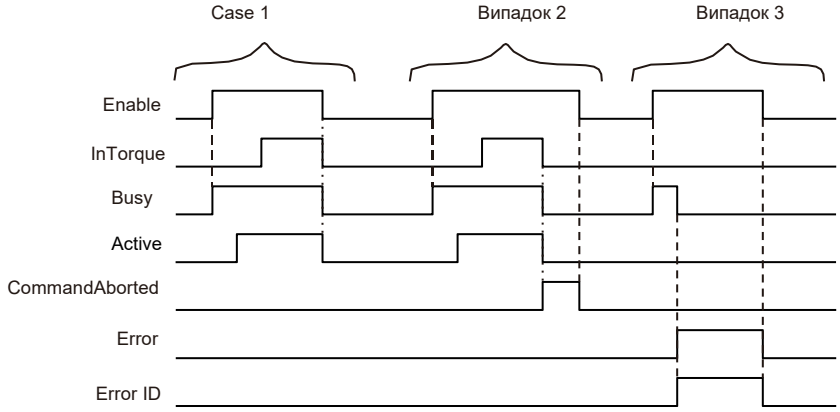
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
InTorque	TRUE, коли встановлений цільовий момент досягається.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного ідентифікатора помилки.	WORD	

● **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
InTorque	◆ Коли досягнуто цільового крутного моменту.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється з TRUE на FALSE.
Busy	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *InTorque* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *InTorque* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Enable* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлено на TRUE, коли інструкція переривається іншою інструкцією після того, як *Enable* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли інструкція починає керувати віссю, *Active* змінюється на TRUE. *InTorque* змінюється на TRUE, коли досягається встановлений цільовий крутний момент. Коли *Enable* змінюється з TRUE на FALSE, *Busy* , *Active* та *Intorque* змінюються на FALSE.

Випадок 2 : коли інструкція переривається MC_Stop після того, *як Enable* змінюється з FALSE на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *InTorque* , *Busy* і *Active* змінюються на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : вхідний параметр є недопустимим (наприклад, номер осі: 0) до виконання інструкції. Коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Error* змінюється на TRUE, *Busy* змінюється на FALSE, а *ErrorID* відображає відповідні коди помилок. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється з TRUE на FALSE, а значення *ErrorID* скидається на 0.

- **функція**

DMC_TorqueControl керує віссю для роботи в режимі крутного моменту та виконання виведення крутного моменту. На основі встановленої швидкості зміни крутного моменту рух осі змінюватиметься в процесі від поточного моменту до цільового крутного моменту. Якщо значення крутного моменту змінено під час виконання інструкції, крутний момент сервоприводу негайно зміниться відповідно до швидкості зміни крутного моменту, а потім сервопривод збереже значення крутного моменту для руху. Під час виконання інструкції інструкція контролюватиме швидкість осі, щоб вона не перевищувала встановлену максимальну швидкість.

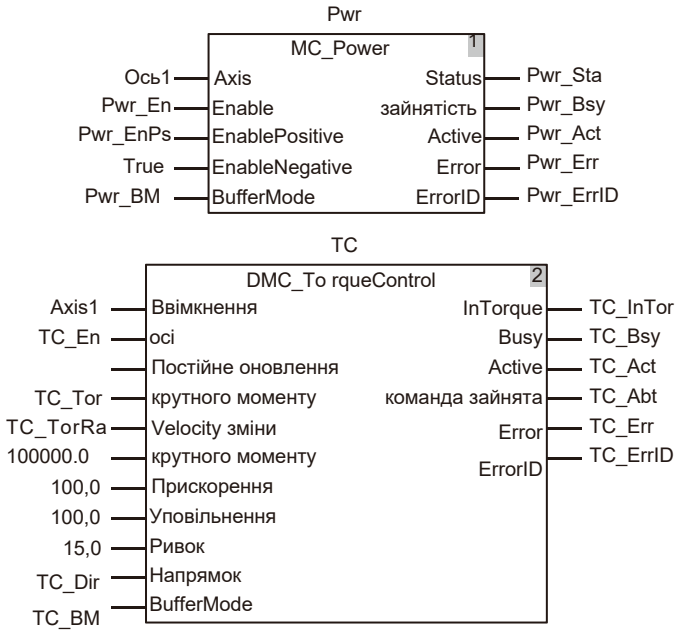
Щоб зупинити інструкцію, змініть *Enable* на FALSE або скористайтеся MC_Stop, щоб скасувати інструкцію. Коли *Enable* змінюється з TRUE на FALSE, вісь вийде з режиму крутного моменту, і крутний момент зміниться негайно (швидкість зміни крутного моменту буде недійсною). Якщо вісь має зупинятися поступово відповідно до швидкості зміни крутного моменту, просто встановіть цільовий крутний момент на 0, зачекайте, доки фактичний вихідний крутний момент осі зміниться на 0, а потім змініть *Enable* на FALSE.

Приклад програмування

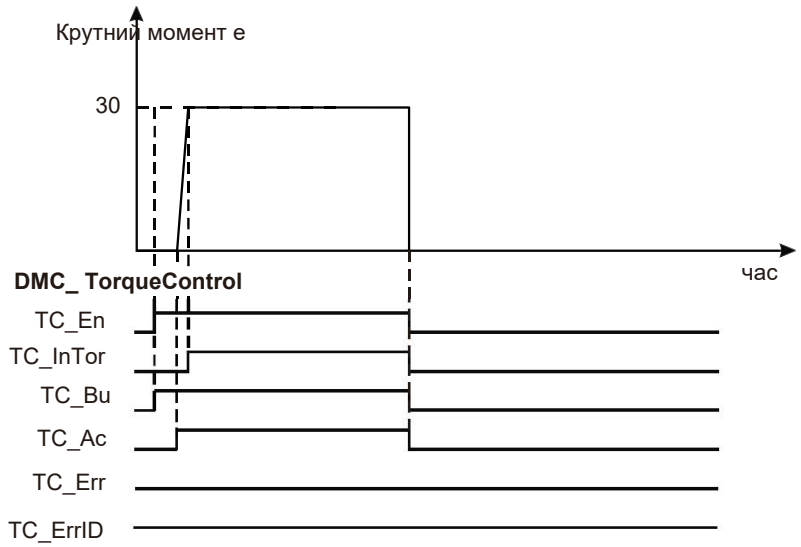
2. **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_EnPs	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
TC	DMC_TorqueControl	
TC_En	BOOL	FALSE
TC_Tor	INT	30
TC_TorRa	LREAL	5.0
TC_Vel	LREAL	100000,0
TC_Dir	MC_Direction	1
TC_InTor	BOOL	
TC_Bsy	BOOL	
TC_Act	BOOL	
TC_Abt	BOOL	
TC_Err	BOOL	
TC_ErrID	WORD	



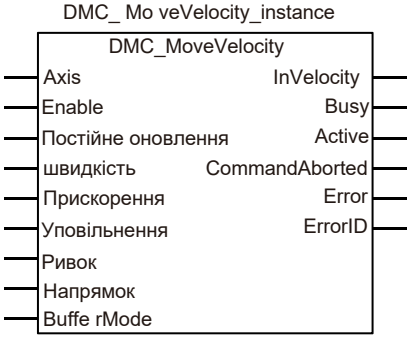
3. **Крива руху та часовий графік**



- Коли TC_En змінюється з FALSE на TRUE, виконується DMC_TorqueControl. Тим часом TC_Bsy змінюється з FALSE на TRUE. Коли TC_Act змінюється на TRUE, інструкція починає керувати віссю, і значення крутного моменту буде збільшено відповідно до встановленої швидкості зміни крутного моменту.
- Після встановлення значення TC_Tor на 0 значення крутного моменту зменшується до 0 відповідно до встановленої швидкості зміни крутного моменту.
- Коли TC_En змінюється з TRUE на FALSE, TC_InTor, TC_Bsy і TC_Act змінюються з FALSE на TRUE, і сервопривід виходить із режиму крутного моменту.

11.3.30 DMC_MoveVelocity

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveVelocity змінює значення параметрів інструкції швидкості, щоб зробити керовану вісь дійсною негайно під час виконання інструкції швидкості.	AS516E-B AS532EST-B AS564EST-B

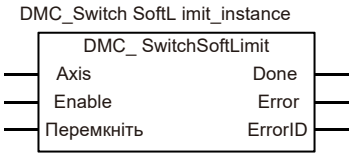


- **функція**

DMC_MoveVelocity змінює швидкість і робить швидкість осі дійсною негайно під час виконання інструкції швидкості. Функція інструкції подібна до інструкції MC_MoveVelocity. Єдина відмінність між двома інструкціями полягає в тому, що зміна значень *Velocity* , *Acceleration* , *Deceleration* , *Jerk* і *Direction* інструкції набуде чинності негайно, оскільки *Enable* має значення TRUE.
Докладніше про параметри та типи даних див. у MC_MoveVelocity.

11.3.31 DMC_SwitchSoftLimit

FB/FC	Пояснення	Застосовна модель
FB	DMC_SwitchSoftLimit керує програмним кінцевим вимикачем.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE .
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE .
Перемикач	НЕПРАВИЛЬНО: Вимкніть програмний кінцевий вимикач. TRUE: увімкніть програмний кінцевий вимикач.	BOOL-	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* змінюється на TRUE .

● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

● **Час оновлення**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *Enable* змінюється з TRUE на FALSE.
Error	◆ Вхідні параметри для інструкції недопустимі.	◆ Коли *Enable* змінюється з TRUE на FALSE.

● **функція**

DMC_SwitchSoftLimit керує програмним кінцевим вимикачем. Коли *Switch* має значення TRUE і виконання інструкції завершено, програмний кінцевий вимикач увімкнеться. Коли *перемикач* має значення FALSE і виконання інструкції завершено, програмний кінцевий вимикач вимикається.

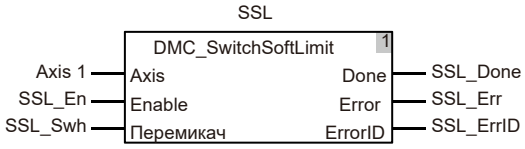
Крім того, користувачі можуть налаштувати перемикач, клацнувши «Конфігурація мережі»-> «Рух»->Конфігурація осі->Загальні в програмному забезпеченні без використання інструкцій.
Максимальне положення та мінімальне положення потрібно встановити за допомогою програмного забезпечення.

Після того, як програмний кінцевий перемикач увімкнено, виникне помилка, якщо позиція осі, яка контролюється інструкціями руху, перевищує діапазон програмного обмеження.

Приклад програмування

■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
SSL	DMC_SwitchSoftLimit	
Axis1	USINT	1
SSL_En	BOOL	FALSE
SSL_Swh	BOOL	FALSE
SSL_Done	BOOL	
SSL_Err	BOOL	
SSL_ErrID	WORD	

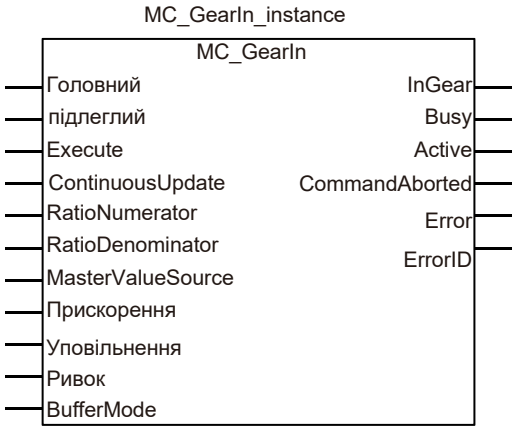


- Коли SSL_Swh має значення TRUE, а SSL_En змінюється з FALSE на TRUE, інструкція виконується. Коли виконання інструкції завершено, SSL_Done змінюється на TRUE і програмний кінцевий перемикач номер 1 увімкнено.
- Коли SSL_Swh має значення FALSE і SSL_En змінюється з FALSE на TRUE, інструкція виконується. Після завершення виконання інструкції SSL_Done змінюється на TRUE, а програмний кінцевий перемикач номер 1 вимикається.

11.4 Інструкції по зчепленню

11.4.1 MC_GearIn

FB/FC	Пояснення	Застосовна модель
FB	MC_GearIn використовується для встановлення електронної передачі між двома осями.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
майстер	Вкажіть номер головної осі, якою має керувати інструкція	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Підлеглий	Вкажіть номер веденої осі, якою керує інструкція	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований	-	-	-
RatioNumerator	Чисельник передавального числа	LREAL	Додатне число і від'ємне число (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ВідношенняЗнаменник	Передавальне число Знаменник	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
MasterValueSource	Вибір джерела команд 0 : позиція команди головної осі, за якою слідує ведена вісь 1 : фактичне положення головної осі, за якою слідує ведена вісь	MC_Source	0:mcSetValue 1:mcActualValue (0)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Прискорення	Вкажіть цільове прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Вкажіть цільове уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни прискорення та уповільнення цілі. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
BufferMode	Укажіть поведінку під час виконання двох інструкцій. 0: переривання 1: Буферизовано	MC_Buffer _ Mode	1 : mcAborting 2 : mcBuffered (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Виконання MC_GearIn починається, коли *Execute* змінюється з FALSE на TRUE. Незалежно від того, завершено виконання інструкції чи ні, інструкція може бути виконана повторно, коли *Execute* знову змінюється з FALSE на TRUE. Під час повторного виконання лише параметри *RatioNumerator* , *RatioDenominator* , *MasterValueSource* , *Acceleration* , *Deceleration* і *Jerk* знову будуть ефективними.
2. Підпорядкована вісь, визначена інструкцією MC_GearIn, може виконувати іншу інструкцію руху під час виконання MC_GearIn. У той час як інша інструкція руху переривае інструкцію MC_GearIn, взаємозв'язок передач між головною віссю та веденою віссю буде роз'єднано. MC_Halt або MC_Stop можуть перервати рух веденої осі.
3. Зверніться до розділу 10.2 щодо співвідношення між *прискоренням* , *уповільненням* і *ривком* .
4. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

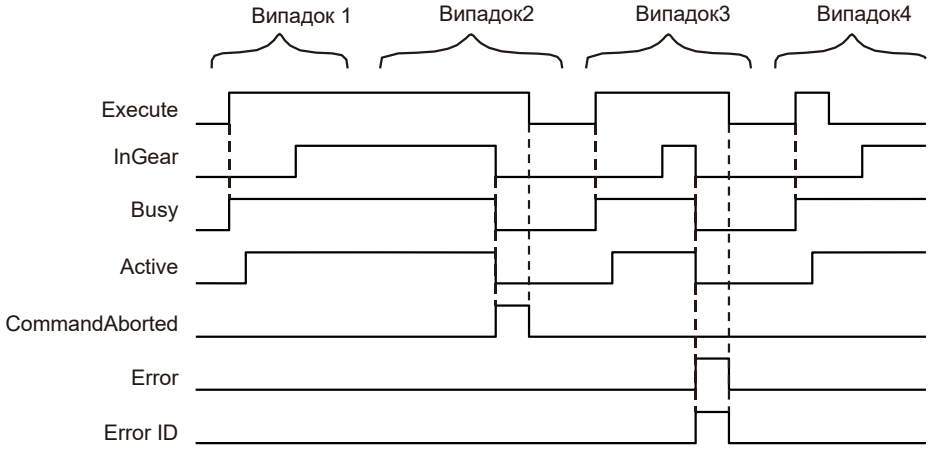
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
InGear	TRUE, коли ведена вісь досягає синхронного стану.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
InGear	◆ Коли ведена вісь переходить в синхронний стан.	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ *InGear* одразу зміниться на FALSE, коли вхідний параметр буде змінено після досягнення синхронного стану, а *Execute* знову зміниться з FALSE на TRUE. ◆ *InGear* одразу зміниться на FALSE, якщо вхідний параметр не буде змінено після завершення виконання інструкції, а *Execute* знову зміниться з FALSE на TRUE. А в наступному періоді *InGear* змінюється на TRUE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Active	◆ Коли вісь починає керуватись інструкцією	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією руху	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через один період *активність* змінюється на TRUE. Коли досягається синхронний стан, *InGear* змінюється на TRUE, а параметри *Busy* та *Active* залишаються TRUE.

Випадок 2 : коли *Execute* змінюється на TRUE, а підлегла вісь керується іншою інструкцією, інструкція MC_GearIn переривається іншою інструкцією, а *CommandAborted* змінюється на TRUE. Тим часом *«Зайнятий»* і *«Активний»* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли *Execute* змінюється з FALSE на TRUE і виникає така помилка, як помилка параметра, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок. Тим часом *InGear* , *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 4 : після того, як *Execute* зміниться з TRUE на FALSE у процесі виконання MC_GearIn, *InGear* змінюється на TRUE, а параметри *Busy* та *Active* залишаються TRUE.

● функція

1. MC_GearIn використовується для встановлення електронної передачі між двома осями. Після виконання інструкції MC_GearIn підпорядкована вісь виконує операцію зубчастого колеса з головною віссю відповідно до параметрів *RatioNumerator* , *RatioDenominator* , *Acceleration* , *Deceleration* , *Jerk* і *BufferMode* . Головна вісь може бути реальною віссю, віртуальною віссю або віссю кодера. Вісь Salve може бути реальною віссю або віртуальною віссю.
2. Під час виконання інструкції ведену вісь потрібно ввімкнути, а головну вісь можна увімкнути або вимкнути.
3. Якщо інструкція MC_GearIn виконується, коли зв’язок електронної передачі між двома осями ще не побудований, швидкість веденої осі досягне цільової швидкості відповідно до значень *RatioNumerator* , *RatioDenomenator* , *Acceleration* , *Deceleration* і *Jerk* , визначених інструкцією. .

$$\text{Прискорення (або уповільнення) підпорядкованої осі} = \text{прискорення (або уповільнення) головної осі} \times \frac{\text{RatioNumerator}}{\text{Знаменник співвідношення}}$$

Після встановлення зв’язку електронної передачі між двома осями (коли *InGear* інструкції змінюється на TRUE), зв’язок між швидкістю веденої осі, чисельником передавального числа, знаменником передавального числа та швидкістю головної осі відображається як нижче.

$$\text{Цільова швидкість підпорядкованої осі} = \text{швидкість головної осі} \times \frac{\text{Передавальне число n umeberator}}{\text{Передавальне число знаменник}}$$

4. E-передавальне число

$$\text{E - передавальне число} = \frac{\text{Чисельник співвідношення}}{\text{Відношення знаменника}}$$

Якщо передаточне число електронної передачі є додатним числом, напрямки руху веденої осі та головної осі однакові.

Якщо передаточне число електронної передачі є від’ємним числом, напрямки руху веденої осі та головної осі протилежні.

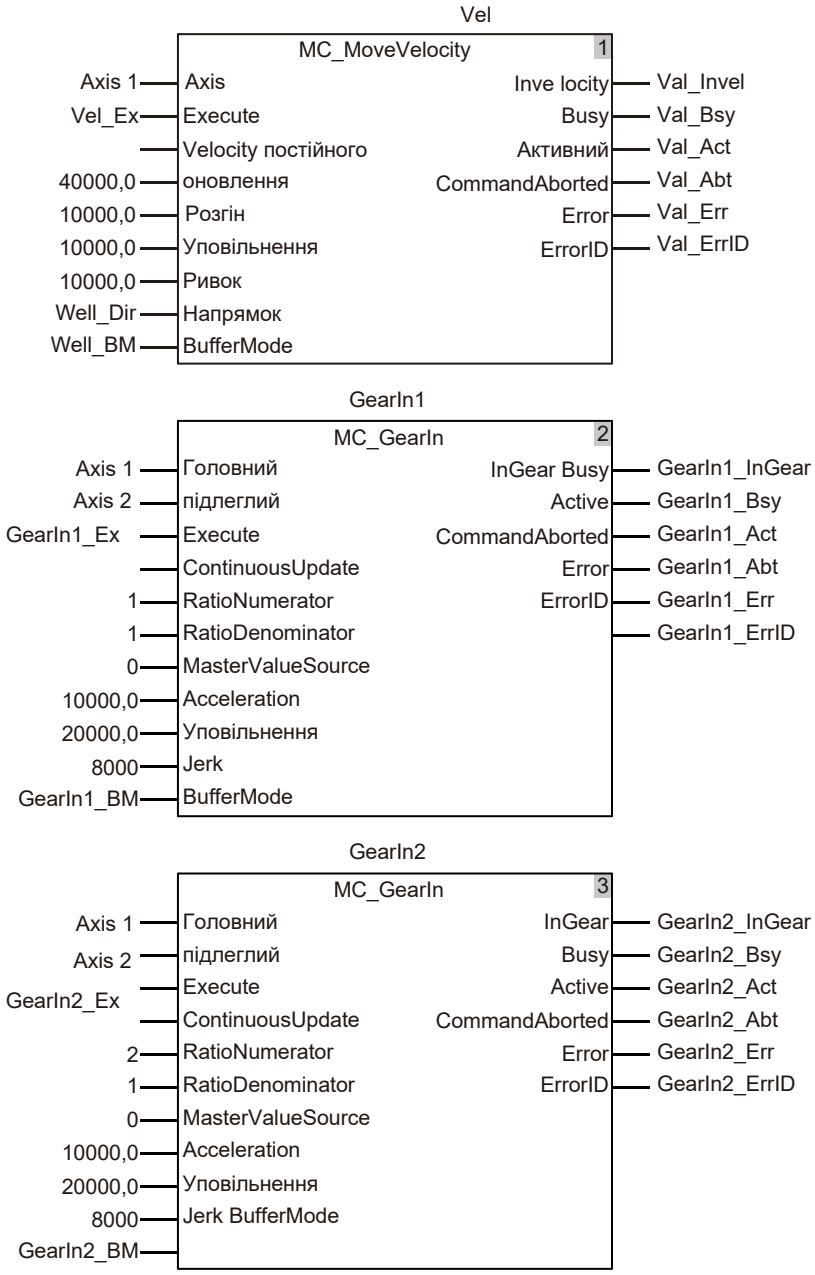
Приклад програмування

Нижче наведено приклад виконання інструкції MC_GearIn.

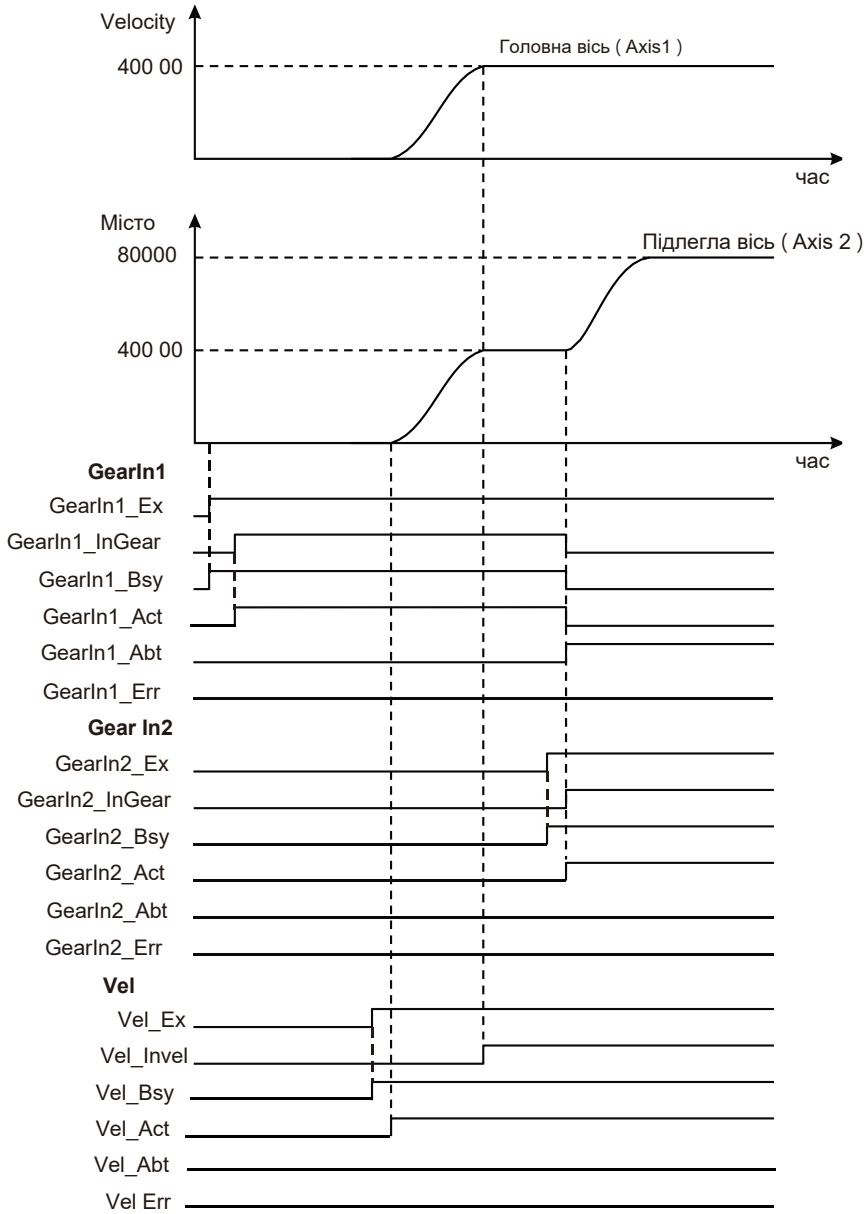
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Vel	MC_MoveVelocity	
Axis1	USINT	1
Axis2	USINT	2
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Well_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
GearIn1	MC_GearIn	
GearIn1_Ex	BOOL	FALSE
GearIn1_BM	MC_Buffer_Mode	0
GearIn1_InGear	BOOL	
GearIn1_Bsy	BOOL	
GearIn1_Act	BOOL	
GearIn1_Abt	BOOL	
GearIn1_Err	BOOL	
GearIn1_ErrID	WORD	
GearIn2	MC_GearIn	
GearIn2_Ex	BOOL	FALSE
GearIn2_BM	MC_Buffer_Mode	0
GearIn2_InGear	BOOL	
GearIn2_Bsy	BOOL	
GearIn2_Act	BOOL	
GearIn2_Abt	BOOL	
GearIn2_Err	BOOL	
GearIn2_ErrID	WORD	



3. Крива руху та часові діаграми:

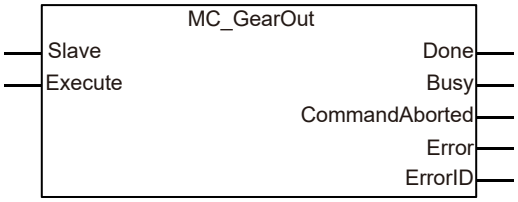


- У GearIn1 значення *RatioNumerator* і *RatioDenomenator* дорівнюють 1. GearIn1_Ex змінюється з FALSE на TRUE, а тим часом GearIn1 _Bsy змінюється на TRUE. Через один період GearIn1_InGear змінюється на TRUE, і між головною віссю та веденою віссю встановлюється зв'язок електронної передачі.
- Vel_Ex змінюється з FALSE на TRUE після встановлення зв'язку електронної передачі між головною віссю та веденою віссю. Через один період Vel_Act змінюється на TRUE, головна вісь виконує команду швидкості, а підпорядкована вісь слідує за головною віссю для руху.
- У GearIn2 значення *RatioNumerator* і *RatioDenomenator* дорівнюють 2 і 1 відповідно. GearIn2_Ex змінюється з FALSE на TRUE, а GearIn2 _Bsy тим часом змінюється на TRUE. Через один період дії GearIn2_Act і GearIn1_Abt змінюються на TRUE, і ведена вісь досягає цільової швидкості на основі значень RatioNumberator, Ratio Denomenator, MasterValueSource, Acceleration, Jerk і BufferMode, визначених інструкцією GearIn2. Оскільки значення *RatioNumerator* і *RatioDenomenator* у GearIn2 дорівнюють 2 і 1 відповідно, цільова швидкість веденої осі вдвічі більша, ніж головна. Коли GearIn2_InGear змінюється на TRUE, швидкість підлеглої осі буде вдвічі більшою, ніж головна вісь.

11.4.2 MC_GearOut

FB/FC	Пояснення	Застосовна модель
FB	MC_GearOut використовується для завершення встановленого зв'язку електронної передачі між головною віссю та веденою віссю.	AS516E-B AS532EST-B AS564EST-B

MC_GearOut_примірник



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Підлеглий	Вкажіть номер підлеглої осі, яку потрібно від'єднати від зв'язку електронної передачі.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

Примітки:

1. Підпорядкована вісь продовжуватиме рухатися зі швидкістю від'єднання, якщо вісь підлеглого від'єднається від зв'язку електронної передачі за допомогою інструкції MC_GearOut після того, як дві осі створили зв'язок електронної передачі за допомогою інструкції MC_GearIn.
2. Підпорядкована вісь може виконувати інші інструкції руху після завершення виконання інструкції MC_GearOut.
3. Зв'язок між головною віссю та веденою віссю розривається за допомогою інструкції MC_GearOut. Щоб зупинити рух веденої осі, можна використати інструкцію MC_Halt або MC_Stop.

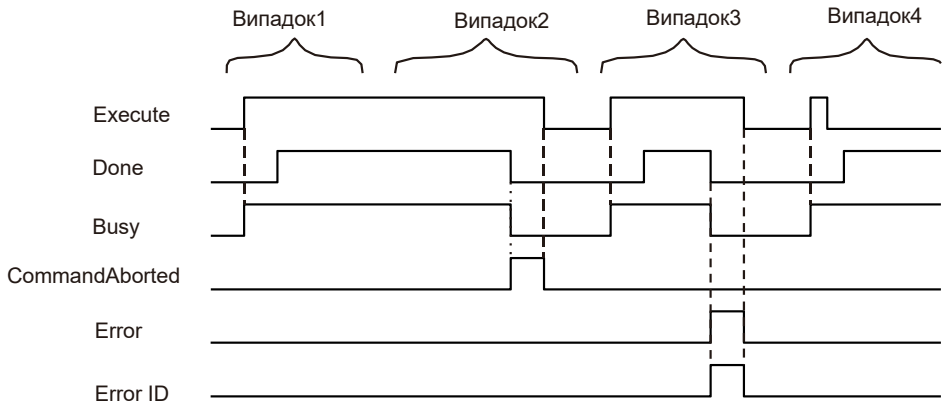
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли взаємозв'язок електронної передачі між підпорядкованою віссю та головною віссю роз'єднано, а інструкція MC_GearOut керує осями.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли зв'язок електронної передачі між підпорядкованою віссю та головною віссю роз'єднаний	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією руху	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через крапку *«Готово»* змінюється на «TRUE». *Busy* і *Done* залишаються TRUE після того, як *Execute* зміниться з TRUE на FALSE.

Випадок 2 : якщо інструкція MC_GearOut переривається іншою інструкцією, оскільки *Execute* змінюється на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Done* змінюються на FALSE. *CommandAborted* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : коли виникає помилка (наприклад, вісь вимкнено), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок після того, як *Execute* змінюється з FALSE на TRUE. Тим часом *Busy* і *Done* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 4 : *Виконайте* зміни з TRUE на FALSE до досягнення періоду під час виконання інструкції MC_GearOut. *Done* змінюється на TRUE, а *Busy* залишається TRUE, коли досягається період.

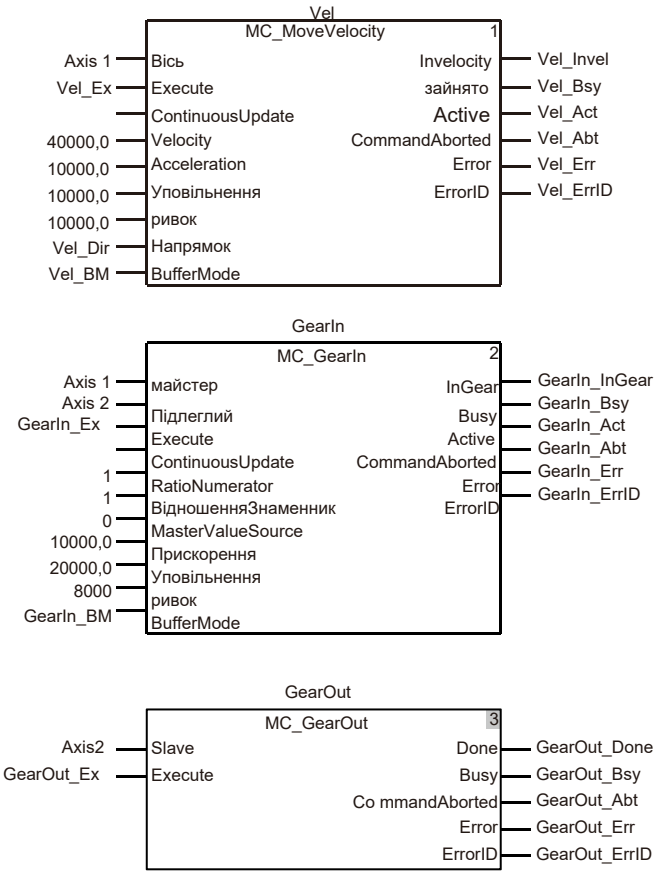
Приклад програмування

Нижче наведено приклад виконання інструкції MC_GearOut.

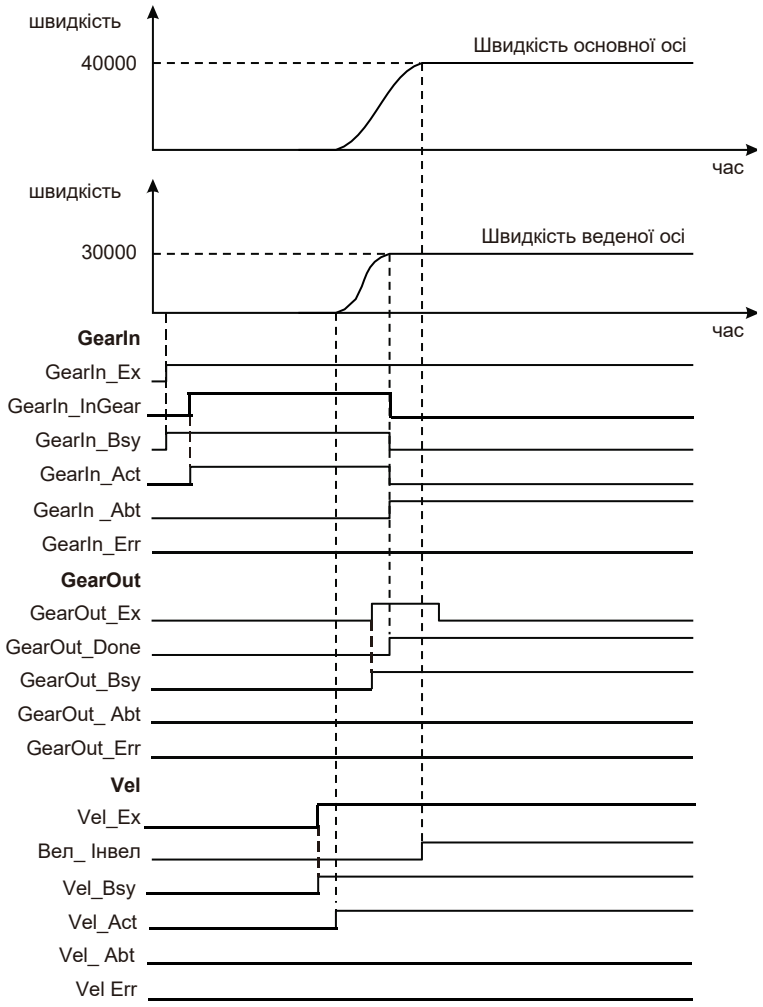
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
Vel	MC_MoveVelocity	
Axis1	USINT	1

Ім'я змінної	Тип даних	Початкове значення
Axis2	USINT	2
Vel_Ex	BOOL	FALSE
Vel_Dir	MC_DIRECTION	1
Vel_BM	MC_Buffer_Mode	0
Vel_Invel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
GearIn	MC_ GearIn	
GearIn_Ex	BOOL	FALSE
GearIn_BM	MC_Buffer_Mode	0
GearIn_InGear	BOOL	
GearIn_Bsy	BOOL	
GearIn_Act	BOOL	
GearIn_Abt	BOOL	
GearIn_Err	BOOL	
GearIn_ErrID	WORD	
GearOut	MC_ GearOut	
GearOut_Ex	BOOL	FALSE
GearOut_Done	BOOL	
GearOut_Bsy	BOOL	
GearOut_Act	BOOL	
GearOut_Abt	BOOL	
GearOut_Err	BOOL	
GearOut_ErrID	WORD	



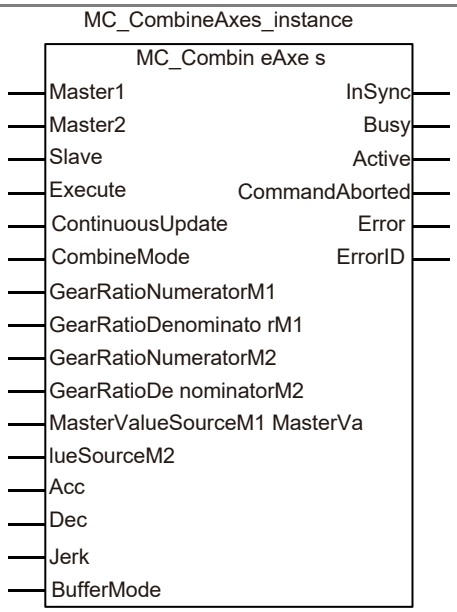
2. Крива та часові діаграми:



- Коли GearIn_Ex змінюється з FALSE на TRUE, GearIn_Bsy змінюється на TRUE. Через один період GearIn_InGear змінюється на TRUE, і між головною та підпорядкованою віссю встановлюється взаємозв'язок передачі.
- Після того, як встановлено передавальне співвідношення між двома осями, Vel_Ex змінюється з FALSE на TRUE. Через один період Vel_Act змінюється на TRUE. Головна вісь виконує команду швидкості, а ведена вісь рухається, слідуючи руху головної осі.
- Поки головна вісь виконує інструкцію швидкості, GearOut_Ex змінюється з FALSE на TRUE, а GearOut_Bsy змінюється на TRUE. Через один період GearOut_Done і GearIn_Abt змінюються на TRUE. І ведена вісь продовжить рухатися з поточною швидкістю.

11.4.3 MC_CombineAxes

FB/FC	Пояснення	Застосовна модель
FB	MC_CombineAxes виводить суму або різницю змін позиції двох головних осей як зміну позиції підпорядкованої.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Майстер1	Джерело позиції осі 1	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Майстер2	Джерело положення осі 2	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Підлеглий	Керований підлеглий	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	--
ContinuousUpdate	Зарезервований	-	-	-
CombineMode	Вибір комбінованого методу. 0: Сума двох варіацій положення головної осі	MC_Combi neMode	0: mcAddAxes 1: mcSubAxes (0)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
	1: Різниця двох варіацій положення головної осі			
GearRatioNumerator M1	Вкажіть чисельник передавального числа головної осі1.	LREAL	Позитивне чи негативне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
GearRatioDenominator M1	Укажіть знаменник передавального числа головної осі1.	LREAL	Позитивне чи негативне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
GearRatioNumerator M2	Укажіть чисельник передавального числа головної осі2.	LREAL	Позитивне чи негативне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
GearRatioDenominator M2	Укажіть знаменник передавального числа головної осі2.	LREAL	Позитивне чи негативне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
MasterValueSource M1	Вкажіть джерело позиції синхронізації головної осі 1. 0 : командна позиція 1 : Фактична позиція	MC_SOURCE	0:mcSetValue 1:mcActualValue (0)	Коли *Execute* змінюється з FALSE на TRUE
MasterValueSource M2	Вкажіть джерело позиції синхронізації головної осі 2. 0 : командна позиція 1 : Фактична позиція	MC_SOURCE	0:mcSetValue 1:mcActualValue (0)	Коли *Execute* змінюється з FALSE на TRUE
Акк	Вкажіть прискорення для веденої осі.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
груд	Вкажіть уповільнення для веденої осі.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Вкажіть швидкість зміни прискорення для веденої осі.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
BufferMode **Примітки:**	Вкажіть поведінку під час виконання двох інструкцій. 0 : перервано 1 : Буферизовано	MC_Buffer_Mode	1 : mcAborting 2 : mcBuffered (0)	Коли *Execute* змінюється з FALSE на TRUE

1. Виконання інструкції починається, коли *Execute* змінюється з FALSE на TRUE. Коли *Execute* знову змінюється з FALSE на TRUE, незалежно від того, завершено виконання інструкції чи ні, інструкція не може бути виконана повторно, а попередні значення налаштувань буде збережено.

2. Зверніться до розділу 10.2 щодо співвідношення між *положенням* , *швидкістю* , *прискоренням* і *ривком* .
3. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

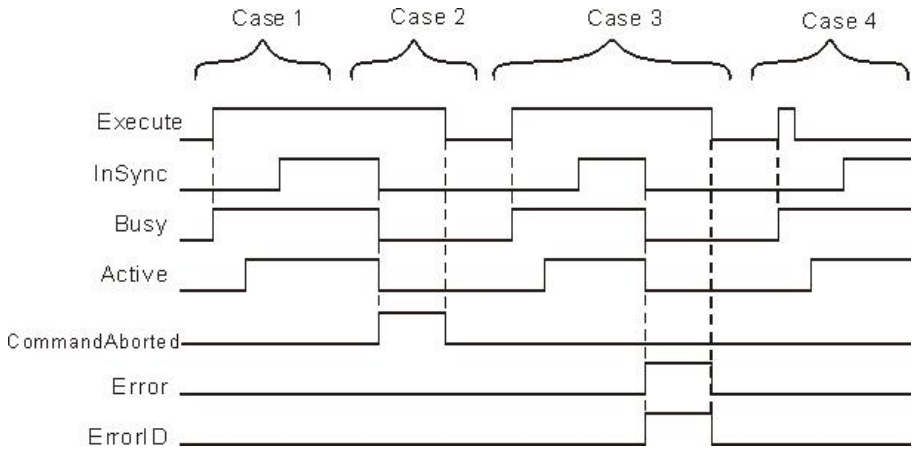
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
InSync	TRUE, коли ведена вісь завершила дію синхронізації.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
InSync	◆ Коли ведена вісь завершує дію синхронізації.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *CommandAborted* змінюється на TRUE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ Коли виконання цієї інструкції переривається іншою інструкцією керування рухом.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлено на TRUE, коли інструкцію переривається після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один цикл *Active* змінюється на TRUE. Коли підлегла вісь синхронізується з двома головними осями, *InSync* змінюється на TRUE, а *Busy* та *Active* залишаються TRUE.

Випадок 2 : коли для *параметра «Execute»* значення «TRUE», *«Зайнятий»* — «TRUE», а *«Активний»* — «TRUE». Коли підлеглий пристрій синхронізовано з двома головними осями, *InSync* має значення TRUE. На даний момент інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Invelocity* , *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, коли виникає помилка, наприклад, сигнали тривоги осі або офлайн. Тим часом *InSync* , *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 4 : інструкція все ще виконується, а стани *«Зайнятий»* і *«Активний»* не змінюються після того, як параметр *«Execute»* змінюється з «TRUE» на «ХИБНЕ» під час виконання інструкції. Коли підлеглу вісь синхронізовано з двома головними осями, *InSync* змінюється на TRUE, а *Busy* та *Active* залишаються TRUE.

- **функція**

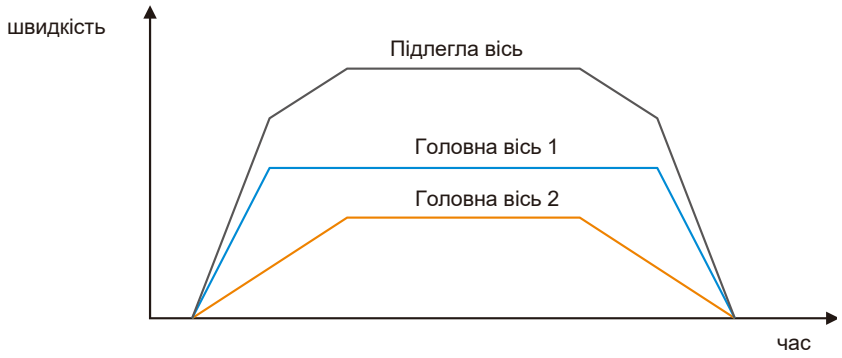
MC_CombineAxes виводить суму або різницю змін позиції двох головних осей як зміну позиції підпорядкованої.

■ **Режими комбінування: додавання або віднімання**

Проводиться додавання або віднімання змін позиції головної осі 1 і головної осі 2, а результат розрахунку виводиться як зміна позиції підлеглої осі.

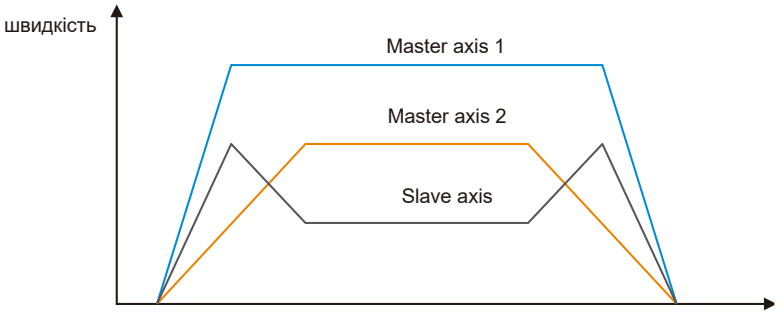
■ ***CombineMode*** **встановлено на 0**

$$\text{Зміна положення підлеглої осі} = \text{Зміна положення головної осі1} \times \frac{\text{GearRationNumeratorM1}}{\text{GearRatioDenominatorM1}} + \text{Зміна положення головної осі2} \times \frac{\text{GearRatioNumeratorM2}}{\text{Знаменник передавального відношенняM2}}$$



■ ***CombineMode*** **встановлено на 1**

$$\text{Зміна положення підлеглої осі} =$$
$$= \text{Зміна положення головної осі1} \times \frac{\text{GearRationNumeratorM1}}{\text{Передаточне відношення ЗнаменникM1}} -$$
$$\text{Варіація положення головної осі2} \times \frac{\text{GearRatioNumeratorM2}}{\text{Знаменник передавального відношенняM2}}$$



- Чисельник і знаменник головного передавального числа є факторами для коригування змін положення двох головних осей. Дивіться формулу вище.
- *Для параметра MasterValueSource* можна встановити значення 0 (позиція команди) і 1 (фактична позиція), щоб указати джерело зміни позиції. Якщо значення встановлено на 0, додайте варіації положення команди головної осі. Якщо значення встановлено на 1, відніміть одну варіацію фактичного положення головної осі від варіації фактичного положення іншої головної осі.
- Acc , *Dec* і *Jerk вказують на те ,* що головна вісь перебувала в русі до виконання інструкції. Якщо інструкція виконується в даний момент, ведена вісь буде прискорюватися або знижуватися відповідно до встановленого прискорення, уповільнення та ривка, щоб реалізувати синхронізацію з варіаціями головного положення. Коли синхронізація досягнута, *InSync* має значення TRUE і виконання інструкції завершено.
- Використовуйте іншу інструкцію руху (наприклад, інструкцію MC_Stop) для контролю над підпорядкованою віссю, щоб завершити зв'язок головної та підлеглої осі в інструкції. Встановіть значення *BufferMode* для іншої інструкції руху, яка має параметр *Buffermode,* рівним 0, щоб перервати інструкцію MC_CombineAxes і від'єднати зв'язок головної та підпорядкованої осі.
- Якщо передавальне число головної осі має бути переключено під час руху, використовуйте іншу інструкцію MC_CombineAxes, щоб перервати інструкцію MC_CombineAxes, яка виконується.

Приклад програмування

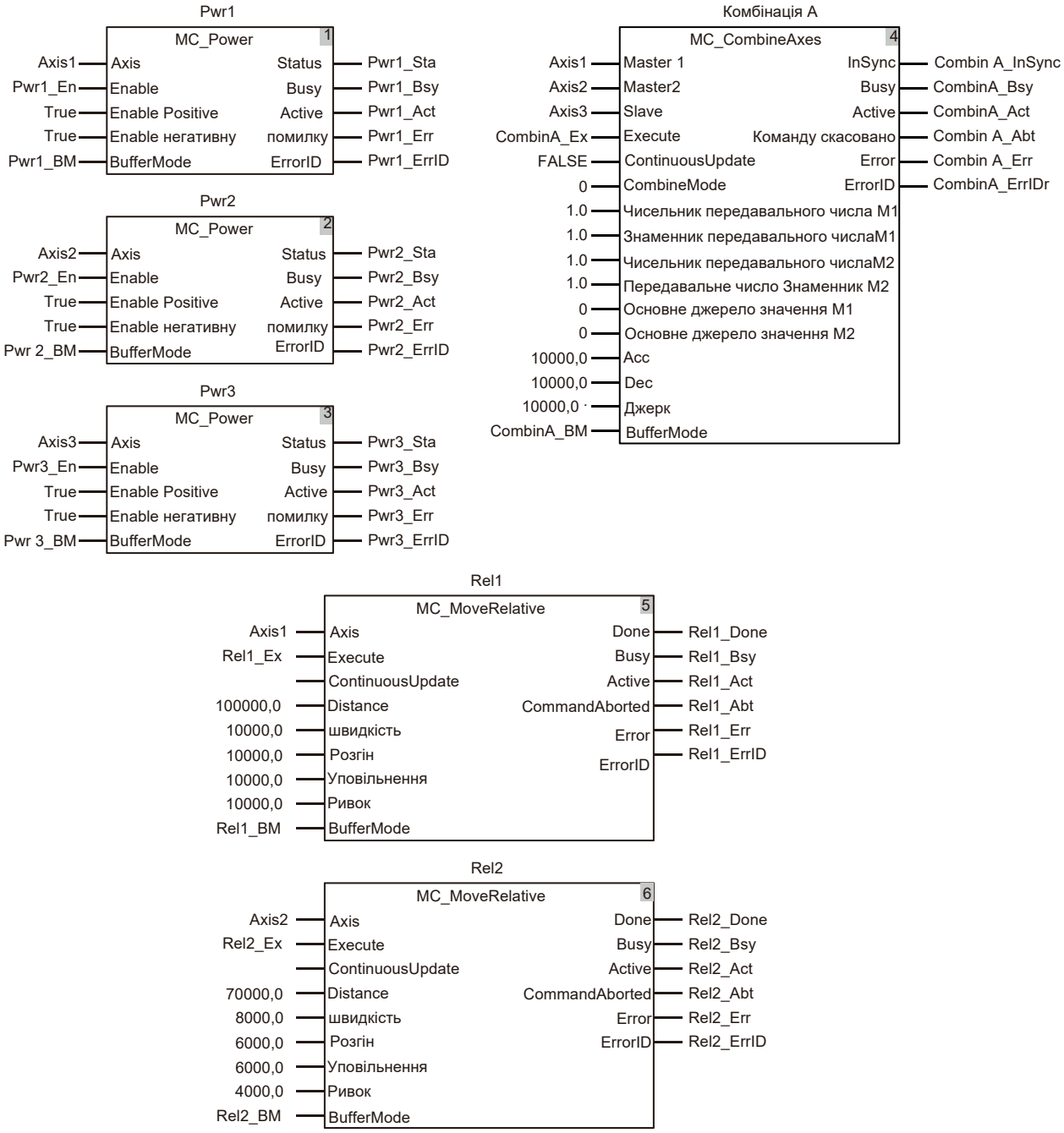
Приклад виконання інструкції MC_CombineAxes описано нижче.

1. Таблиця змінних і програма

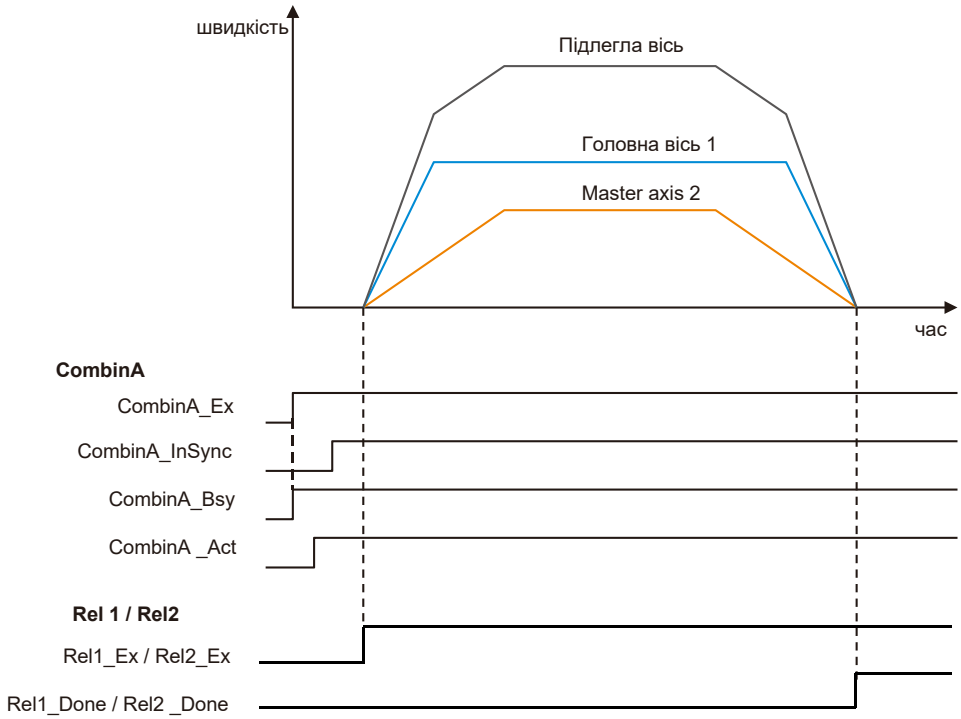
Ім'я змінної	Тип даних	Початкове значення
Pwr1	MC_Power	
Axis1	USINT	1
Pwr1_BM	MC_Buffer_Mode	1
Pwr1_Sta	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
Pwr2	MC_Power	
Axis2	USINT	1
Pwr2_BM	MC_Buffer_Mode	1
Pwr2_Sta	BOOL	
Pwr2_Bsy	BOOL	
Pwr2_Act	BOOL	
Pwr2_Err	BOOL	
Pwr2_ErrID	WORD	
Pwr3	MC_Power	
Axis3	USINT	1
Pwr3_BM	MC_Buffer_Mode	1
Pwr3_Sta	BOOL	
Pwr3_Bsy	BOOL	
Pwr3_Act	BOOL	
Pwr3_Err	BOOL	
Pwr3_ErrID	WORD	
CombinA	MC_CombineAxes	
CombinA_Ex	BOOL	FALSE
CombinA_BM	MC_Buffer_Mode	1
CombinA_InSync	BOOL	
CombinA_Bsy	BOOL	
CombinA_Act	BOOL	
CombinA_Abt	BOOL	
CombinA_Err	BOOL	
CombinA_ErrID	WORD	
Rel1	MC_MoveRelative	
Rel1_Ex	BOOL	FALSE
Rel1_Dir	MC_DIRECTION	1
Rel1_BM	MC_Buffer_Mode	0
Rel1_Done	BOOL	
Rel1_Bsy	BOOL	
Rel1_Act	BOOL	
Rel1_Abt	BOOL	
Rel1_Err	BOOL	
Rel1_ErrID	WORD	
Rel2	MC_MoveRelative	
Rel2_Ex	BOOL	FALSE
Rel2_Dir	MC_DIRECTION	1
Rel2_BM	MC_Buffer_Mode	0
Rel2_Done	BOOL	
Rel2_Bsy	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Rel2_Act	BOOL	
Rel2_Abt	BOOL	
Rel2_Err	BOOL	
Rel2_ErrID	WORD	



2. Крива руху та часовий графік



❖ Коли CombinA_Ex змінюється з FALSE на TRUE, починається виконання інструкції MC_CombineAxes. Через певний проміжок часу інструкція виконується успішно, CombinA_InSync змінюється на TRUE, і три осі можуть перейти в синхронізований рух за потреби. На даний момент *Execute* s MC_MoveRelatives для двох головних осей встановлено на TRUE, а потім дві головні осі починають працювати, а тим часом підпорядкована також починає працювати відповідно до суми варіацій положення двох головних осей. Зміна положення веденої осі є сумою змін положення двох головної осі в одиницю часу.

Після завершення інструкцій, виконаних для головних осей, три осі залишаються в синхронізованому стані. Щоб перервати стан синхронізації трьох осей, скористайтеся інструкцією MC_Stop, щоб перервати рух підлеглої осі та відключити стан синхронізації.

11.4.4 Представлення електронної камери

Кулачок - це компонент із вигнутим профілем або канавками. Він передає рух ведучому механізму біля свого краю, і стійка періодично обертається слідом за провідним механізмом. Кулачковий механізм складається з кулачка, толкателя і зубчастої рейки. На наступному малюнку показано профіль кулачка, що складається з точок A, B, C і D. AB' — це опорна опора, яка з'єднана зі стійкою. δ4 – внутрішній кут природного укосу; δ2 – зовнішній кут природного укосу. Радіус базового кола дорівнює r0, а S — кулачкова крива.

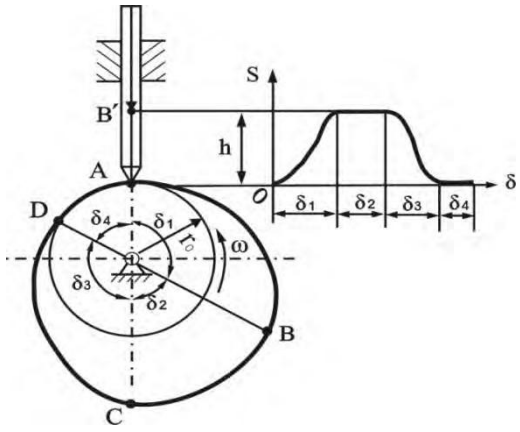


Рисунок 11.4.4.1

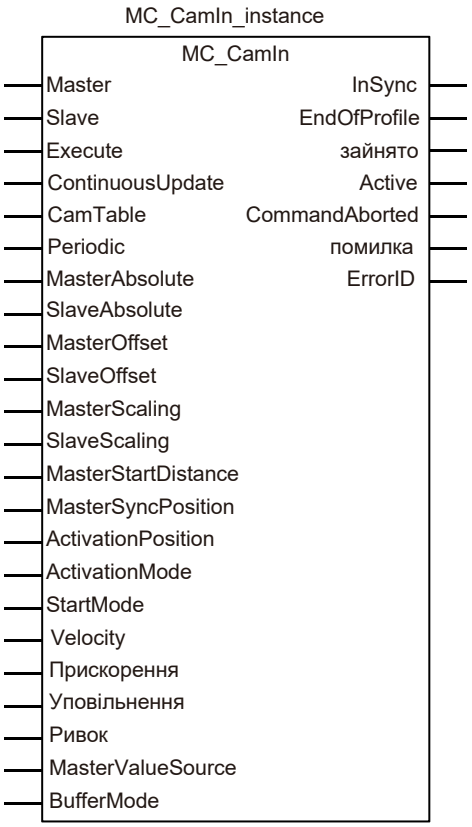
Електронний кулачок є аналогом механічного кулачка за допомогою комп'ютерної технології. Порівняно з механічним кулачком, електронний кулачок має багато переваг, оскільки його легко конструювати та модифікувати; економія витрат; більш висока ефективність і точність. Оскільки електронний кулачок є аналоговим кулачком, для електронного кулачка можна уникнути дефектів механічного кулачка, таких як те, що його легко пошкодити та він не підходить для високошвидкісного обертання та передачі.

Контролер руху підтримує функцію електронної камери. Користувач може редагувати криву кулачка у відповідному програмному забезпеченні редактора кулачка.

Після редагування криву кулачка потрібно викликати в програмі керування рухом. Програма керування рухом може викликати криву кулачка за допомогою інструкції MC_CamIn.

11.4.5 MC_CamIn

FB/FC	Пояснення	Застосовна модель
FB	MC_CamIn використовується для побудови співвідношення кулачків між двома осями відповідно до заданих параметрів.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
майстер	Вкажіть номер головної осі в роботі електронного кулачка.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Підлеглий	Вкажіть номер веденої осі в роботі електронного кулачка.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
ContinuousUpdate	Зарезервований			
CamTable	Укажіть таблицю кулачків, яка використовується для побудови зв'язку між головною віссю та веденою віссю	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Періодичні	Укажіть, чи виконувати вказану таблицю cam періодично чи лише один період.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
МайстерАбсолют	Вкажіть режим положення головної осі. TRUE: Абсолютна позиція FALSE: Відносна позиція	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
SlaveAbsolute	Вкажіть режим положення веденої осі. TRUE: Абсолютна позиція FALSE: Відносна позиція	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Execute* змінюється з FALSE на TRUE
MasterOffset	Укажіть, на скільки одиниць зсувається позиція головної осі. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число і 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
SlaveOffset	Укажіть, на скільки одиниць зміщується положення підпорядкованої осі. (Одиниця: Одиниця)	LREAL	Від'ємне число, додатне число і 0 (0)	Коли Execute змінюється з FALSE на TRUE
MasterScaling	Вкажіть масштабування положення головної осі.	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли Execute змінюється з FALSE на TRUE
SlaveScaling	Вкажіть масштаб положення осі веденого пристрою.	LREAL	Додатне число або від'ємне число (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
MasterStartDistance	Зарезервований			
MasterSyncPosition	Зарезервований			
ActivationPosition	Укажіть положення головної осі на початку зачеплення. Іншими словами, коли провідна вісь проходить позицію, ведена вісь починає виконувати дію зачеплення.	LREAL	Від'ємне число, додатне число і 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
ActivationMode	Вкажіть режим позиції, з якої розпочати залучення	MC_ACTIVATION_MODE	0: mcRelative (відносне положення осі) 1: mcAbsolute (абсолютне положення осі) 2: mcPhase_Axis (Абсолютна фаза осі) 3: mcPhase_CAM (Абсолютна фаза кулачка) (0)	Коли Execute змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
StartMode	Вкажіть, як ведена вісь виконує дію зачеплення.	MC_START_MODE	0: mcRampInShortest (найкоротший шлях) 1: mcRampInPositive (позитивний напрямок) -1: mcRampInNegative (Негативний напрямок) (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Укажіть максимальну швидкість штабелювання веденої осі протягом періоду, коли ведена вісь виконує дію зачеплення. (Одиниця: одиниця/секунда)	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
Прискорення	Вкажіть максимальне прискорення веденої осі протягом періоду, коли ведена вісь виконує дію зачеплення. (Одиниця: одиниця/секунда ^{2})	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
Уповільнення	Вкажіть максимальне уповільнення веденої осі протягом періоду, коли ведена вісь виконує дію зачеплення. (Одиниця: одиниця/секунда ^{2} .	LREAL	Позитивне число (Значення змінної повинно бути встановленим)	Коли *Execute* змінюється з FALSE на TRUE
ривок	Зарезервований	-	-	-
MasterValueSource	Укажіть тип положення головної осі в розрахунку електронного кулачка.	MC_SOURCE	0:mcSetValue 1:mcActualValue (0)	Коли Execute змінюється з FALSE на TRUE
BufferMode	Вкажіть поведінку під час виконання двох інструкцій.	MC_Buffer_Mode	0: mcAborting 1: mcBuffered (0)	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. Виконання інструкції MC_CamIn починається, коли *Execute* змінюється з FALSE на TRUE. Зміна *параметра Execute* з TRUE на FALSE не впливає на виконання інструкції під час виконання інструкції.
2. зміна *Execute* з FALSE на TRUE не впливає на виконання інструкції під час виконання інструкції. Інструкція продовжиться в попередньому порядку.
3. Зверніться до розділу 10.3, щоб дізнатися більше про *BufferMode* .

- **Вихідні параметри**

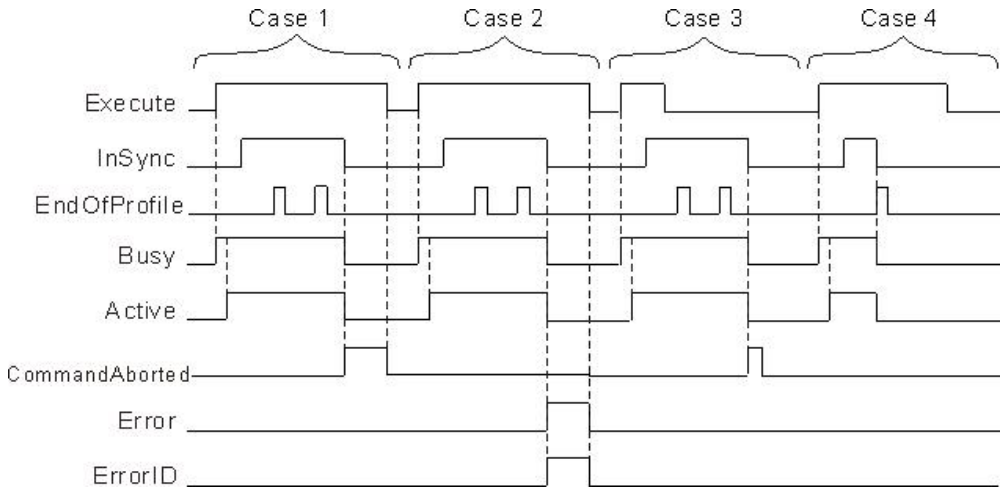
Назва параметра	функція	Тип даних	Допустимий діапазон
InSync	TRUE, коли головна вісь і ведена вісь рухаються синхронно на основі кривої кулачка.	BOOL	TRUE / FALSE
EndOfProfile	TRUE, коли рух кулачка досягає кінцевої точки.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь керується.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
InSync	◆ Коли ведена вісь і головна вісь синхронні в русі кулачка.	◆ Коли взаємозв'язок кулачка між веденою віссю та головною віссю роз'єднаний. ◆ Коли виконується ациклічний рух кулачка (*Periodic* = FALSE) і *EndOfProfile* змінюється на TRUE ◆ Коли *CommandAborted* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
EndOfProfile	◆ Коли рух кулачка досягає кінцевої точки в столі кулачка.	◆ Через крапку *EndOfProfile* змінюється на TRUE
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли виконується ациклічний рух кулачка (*Periodic* = FALSE) і *EndOfProfile* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAborted* змінюється на TRUE
Active	◆ Коли інструкція починає керувати осями	◆ Коли виконується ациклічний рух кулачка (*Periodic* = FALSE) і *EndOfProfile* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAborted* змінюється на TRUE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією руху	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через один період *активний* змінюється на TRUE. Коли ведена вісь і головна вісь перебувають у синхронному русі, *InSync* змінюється з FALSE на TRUE. Коли досягається кінцева точка циклу кулачка, *EndOfProfile* змінюється з FALSE на TRUE і змінюється на FALSE через один цикл. Коли взаємозв'язок кулачка між підпорядкованою віссю та головною віссю розривається (наприклад, шляхом виконання інструкції MC_CamOut), *CommandAborted* змінюється з FALSE на TRUE, а *InSync, Busy* та *Active* змінюються з TRUE на FALSE. Після цього *CommandAborted* змінюється з TRUE на FALSE, а *Execute* змінюється з TRUE на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *помилка* змінюється з FALSE на TRUE, *ErrorID* показує відповідні коди помилок, а *InSync* , *Busy* та *Active* змінюються з TRUE на FALSE. Після цього *Error* змінюється з TRUE на FALSE, а значення *ErrorID* змінюється на 0, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : виконання інструкції продовжується після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Час зміни стану *InSync* , *EndOfProfile, Busy* та *Active* узгоджується зі станом, у якому вони перебувають, оскільки *Execute* має значення TRUE. Після цього *параметри InSync* , *Busy* і *Active* змінюються з TRUE на FALSE після роз'єднання кулачка між підпорядкованою віссю та головною віссю. Тим часом CommandAborted змінюється з FALSE на TRUE і змінюється на FALSE через один цикл.

Випадок 4 : якщо рух кулачка виконується ациклічним способом (*Periodic* = FALSE), *EndOfProfile* змінюється з FALSE на TRUE, коли досягається кінцева точка циклу кулачка. Тим часом *InSync* , *Busy* та *Active* змінюються з TRUE на FALSE, а *EndOfProfile* змінюється з TRUE на FALSE через один цикл.

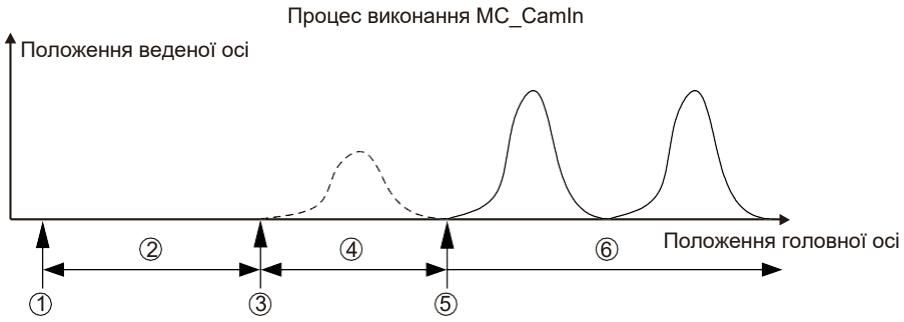
- **функція**

Інструкція *MC_CamIn* використовується для синхронного переміщення веденої осі та головної осі відповідно до запланованого співвідношення кулачка. Інструкція *MC_CamOut* використовується для роз'єднання кулачка між двома осями.

■ **Про MC_CamIn Інструкція**

➢ **Процес виконання *MC_CamIn***

Фігура процесу виконання MC_CamIn:



Етап 1: Запуск і виконання інструкції MC_CamIn.
Етап 2: Дочекайтеся початку заручин.
Етап 3: ведена вісь починає виконувати дію зачеплення, коли головна вісь досягає положення, де починається зачеплення.
Етап 4: Залучення триває.
Етап 5: головна вісь і ведена вісь досягають синхронізації після завершення залучення.
Етап 6: Головна вісь і ведена вісь знаходяться в синхронному русі.

Етап 1: Запуск і виконання інструкції MC_CamIn.
інструкція *MC_CamIn* , а потім підлеглий пристрій негайно переходить у стан очікування початку залучення.

ПРИМІТКА. *Я*кщо *ActivationPosition* = 0 і *ActivationMode* = 0 (відносне положення осі), ведена вісь перейде з поточної швидкості на швидкість SYNC. За винятком вищезазначеного випадку, ведена вісь негайно припинить рух! Усі встановлені вхідні параметри інструкції *MC_CamIn* будуть прочитані та збережені для використання під час виконання.

Етап 2: Дочекайтеся початку заручин.
Підпорядкована вісь очікує часу для виконання дії зачеплення в стані очікування. Час для початку залучення - коли головна вісь проходить позицію, визначену параметром *ActivationPosition* . За різних обставин період часу, який чекає ведена вісь, різний. Якщо під час виконання інструкції *MC_CamIn* головна вісь знаходиться в положенні, визначеному *ActivationPosition* , підпорядкована вісь негайно починає дію залучення. Якщо головна вісь ніколи не досягне положення, визначеного *ActivationPosition,* ведена вісь ніколи не почне виконувати дію залучення, а синхронізація кулачка ніколи не відбудеться. На цьому етапі використовуються параметри *ActivationPosition* і *ActivationMode* .
Етап 3: підпорядкована вісь починає виконувати дію залучення, коли головна вісь проходить позицію, визначену *ActivationPosition.* Параметри *MasterAbsolute, SlaveAbsolute* , *MasterOffset* , *SlaveOffset* , *MasterScaling* і *SlaveScaling* працюватимуть на даний момент для забезпечення відповідності між положенням головної осі та положенням підпорядкованої осі та фазою кулачка.
Етап 4: Залучення триває.
Ведувана вісь виконує зачеплення у спосіб, визначений параметром *StartMode* . Крім *StartMode,* на цьому етапі також працюють параметри *Velocity, Acceleration* і *Deceleration* . Характеристики руху щодо швидкості, прискорення/уповільнення веденої осі визначаються цими параметрами в зачепленні.

Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації. Зчеплення завершено, і підпорядкована вісь і головна вісь досягають синхронізації кулачка, якщо фаза кулачка, якій відповідають головна вісь і підпорядкована вісь, відповідає запланованому взаємозв'язку кулачка після того, як підпорядкована вісь починає виконувати дію зачеплення.

ПРИМІТКА. На малюнку вище встановлене положення головної осі в момент початку залучення більше, ніж головне положення в момент початку виконання інструкції *MC_CamIn* . Подібний спосіб також застосовується до тієї обставини, що задане положення головної осі в момент початку залучення менше або дорівнює головному положенню в момент початку виконання інструкції *MC_CamIn .*

■ **ActivationPosition**

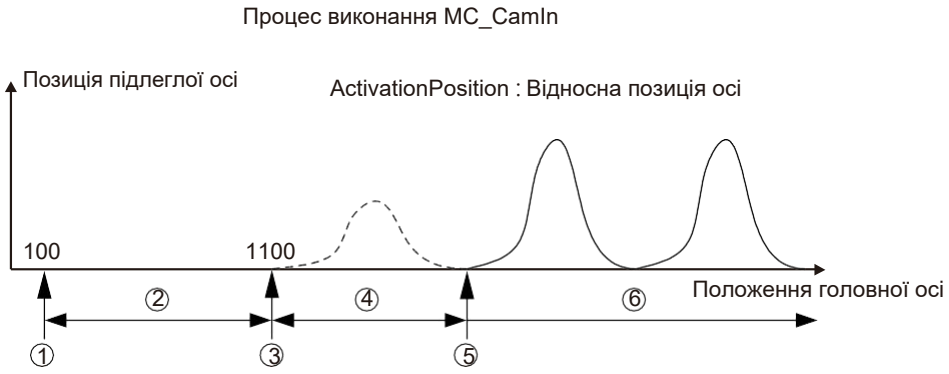
Параметр *ActivationPosition* — це початкове положення зачеплення кулачка (це положення головної осі). Іншими словами, підпорядкована вісь починає виконувати зачеплення, коли головна вісь досягає положення, визначеного *ActivationPosition* після запуску та виконання інструкції *MC_CamIn .*

ActivationPosition може бути положенням головної осі, фазою головної осі, фазою кулачка головної осі, які можна вибрати за допомогою параметра *ActivationMode* .

➢ ***ActivationPosition:* відносне положення осі**

Оскільки *ActivationMode* =0, *ActivationPosition* є положенням осі, яке є відносно положення головної осі під час виконання інструкції *MC_CamIn .* Положення головної осі на момент початку фактичного залучення є значенням *ActivationPosition* плюс головне положення моменту початку виконання інструкції *MC_CamIn .*

Наприклад: позиція головної осі становить 100, а *ActivationPosition* 1000 у момент початку виконання інструкції *MC_CamIn* . Положення головної осі становить 1100 (1100=100+1000), коли починається фактичне зачеплення.



Етап 1: Запуск і виконання інструкції MC_CamIn. Абсолютне положення головної осі на даний момент становить 100.

Етап 2: Дочекайтеся початку заручин.

Етап 3: головна вісь досягає положення для початку зачеплення (1100), а ведена вісь починає виконувати дію зачеплення.

Етап 4: Залучення триває.

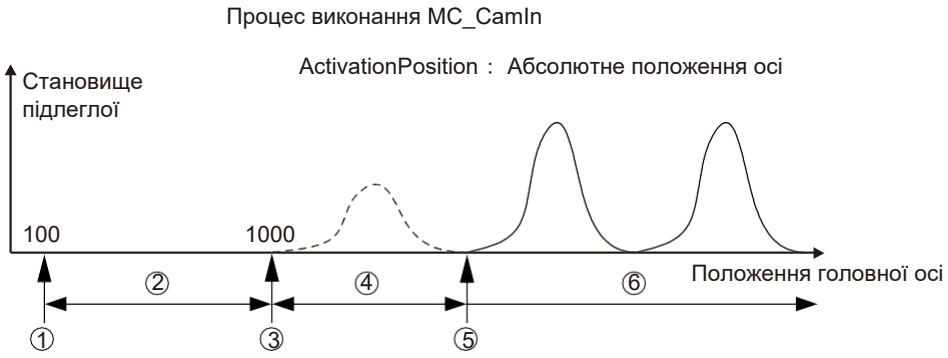
Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації.

Етап 6: головна вісь і ведена вісь знаходяться в синхронному русі.

➢ ***ActivationPosition* : абсолютна позиція осі**

Коли *ActivationMode* =1, *ActivationPosition* є положенням осі, яке є абсолютним щодо положення головної осі під час виконання інструкції *MC_CamIn .* Положенням головної осі на початку фактичного залучення є *ActivationPosition* .

Наприклад: позиція головної осі становить 100, а *ActivationPosition* 1000 у момент початку виконання інструкції *MC_CamIn* . Позиція головної осі становить 1000 (1000= *ActivationPosition*), коли починається фактичне залучення.



Етап 1: Запуск і виконання інструкції MC_CamIn. Абсолютне положення головної осі на даний момент становить 100.

Етап 2: Дочекайтеся початку заручин.

Етап 3: головна вісь досягає положення для початку зачеплення (1000), а ведена вісь починає виконувати дію зачеплення.

Етап 4: Ведеться залучення.

Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації.

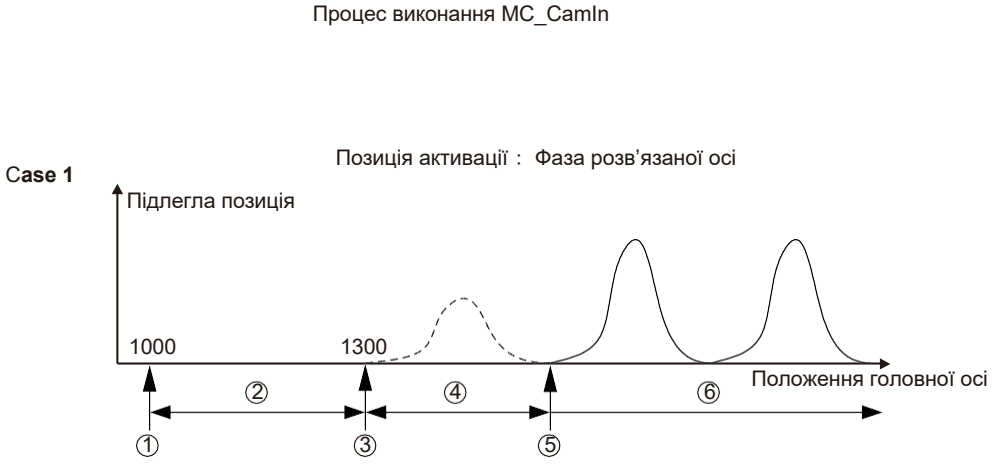
Етап 6: Головна вісь і ведена вісь знаходяться в синхронному русі.

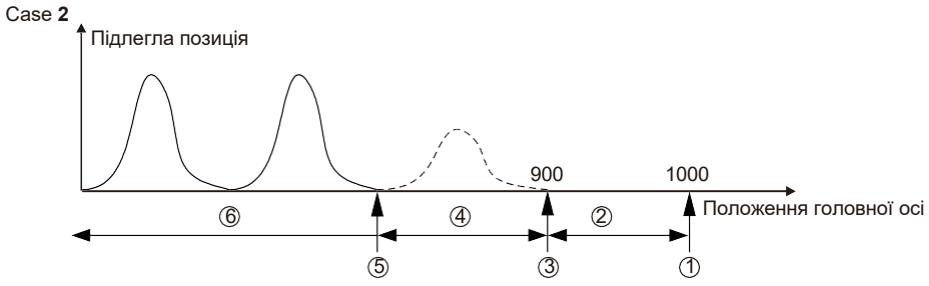
- ***ActivationPosition* : абсолютна фаза осі**

Коли *ActivationMode* =2, *ActivationPosition* — це абсолютна фаза осі, яка є залишком, отриманим шляхом ділення абсолютної позиції осі на модуль. Підпорядкована вісь починає виконувати дію залучення, оскільки абсолютна фаза головної осі – *ActivationPosition* .

Фаза абсолютної осі є циклічною. Його абсолютна фаза осі може дорівнювати *ActivationPosition* багато разів у русі головної осі. Але підпорядкована вісь починає виконувати дію залучення лише тоді, коли абсолютна фаза осі головної осі дорівнює *ActivationPosition* вперше після виконання інструкції MC_CamIn.

Наприклад, модуль головної осі дорівнює 400, *ActivationPosition* =100, а позиція головної осі становить 1000 під час виконання інструкції *MC_CamIn .* Підпорядкована вісь не виконуватиме дію залучення, оскільки абсолютна фаза осі головної осі становить 200 (200=1000%400) під час виконання інструкції *MC_CamIn .* Підпорядкована вісь починає виконувати дію зачеплення, коли положення головної осі становить 1300 (абсолютна фаза осі становить 100=1300%400) або 900 (абсолютна фаза осі становить 100=900%400). (% означає математичну операцію для знаходження залишку)





Етап 1: Запуск і виконання інструкції MC_CamIn. Абсолютне положення головної осі на даний момент становить 1000. (Абсолютна фаза осі дорівнює 200)

Етап 2: Дочекайтеся початку заручин.

Етап 3: головна вісь досягає положення для початку зчеплення (1300 в обставині 1 і 900 в обставині 2), а ведена вісь починає виконувати дію зчеплення.

Етап 4: Ведеться залучення.

Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації.

Етап 6: головна вісь і ведена вісь знаходяться в синхронному русі.

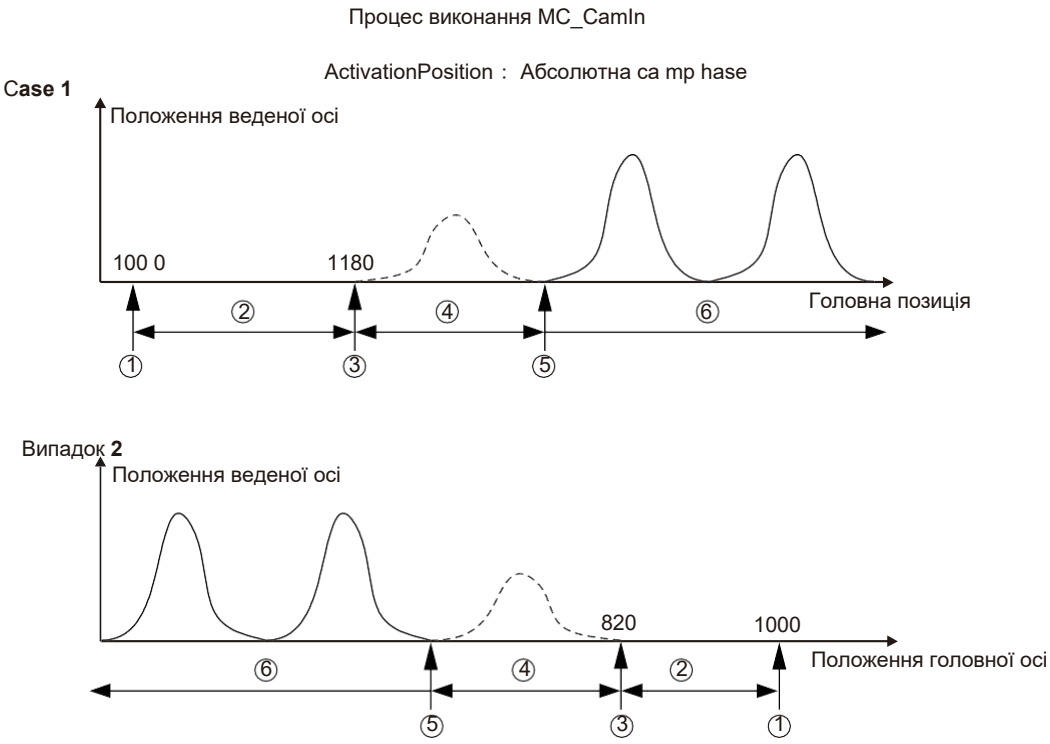
ПРИМІТКА. Оскільки *ActivationPosition* є абсолютною фазою осі, діапазон значення параметра *ActivationPosition* становить 0~модуль (за винятком модуля). Якщо значення *ActivationPosition* перевищить дійсний діапазон, станеться помилка, і виконання інструкції буде невдалим, оскільки буде виконано інструкцію *MC_CamIn* .

- ***ActivationPosition* : абсолютна фаза кулачка**

Коли *ActivationMode* =3, *ActivationPosition* — це абсолютна фаза кулачка, яка є залишком, отриманим шляхом ділення абсолютної позиції осі на її цикл кулачка. Підпорядкована вісь починає виконувати дію залучення, оскільки фаза кулачка головної осі є *ActivationPosition* .

Кулачкова фаза є циклічною. Його фаза кулачка може дорівнювати *ActivationPosition* багато разів під час руху головної осі. Але підпорядкована вісь починає виконувати дію залучення лише тоді, коли фаза кулачка головної осі дорівнює *ActivationPosition* вперше після виконання інструкції MC_CamIn.

Наприклад, максимальне положення головної осі в таблиці кулачків становить 360, *ActivationPosition* =100, а положення головної осі становить 1000 під час виконання інструкції *MC_CamIn* . Підпорядкована вісь не виконуватиме дію залучення, оскільки абсолютна фаза кулачка головної осі становить 280 (280=1000%360) у момент початку виконання інструкції *MC_CamIn* . Потім ведена вісь починає виконувати дію зачеплення, оскільки положення головної осі становить 1180 (абсолютна фаза кулачка становить 100=1180%360) або 820 (абсолютна фаза кулачка становить 100=820%360).



Етап 1: Запуск і виконання інструкції MC_CamIn. Абсолютне положення головної осі на даний момент становить 1000. (Абсолютна фаза кулачка становить 280)

Етап 2: Дочекайтеся початку заручин.

Етап 3: головна вісь досягає положення для початку зчеплення (положення головної осі становить 1180 в обставині 1 і 820 в обставині 2), а ведена вісь починає виконувати дію зчеплення.

Етап 4: Ведеться залучення.

Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації.

Етап 6: головна вісь і ведена вісь знаходяться в синхронному русі.

Примітка. Оскільки *ActivationPosition* є абсолютною фазою кулачка, діапазон значення параметра *ActivationPosition* становить 0~ значення циклу кулачка (за винятком значення циклу кулачка). Якщо значення *ActivationPosition* перевищує дійсний діапазон, виникне помилка, і виконання не вдасться, оскільки виконується інструкція *MC_CamIn .*

■ Зв’язок між положенням головної осі та положенням підлеглої осі

Співвідношення кулачка, яке планується в програмному забезпеченні, є співвідношенням позицій між головною віссю та веденою віссю. «Положення», згадане тут, є фазою кулачка головної осі/підпорядкованої осі замість фактичного положення осі. Якщо запланований зв’язок кулачка розглядається як функція CAM, як показано нижче, вхід функції CAM є фазою кулачка головної осі, а вихід – фазою кулачка підлеглої осі. Формула показана нижче.

y = CAM (x)

x : Фаза кулачка головної осі

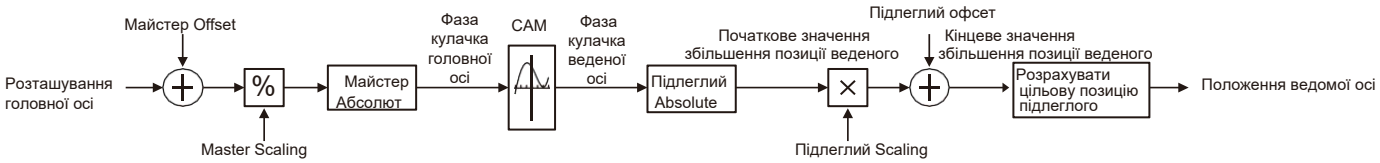
y : Фаза кулачка веденої осі

Фаза кулачка походить від положень осі, і між ними відбувається перетворення. Перетворення між положенням осі та фазою кулачка пов’язане з параметрами *MasterAbsolute* , *SlaveAbsolute* , *MasterOffset* , *SlaveOffset* , *MasterScaling* і *SlaveScaling* .

Для отримання додаткової інформації зверніться до відповідних розділів.

/

Підпорядкована вісь слідує за головною віссю, щоб здійснювати синхронний рух кулачка за допомогою інструкції MC *_CamIn* . У синхронному русі кулачка відповідне співвідношення між положенням головної осі та положенням підлеглої осі базується на попередньо запланованому співвідношенні кулачка (крива кулачка або таблиця кулачка). Процес, у якому положення підпорядкованої осі обчислюється через положення головної осі, проілюстровано таким чином.



■ ***MasterAbsolute* і *SlaveAbsolute***

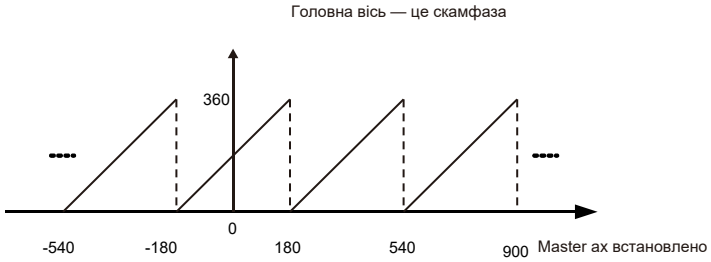
Параметр *MasterAbsolute* використовується для визначення відповідного співвідношення між положенням головної осі та фазою кулачка. Оскільки *MasterAbsolute* дорівнює TRUE, положення головної осі та фаза кулачка знаходяться в абсолютному співвідношенні. Оскільки *MasterAbsolute* має значення FALSE, положення головної осі та фаза кулачка перебувають у відносному зв'язку. Для *SlaveAbsolute* пояснення подібне до *MasterAbsolute* .

MasterAbsolute і *SlaveAbsolute* працюють в момент початку залучення. Тобто відповідне співвідношення між положенням осі та фазою кулачка будується на початку зачеплення. (**ПРИМІТКА:** відповідний зв'язок створюється не під час початку виконання інструкції *MC_CamIn* , а під час залучення.) Після цього фаза кулачка обчислюється відповідно до відповідного зв'язку.

➢ **Відносний режим**

Положення головної осі та її фаза кулачка знаходяться у відносному зв'язку, оскільки параметр *MasterAbsolute* має значення FALSE. Тобто положення головної осі відповідає його фазі кулачка 0 у момент початку зачеплення. Після цього фаза головного кулачка буде розрахована відповідно до відповідного співвідношення. Наприклад, головна вісь знаходиться у відносному режимі, максимальне значення фази кулачка головної осі у взаємозв'язку кулачка становить 360, а положення головної осі – 180 у момент початку зачеплення. Таким чином, положення головної осі 180 відповідає її фазі кулачка 0; позиція головної осі 200 відповідає її фазі кулачка 20 (20= (200-180) %360) і так далі.

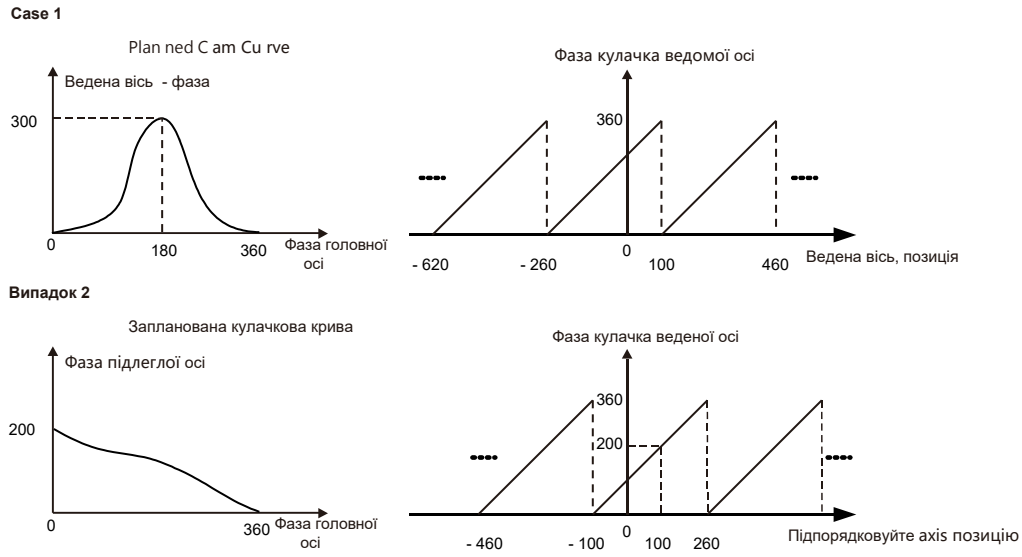
У цьому випадку положення головної осі відповідає її фазі кулачка, як показано на наступному малюнку.



Оскільки параметр *SlaveAbsolute* має значення FALSE, позиція ведомої осі та її фаза кулачка перебувають у відносному зв'язку. Тобто фаза кулачка підпорядкованої осі та фаза кулачка головної осі відповідають запланованому співвідношенню кулачка в момент початку зачеплення. Якщо ведена вісь перебуває у відносному режимі, метод перевірки фази кулачка веденої осі відрізняється від провідної осі. Коли фаза кулачка підпорядкованої осі впевнена, вона повинна відповідати умові, що фаза кулачка підпорядкованої осі та фаза кулачка головної осі відповідають запланованому співвідношенню кулачка на момент початку зачеплення.

Наприклад, ведена вісь знаходиться у відносному режимі, максимальне значення фази кулачка веденої осі у взаємозв'язку кулачка становить 360, а позиція веденої осі становить 100 у момент початку зачеплення. Якщо фаза кулачка головної осі дорівнює 0 на даний момент (а фаза кулачка підпорядкованої осі дорівнює 0, як вимагається у взаємозв'язку кулачка), позиція 100 веденої осі відповідатиме її фазі кулачка 0, як показано в наступній обставині 1. Якщо Фаза кулачка веденої осі дорівнює 200 відповідно до вимог кулачка

позиція 100 осі підпорядкованого пристрою відповідатиме її фазі 200 кулачка, як показано в наступній обставині 2.



- **Абсолютний режим**

Коли параметр *MasterAbsolute* має значення TRUE, позиція головної осі та її фаза кулачка знаходяться в абсолютному співвідношенні. У будь-який час фаза кулачка головної осі дорівнює залишку, отриманому шляхом ділення положення головної осі в цей час на максимальне значення фази кулачка головної осі у співвідношенні кулачка.

Наприклад, головна вісь знаходиться в абсолютному режимі, а максимальне значення головної осі у відношенні кулачка становить 360. Отже, його фаза кулачка дорівнює 100, оскільки позиція головної осі дорівнює 100 (100=100%360); його фаза кулачка становить 140 (140=500%360), оскільки положення головної осі становить 500 і так далі. Положення головної осі відповідає її фазі кулачка, як показано на малюнку нижче.



Коли параметр *SlaveAbsolute* має значення TRUE, позиція ведомої осі та її фаза кулачка знаходяться в абсолютному співвідношенні. У будь-який час фаза кулачка підпорядкованої осі дорівнює залишку, отриманому шляхом ділення положення підпорядкованої осі в цей час на максимальне значення фази кулачка підпорядкованої осі у взаємозв'язку кулачка. Коли ведена вісь знаходиться в абсолютному режимі, відповідне співвідношення між положенням веденої осі та її фазою кулачка узгоджується зі співвідношенням між положенням ведучої осі та її фазою кулачка, коли головна вісь перебуває в абсолютному режимі.

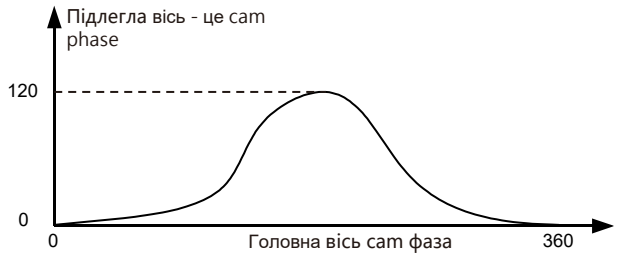
■ *Зсув* і *масштабування*

Співвідношення кулачка між головною віссю та веденою віссю попередньо сплановано. Але коли рух кулачка виконується, зміщення позиції або масштабування на основі попередньо запланованого співвідношення кулачка можна виконати за допомогою встановлення параметрів *«Зміщення»* та *«Масштабування»* . Наприклад, існують різні розміри для одного продукту, який обробляється. Необхідно спланувати лише одне співвідношення кулачка, а потім зміна значень *«Зміщення»* та *«Масштабування»* відповідає обробці виробів різних розмірів.

Параметр *MasterOffset* дійсний лише тоді, коли головна вісь знаходиться в абсолютному або відносному режимі. (*MasterAbsolute* = TRUE або FALSE). Параметр *SlaveOffset* дійсний лише тоді, коли ведена вісь знаходиться всередині

абсолютний режим (*SlaveAbsolute* = TRUE). Параметр *SlaveOffset* недійсний, оскільки ведена вісь перебуває у відносному режимі (*SlaveAbsolute* = FALSE).

Зміщення позиції та масштабування головної осі та підлеглої осі визначають фактично виконане співвідношення кулачка. Ефект описано в наступному прикладі. Планове співвідношення кулачків показано на малюнку нижче.

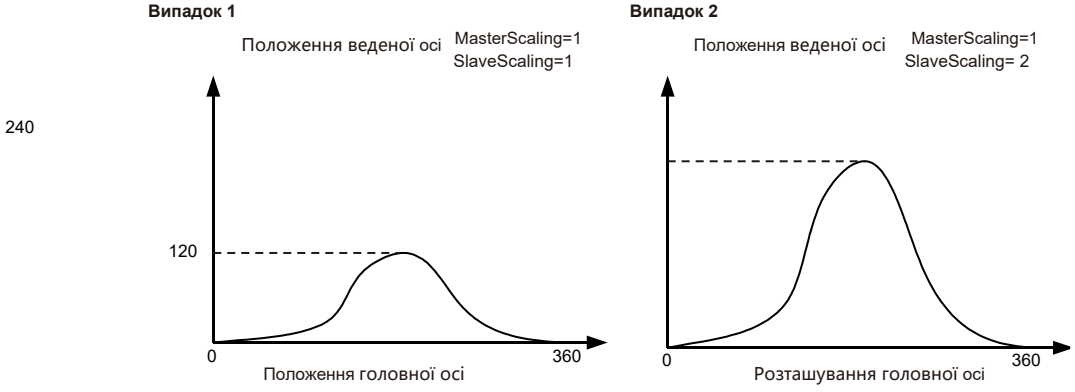


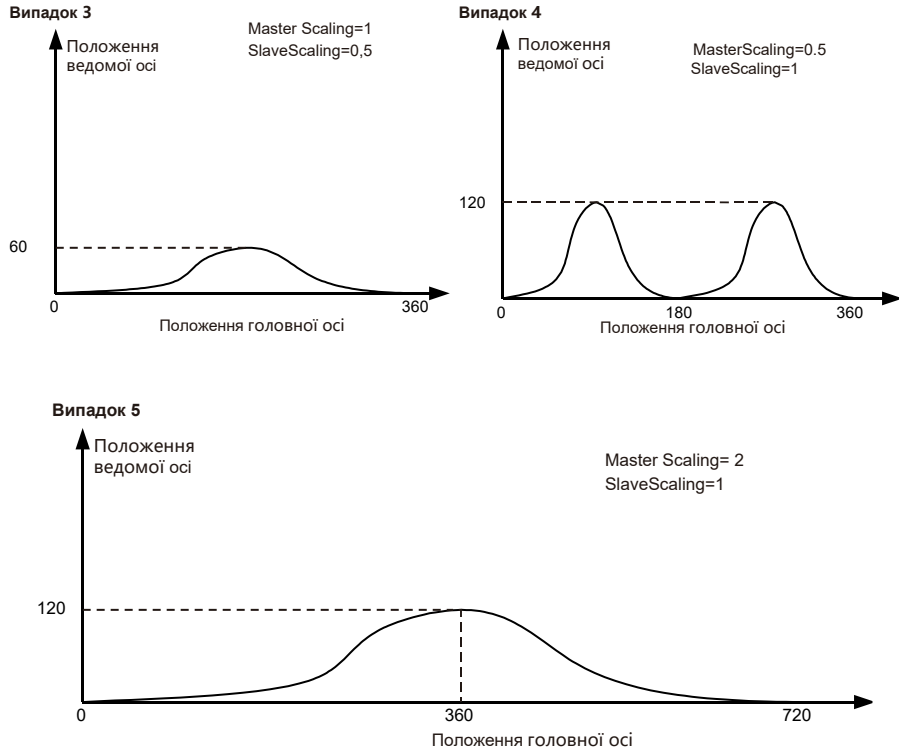
Коли головна вісь і підпорядкована вісь знаходяться в абсолютному режимі та починається залучення, положення головної осі та положення підлеглої осі дорівнюють 0. Якщо немає зміщення позиції та масштабування (зміщення та масштабування є значеннями за замовчуванням), фактична головна вісь положення осі відповідає фактичному положенню осі підпорядкованого під час виконання руху кулачка, як показано на наступному малюнку.



Якщо зміщення та масштабування не є значеннями за замовчуванням, відповідне співвідношення між фактичним положенням головної осі та фактичним положенням підлеглої осі впливає під час виконання руху кулачка, як показано нижче.

- **MasterOffset:0 і SlaveOffset:0 і вплив MasterScaling і SlaveScaling на взаємозв'язок камери**





Випадок 1 : фактичне співвідношення кулачків узгоджується із заздалегідь запланованим, оскільки значення MasterScaling і SlaveScaling дорівнюють 1, а їхні зміщення дорівнюють 0.

Випадок 2 : Положення підлеглого, що відповідає положенню головної осі, у два рази більше, ніж заплановано у взаємозв'язку кулачка, оскільки значення *MasterScaling* дорівнює 1, *SlaveScaling* дорівнює 2, а їх зсуви дорівнюють 0.

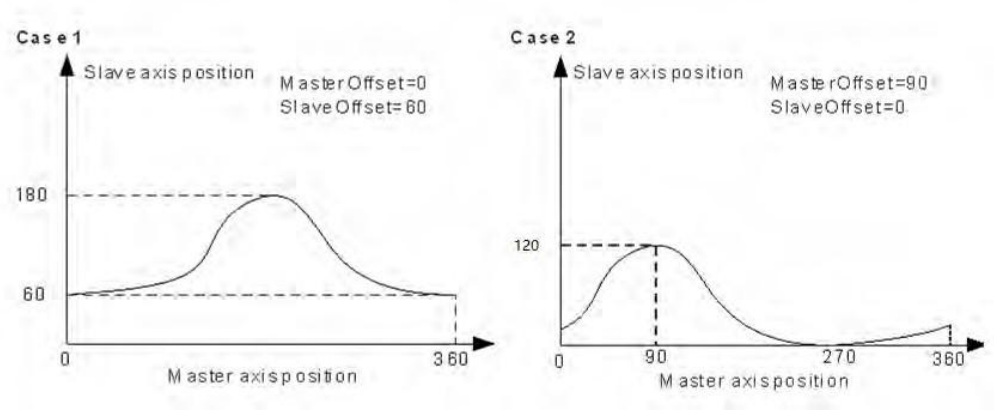
Випадок 3 : Положення підлеглого, що відповідає положенню головної осі, дорівнює 1/2 від запланованого співвідношення кулачка, оскільки значення MasterScaling дорівнює 1, SlaveScaling дорівнює 0,5, а їх зсуви дорівнюють 0.

Випадок 4 : Положення головної осі, що відповідає положенню підпорядкованої осі, становить 1/2 від запланованого, оскільки значення *MasterScaling* дорівнює 0,5, *SlaveScaling* дорівнює 1, а їх зсуви дорівнюють 0. Якщо це спостерігати з точки зору фази кулачка, фаза кулачка головної осі становить 1/2 від попередньо запланованої. Тобто цикл головного кулачка змінюється з 360 до 180 (180=360*0,5), а фаза кулачка веденої осі не змінюється.

Випадок 5 : Положення головної осі, що відповідає положенню підлеглої осі, у 2 рази перевищує заплановане, оскільки значення *MasterScaling* дорівнює 2, *SlaveScaling* дорівнює 1, а їх зсуви дорівнюють 0. Якщо це спостерігати з точки зору фази кулачка, головний фаза кулачка осі в два рази перевищує початкову. Тобто цикл кулачка головної осі змінюється з 360 до 720 (720=360*2), а фаза кулачка веденої осі не змінюється.

- **MasterScaling:1 і SlaveScaling:1 і вплив MasterOffset і SlaveOffset на фактично виконане співвідношення камер**

  MasterOffset означає зміщення фактичної кривої положення осі горизонтально під час виконання руху кулачка. *SlaveOffset* вказує на зміщення кривої положення осі вертикально під час виконання руху кулачка.



Випадок 1 : Положення підпорядкованої осі, що відповідає положенню головної осі, додасться на 60 на основі запланованої позиції, оскільки *MasterScaling* і *SlaveScaling* дорівнюють 1, *MasterOffset* дорівнює 0, а *SlaveOffset* дорівнює 60.

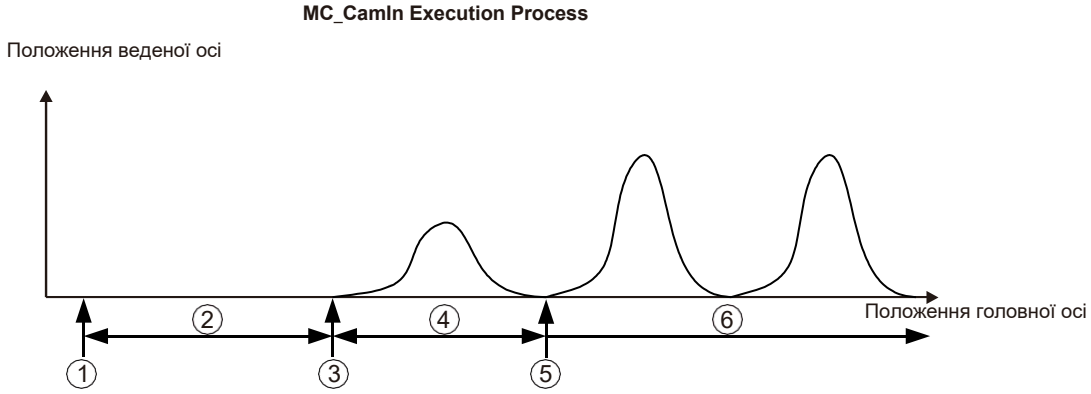
Наприклад, у запланованому співвідношенні кулачка положення 180 головної осі відповідає положенню 120 веденої осі, а в фактичному виконанні відповідне положення веденої осі становить 180 (180=120+60).

Випадок 2 : Положення головної осі, що відповідає положенню підлеглої осі, буде зміщено (додано) на 90 на основі запланованої позиції, оскільки *MasterScaling* і *SlaveScaling* дорівнюють 1, *MasterOffset* — 90, а *SlaveOffset* — 0.

Наприклад, у запланованому взаємозв'язку кулачка позиція 180 головної осі відповідає положенню 120 веденої осі, а в фактичному виконанні позиція 90 ведучої осі відповідає положенню 120 веденої осі, яке є положенням веденої осі, якому відповідає головний положення осі 180 (180=90+90) у запланованому співвідношенні кулачка.

■ StartMode

У залученні спосіб переміщення веденої осі визначається параметром *StartMode* . Тобто *StartMode* працює на етапі 4 у виконанні інструкції *MC_CamIn* , як показано на наступному малюнку.



Етап 1: Запуск і виконання інструкції MC_CamIn.

Етап 2: Дочекайтеся початку заручин.

Етап 3: головна вісь досягає положення, де починається зачеплення, а ведена вісь починає виконувати дію зачеплення.

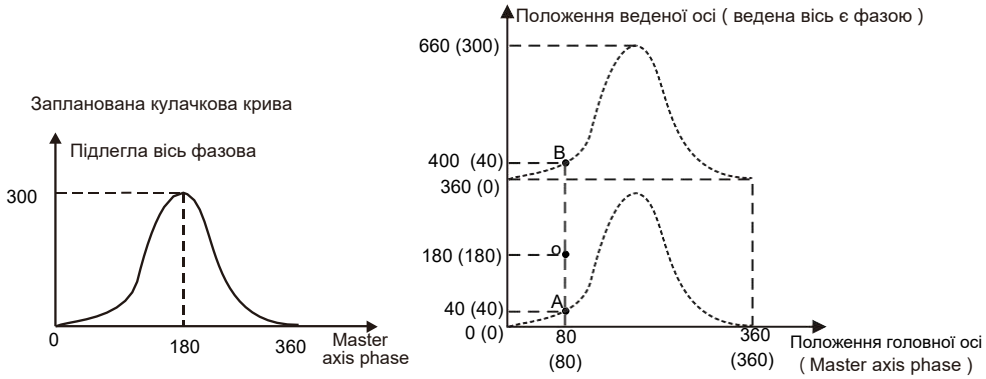
Етап 4: Залучення триває.

Етап 5: Зчеплення завершено, і головна вісь і ведена вісь досягають синхронізації.

Етап 6: Головна вісь і ведена вісь знаходяться в синхронному русі.

Синхронізація кулачка вимагає, щоб фаза кулачка головної осі та фаза кулачка підпорядкованої осі відповідали визначеному співвідношенню кулачка. Процес зачеплення – це процес, під час якого вісь веденого пристрою рухається до синхронної фази. Синхронна фаза та фаза кулачка головної осі відповідають визначеному співвідношенню кулачка. Оскільки фаза кулачка осі є циклічною, кожній фазі кулачка відповідає кілька положень осі. Коли відбувається залучення, існує багато варіантів для очікуваної позиції синхронізації. І, таким чином, є кілька способів залучення для опції.

Наприклад, коли починається зачеплення, фаза кулачка головної осі та фаза кулачка підпорядкованої осі становлять 80 і 180 відповідно як точка O на наступному малюнку. Але визначене співвідношення кулачка вимагає, щоб фаза кулачка підпорядкованої осі становила 40, і, отже, синхронне положення, яке очікує підпорядкована вісь, на даний момент становить 40 або 400 (точка A або точка B на наступному малюнку). Процес залучення від точки O до A або точки O до B можна вибрати за допомогою параметра *StartMode* .



Існує три режими *StartMode* для вибору: найкоротший шлях (mcRamplnShortest), позитивний напрямок (mcRamplnPositive) і негативний напрямок (mcRamplnNegative) . Користувачі можуть вибрати правильний режим залучення відповідно до реальних потреб.

- ***StartMode=0*** (**Найкоротший шлях)**

  Якщо *StartMode* =0, під час виконання дії зчеплення ведена вісь переміщується до позиції для синхронізації, вибираючи найкоротший шлях. На даний момент на рух веденої осі впливають параметри *швидкості, прискорення, уповільнення* та *ривка* .

- ***StartMode=1*** (**Позитивний напрямок**)

  Якщо *StartMode* =1, під час виконання дії зчеплення ведена вісь переміщується до позиції для синхронізації в позитивному напрямку. На даний момент на рух веденої осі впливають параметри *швидкості, прискорення, уповільнення* та *ривка* .

- ***StartMode=-1*** (**Негативний напрямок**)

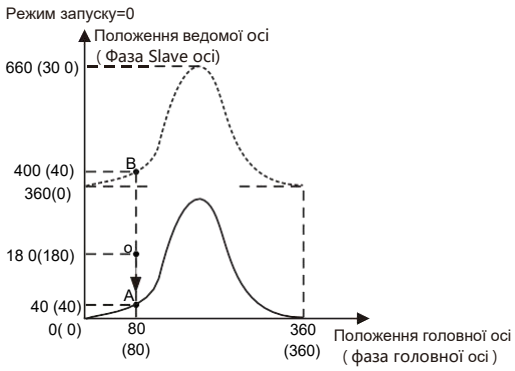
  Якщо *StartMode* =-1, під час виконання дії зчеплення ведена вісь переміщується до позиції для синхронізації у негативному напрямку. На даний момент на рух веденої осі впливають параметри *швидкості, прискорення, уповільнення* та *ривка* .

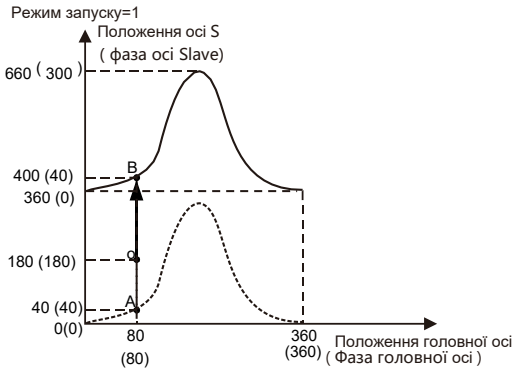
Наприклад, коли починається зачеплення, фаза кулачка головної осі та фаза кулачка веденої осі становлять 80 і 180 відповідно (як точка O нижче). Відповідно до визначеного співвідношення кулачка, фаза кулачка головної осі становить 80, а фаза кулачка підпорядкованої осі становить 40 (як точка A або B нижче). Якщо значення *StartMode* інше, спосіб переміщення підлеглої осі в процесі залучення буде іншим.

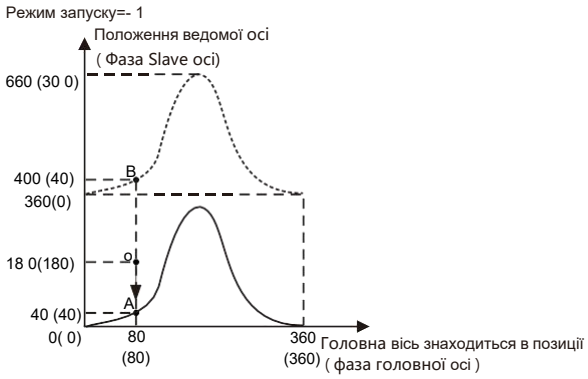
StartMode =0 : ведена вісь переміщується від точки O до точки A, і синхронізація досягається в точки A, оскільки відстань від точки O до точки A менша, ніж від точки O до точки B.

StartMode =1 : Ведуча вісь поступово переміщується від точки O до точки B у позитивному напрямку.

StartMode =-1 : Ведуча вісь поступово переміщується від точки O до точки A у негативному напрямку.







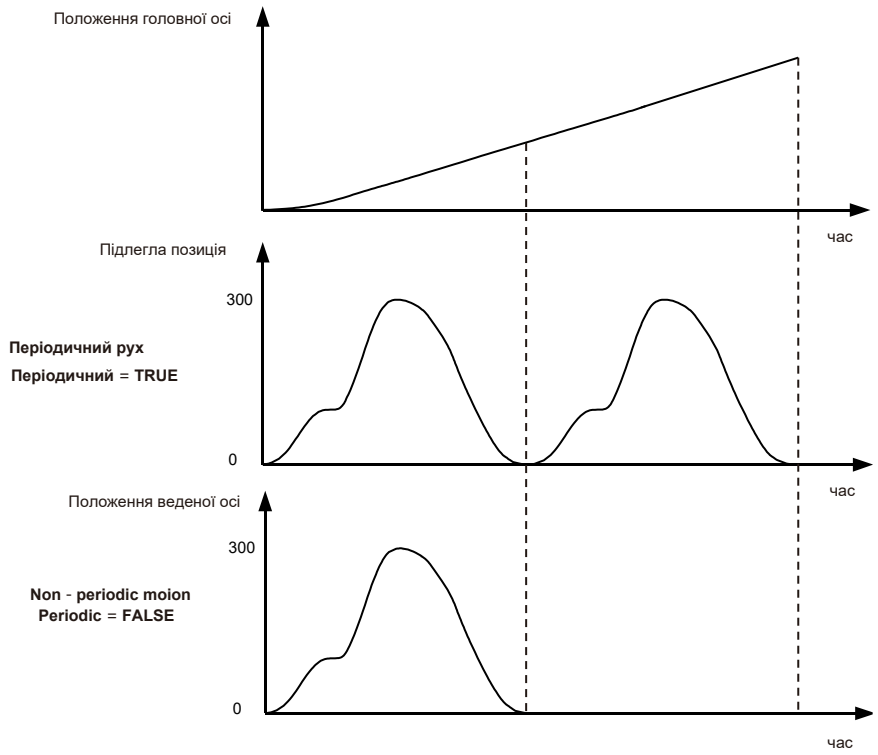
■ **Періодичне/неперіодичне функціонування кулачка (Періодичне)**

У фактичному застосуванні електронних кулачків деякі можуть виконуватися періодично, а деякі просто потрібно виконувати протягом одного циклу. Періодичний параметр використовується для вибору одного з двох випадків руху електронного кулачка .

Якщо *Periodic* = TRUE, ведена вісь слідує за головною віссю, щоб періодично виконувати рух кулачка, доки взаємозв'язок кулачка не роз'єднається.

Як *Periodic* = FALSE, коли досягнуто кінцевої точки циклу кулачка після того, як ведена вісь і головна вісь увійшли в синхронний рух кулачка, зв'язок кулачка між веденою віссю та головною віссю буде роз'єднано, а ведена вісь негайно припинить рух.

Якщо швидкість у кінцевій точці запланованого співвідношення кулачка не дорівнює 0, ведена вісь буде постійно рухатися зі швидкістю роз'єднання після роз'єднання зв'язку кулачка.



■ **Вплив інших інструкцій на роботу камери**

➢ ***MC_CamOut***

Інструкція *MC_CamOut* може бути використана для завершення операції кулачка, яка виконується.

➢ ***MC_SetPosition***

Інструкція *MC_SetPosition* не впливає на інструкції руху, що виконуються. Таким чином, під час роботи кулачка, виконання інструкції *MC_SetPosition* для головної осі та підлеглої осі не вплине на роботу кулачка. Якщо операція кулачка запускається після виконання інструкції *MC_SetPosition* , на кулачок впливатиме зміна положення осі, що відбувається за допомогою інструкції *MC_SetPosition* .

➢ ***MC_Stop* і *MC_Halt***

Коли інструкції *MC_Stop* і *MC_Halt* виконуються на веденій осі, інструкція *MC_CamIn* переривається, взаємозв'язок кулачка роз'єднується, а ведена вісь сповільнюється, поки не зупиниться.

➢ ***MC_Home***

Інструкція *MC_Home* не може бути виконана на підлеглій осі, але на головній. Оскільки інструкція *MC_Home* виконується на головній осі, положення головної осі може сильно змінитися за дуже короткий час, що може спричинити вібрацію підпорядкованої осі. Тому інструкцію *MC_Home* рекомендується виконувати після розриву синхронного зв'язку між двома осями.

■ **Інші запобіжні заходи**

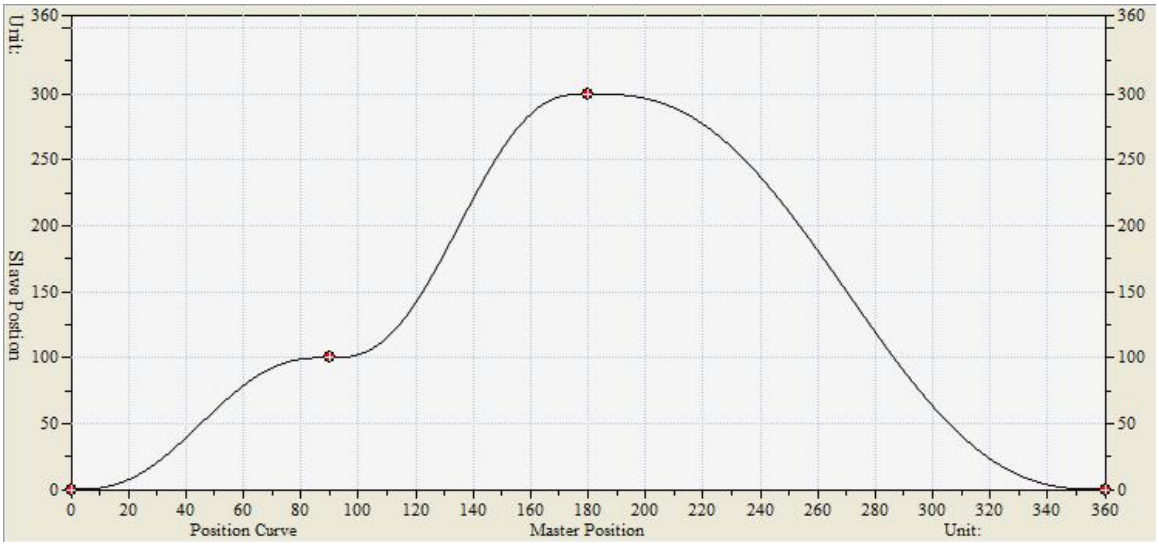
Перегляньте правило для різних типів осей, які працюють як головна вісь або підпорядкована вісь у взаємозв'язку кулачка, у наступній таблиці.

Тип осі	Як головна вісь кулачка	Як кулачок веденої осі
Серво реальної осі	OK	OK
Кодувальник	OK	NO
Віртуальна вісь	OK	OK

11

Приклад програмування

- **Ефект виконання інструкції *MC_CamIn* описано в наступному прикладі.**
  - **Крива кулачка планується, як показано нижче.**



  - **Ключові точки кулачкової кривої**

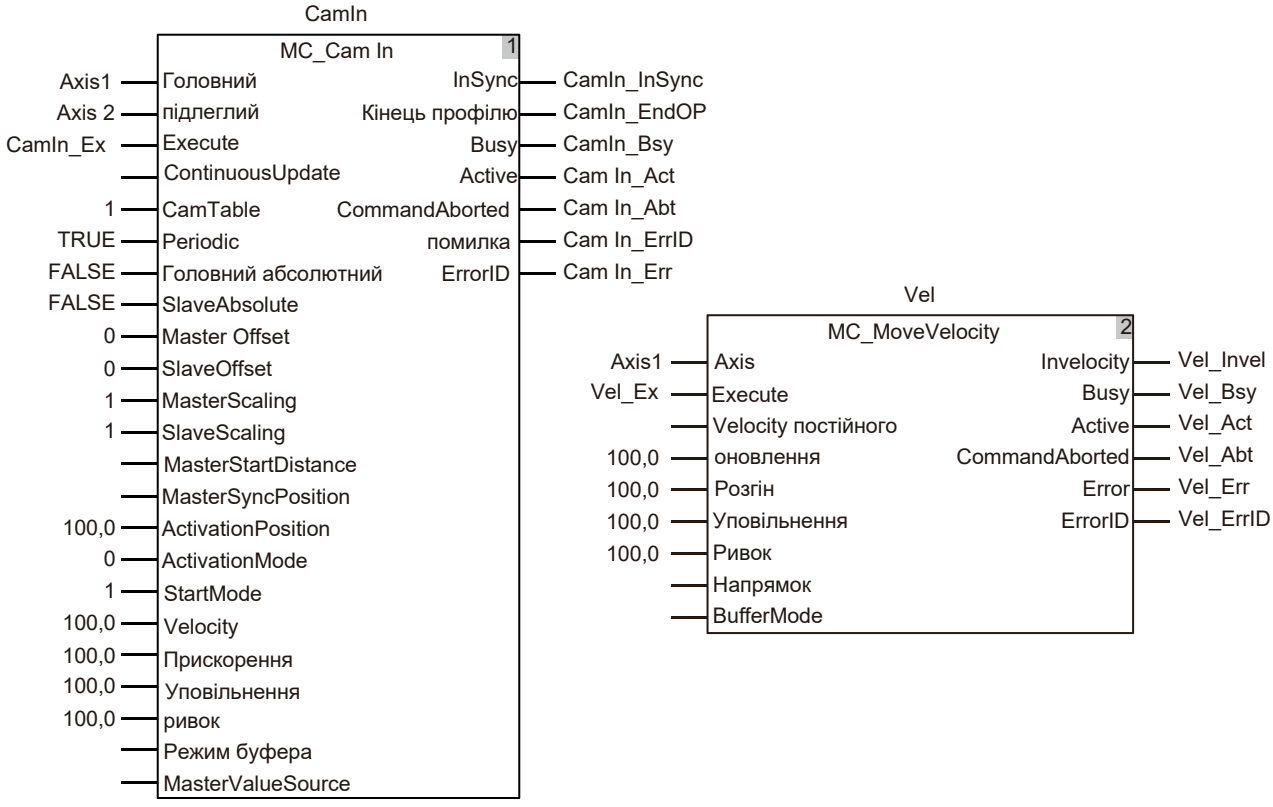
№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
1	0	0	0	0
2	90	100	0	0
3	180	300	0	0
4	360	0	0	0

  - **Пояснення:**

Період кулачка головної осі та веденої осі	360
Master Scaling і SlaveScaling	1
MasterOffset і SlaveOffset	0
МайстерАбсолют	Відносна
SlaveAbsolute	Відносна
Periodic	Періодичні
ActivationPosition	Відносне положення осі: 100
StartMode	Найкоротший шлях

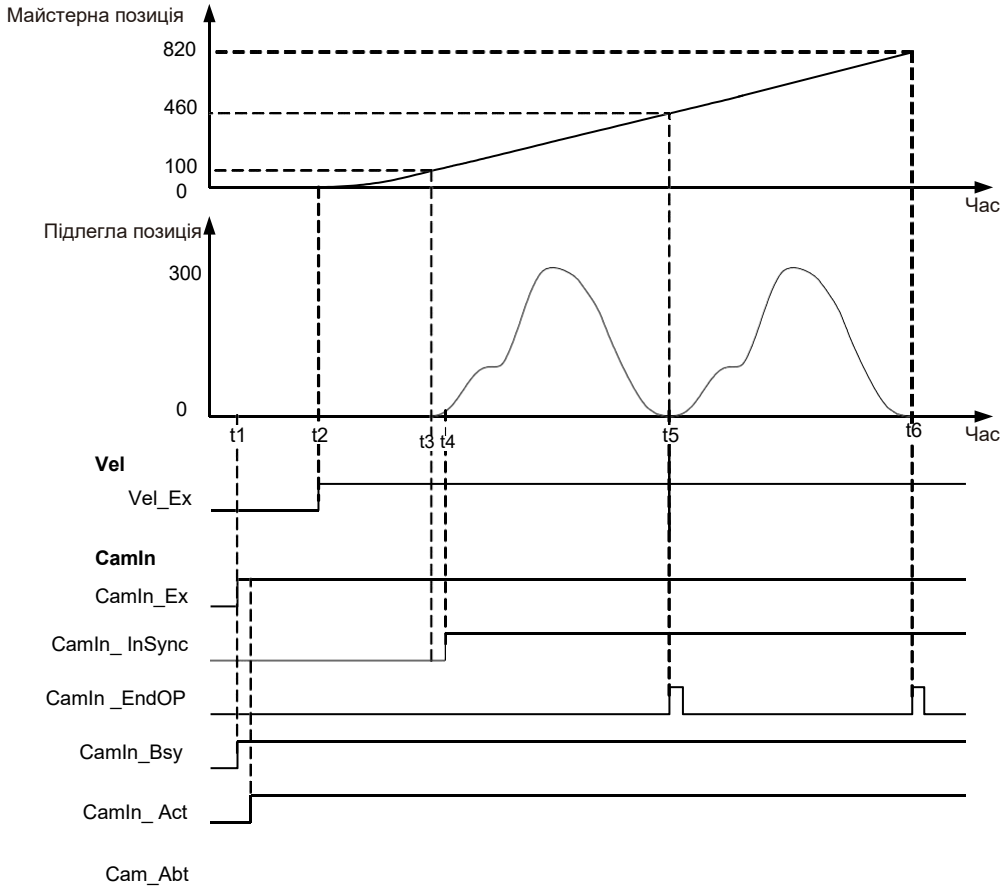
■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
CamIn	MC_CamIn	
CamIn_Ex	BOOL	
CamIn_InSync	BOOL	
CamIn_EndOP	BOOL	
CamIn_Bsy	BOOL	
CamIn_Act	BOOL	
CamIn_Abt	BOOL	
CamIn_Err	BOOL	
CamIn_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	
Vel _InVel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel_Err	BOOL	
Vel _ ErrID	WORD	



11

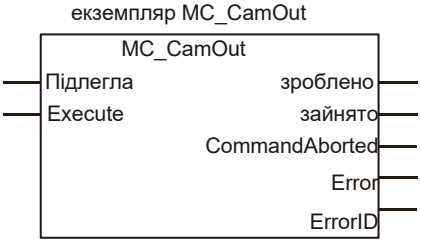
■ **Крива руху та часовий графік**



- Коли CamIn_Ex змінюється з FALSE на TRUE, інструкція MC_CamIn виконується, і в момент t1 положення головної та підпорядкованої осей дорівнюють 0. Значення *ActivationPosition* дорівнює 100, а *ActivationMode* дорівнює 0, тому підлеглий пристрій не почне виконувати дію залучення, доки положення головної осі не стане 100 (положення головної осі в момент t1 + *ActivationPosition*).
- Коли Vel_Ex змінюється з FALSE на TRUE, інструкція MC_MoveVelocity виконується, і в момент t2 позиція головної осі дорівнює 0, а ведена вісь продовжує очікувати початку залучення. Після цього головна вісь переміститься від 0 у позитивному напрямку під керуванням інструкції MC_MoveVelocity.
- Коли головна вісь проходить 100, положення, де починається зачеплення, досягається в момент t3. Ведувана вісь починає виконувати дію зачеплення відповідно до *StartMode* в момент t3. Синхронізація досягається в t4, а вихідний параметр *InSync* (CamIn1_InSync) змінюється з FALSE на TRUE.
- Кожного разу, коли синхронний рух досягає кінцевої точки в періоді кулачка, як показано на t5 і t6, вихідний параметр *EndOfProfile* (CamIn1_EndPro) зміниться на TRUE і зміниться на FALSE після періоду програми.

11.4.6 MC_CamOut

FB/FC	Пояснення	Застосовна модель
FB	MC_CamOut може від'єднати встановлений зв'язок електронної камери.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Підлеглий	Вкажіть номер підпорядкованої осі, яка має бути від'єднана від зв'язку кулачка.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-

- **Вихідні параметри**

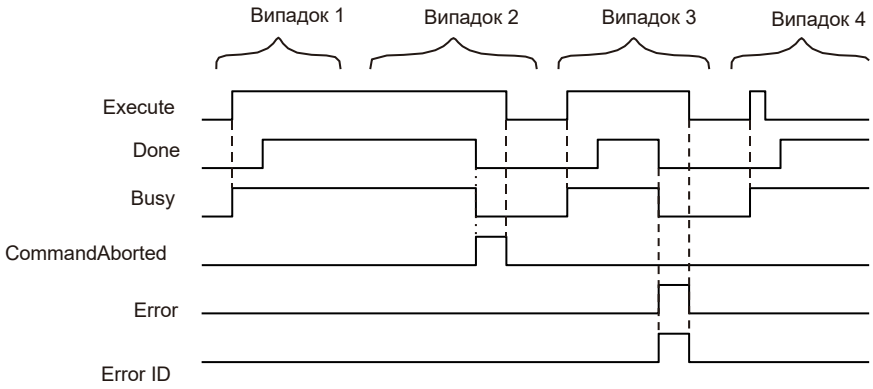
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли зв'язок електронного кулачка між веденою віссю та головною віссю роз'єднаний.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
CommandAborted	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ *CommandAborted* встановлюється на TRUE, коли інструкція переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється з FALSE на TRUE. Через крапку *«Готово»* змінюється на «TRUE». *Busy* і *Done* залишаються TRUE після того, як *Execute* зміниться з TRUE на FALSE.

Випадок 2 : коли *Execute* має значення TRUE, *CommandAborted* змінюється на TRUE, а параметри *Busy* та *Done* змінюються на FALSE, якщо інструкцію переривається іншою інструкцією. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, вісь вимкнено), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок. Тим часом *«Зайнятий»* і *«Готово»* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 4 : *Виконайте* зміни з TRUE на FALSE, оскільки виконання інструкції триває менше одного періоду. Після цього *значення Done* змінюються на TRUE і *Busy* залишаються TRUE, коли досягається один період.

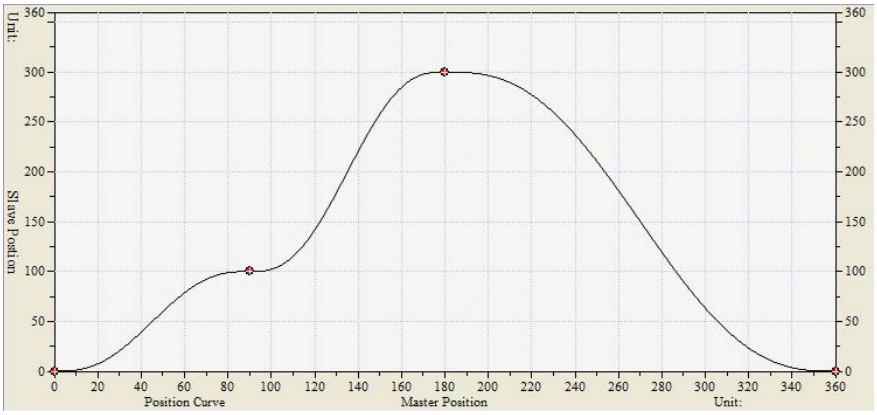
● **Функції**

MC_CamOut використовується для відключення встановленого зв'язку електронного кулачка. Інструкція працює на веденій осі в режимі роботи кулачка, і ведена вісь продовжуватиме рухатися зі швидкістю, коли вона буде від'єднана від зв'язку кулачка.

Інструкції MC_Halt або MC_Stop можуть бути виконані на веденій осі, щоб зупинити рух веденої осі. Підпорядкована вісь припинить рух і взаємозв'язок кулачка буде роз'єднано після завершення виконання інструкції MC_Halt або інструкції MC_Stop.

Приклад програмування

■ **Ефект виконання інструкції *MC_CamOut* описано в наступному прикладі. Крива кулачка планується, як показано нижче.**



■ **Ключові точки кулачкової кривої**

№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
1	0	0	0	0
2	90	100	0	0
3	180	300	0	0
4	360	0	0	0

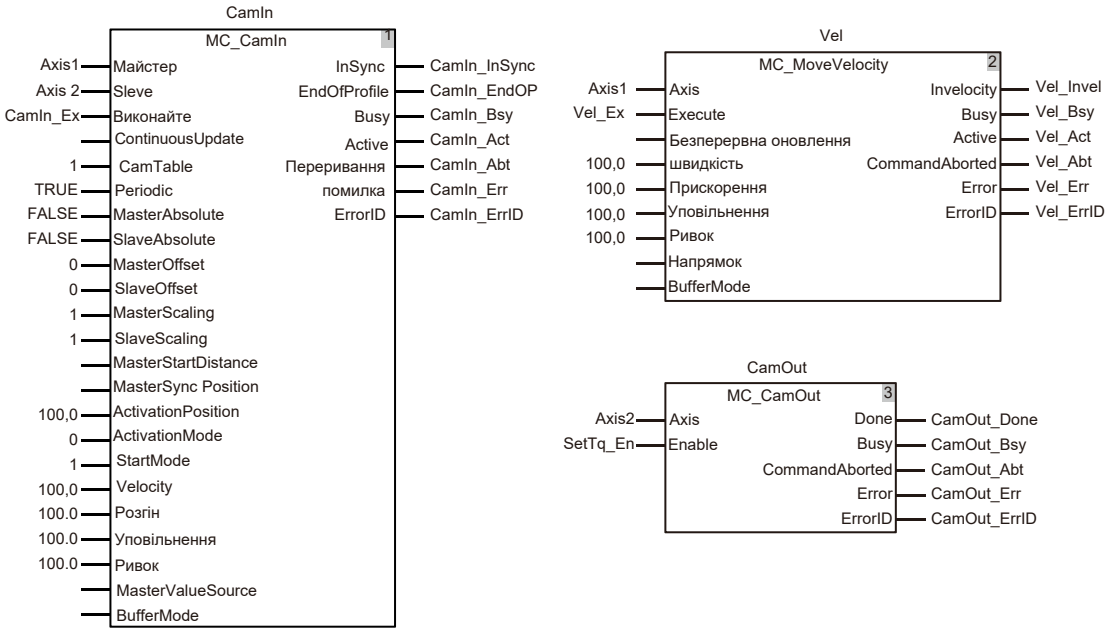
■ **Пояснення:**

Період кулачка головної осі та веденої осі	360
MasterScaling і SlaveScaling	1
MasterOffset і SlaveOffset	0
МайстерАбсолют	Відносна
SlaveAbsolute	Відносна
Періодичні	Періодичні
ActivationPosition	Відносне положення осі: 100
StartMode	Найкоротший шлях

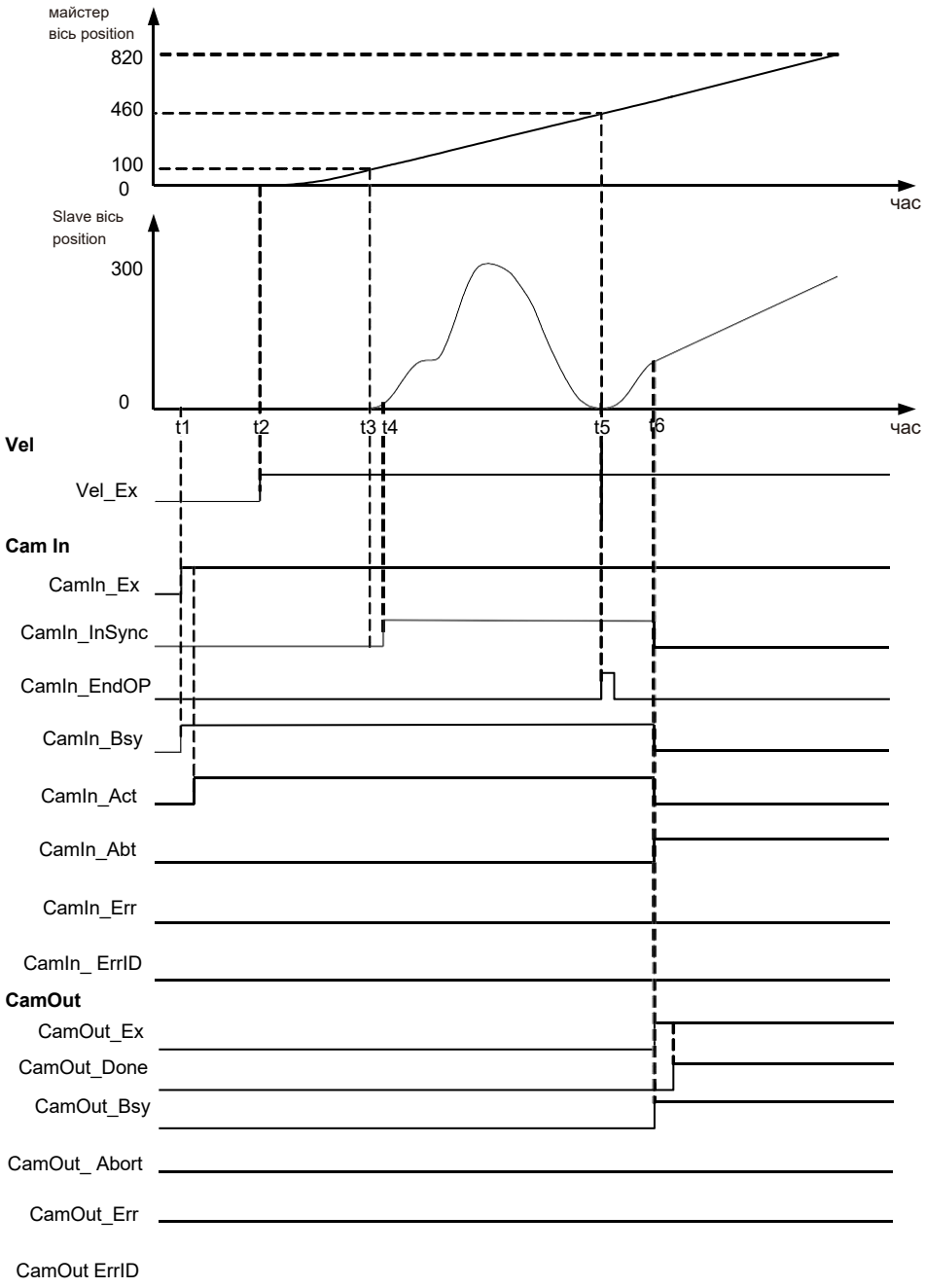
■ **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
CamIn	MC_CamIn	
CamIn_Ex	BOOL	
CamIn_InSync	BOOL	
CamIn_EndOP	BOOL	
CamIn_Bsy	BOOL	
CamIn_Act	BOOL	
CamIn_Abt	BOOL	
CamIn_Err	BOOL	
CamIn_ErrID	WORD	
Vel	MC_MoveVelocity	
Vel_Ex	BOOL	
Vel_InVel	BOOL	
Vel_Bsy	BOOL	
Vel_Act	BOOL	

Ім'я змінної	Тип даних	Початкове значення
Vel_Abt	BOOL	
Vel_Err	BOOL	
Vel_ErrID	WORD	
CamOut	MC_CamOut	
CamOut_Ex	BOOL	
CamOut_Done	BOOL	
CamOut_Bsy	BOOL	
CamOut_Abt	BOOL	
CamOut_Err	BOOL	
CamOut_ErrID	WORD	



■ Крива руху та часовий графік

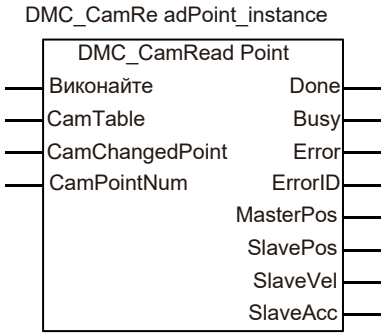


- Коли CamIn_Ex змінюється з FALSE на TRUE в t1, інструкція MC_CamIn виконується, і на даний момент положення головної осі та підпорядкованої осі дорівнюють 0. Значення *ActivationPosition* дорівнює 100, а *ActivationMode* дорівнює 0, тому підпорядкована вісь не запускається щоб виконати дію залучення, доки позиція головної осі не стане 100 (позиція головної осі в t1 + *ActivationPosition*).
- Коли Vel_Ex змінюється з FALSE на TRUE в t2, починається виконання інструкції MC_MoveVelocity. На даний момент позиція головної осі дорівнює 0, а ведена вісь продовжує очікувати на виконання дії залучення. Після цього головна вісь рухається від 0 у позитивному напрямку під керуванням інструкції MC_MoveVelocity.

- Положення, де починається зачеплення, досягається, коли головна вісь проходить 100 в t3. Підпорядкована вісь починає виконувати дію зачеплення відповідно до *StartMode* у t3. Синхронізація досягається в t4, а вихідний параметр *InSync* (CamIn1_InSync) змінюється з FALSE на TRUE.
- Під час синхронного руху кулачка, в якому ведена вісь слідує за рухом головної осі, шляхом виконання інструкції MC_CamOut взаємозв'язок кулачка роз'єднується в момент t6. Після виконання інструкції MC_CamOut ведена вісь продовжуватиме рухатися зі швидкістю, яку вона має, коли ведену вісь від'єднано від зв'язку кулачка.

11.4.7 DMC_CamReadPoint

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamReadPoint зчитує інформацію про камеру.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
CamChangedPoint	Якщо *CamChangedPoint* має значення FALSE, інструкція зчитує інформацію про точку кулачка, яка знаходиться до того, як точку кулачка було змінено. Якщо *CamChangedPoint* має значення TRUE, інструкція зчитує інформацію про точку кулачка, яка з'являється після того, як точку кулачка було змінено.	BOOL	TRUE/FALSE	Коли *Execute* змінюється з FALSE на TRUE.
CamPointNum	Номер точки кулачка, який потрібно вибрати.	UINT	1~2048 (0)	Коли *Execute* змінюється з FALSE на TRUE.

● **Вихідні параметри**

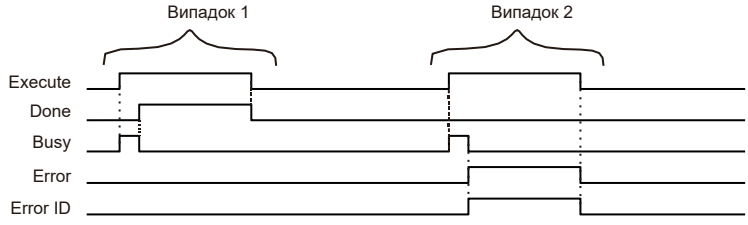
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

Назва параметра	функція	Тип даних	Допустимий діапазон
MasterPos	Положення головної осі вибраного електронного кулачка.	LREAL	0, додатне число
SlavePos	Положення веденої осі вибраного електронного кулачка.	LREAL	0, додатне число
SlaveVel	Швидкість веденої осі вибраного електронного кулачка.	LREAL	0, додатне число
SlaveAcc	Прискорення веденої осі вибраного електронного кулачка.	LREAL	0, додатне число

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

- **Функції**

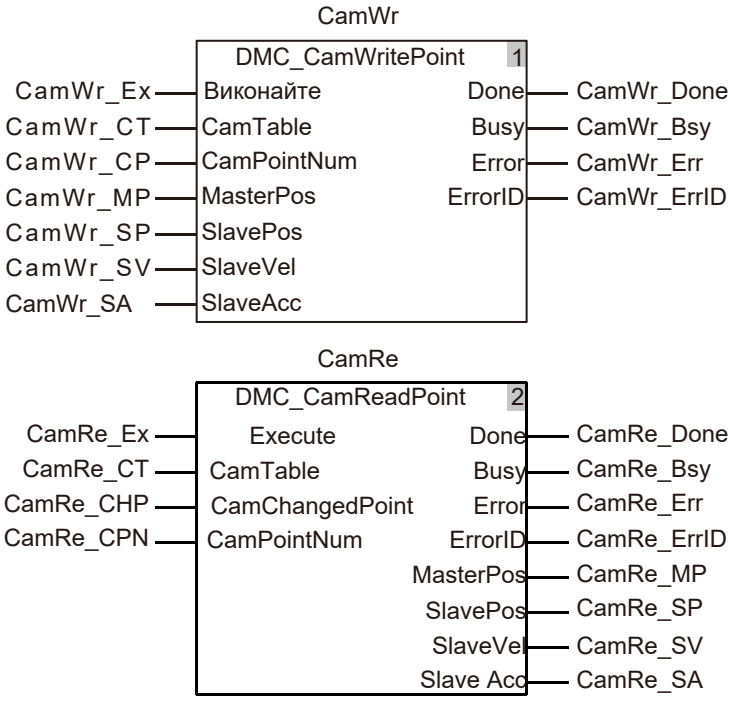
DMC_CamReadPoint зчитує інформацію про точку кулачка в електронній таблиці кулачка.
Якщо *CamChangedPoint* має значення FALSE, інструкція зчитує параметри точки кулачка, яка знаходиться до того, як інформація про точку кулачка змінюється за допомогою DMC_CamSet.
Якщо *CamChangedPoint* має значення TRUE, інструкція зчитує параметри точки кулачка, яка є після того, як інформацію про точку кулачка змінено за допомогою DMC_CamSet.

Приклад програмування

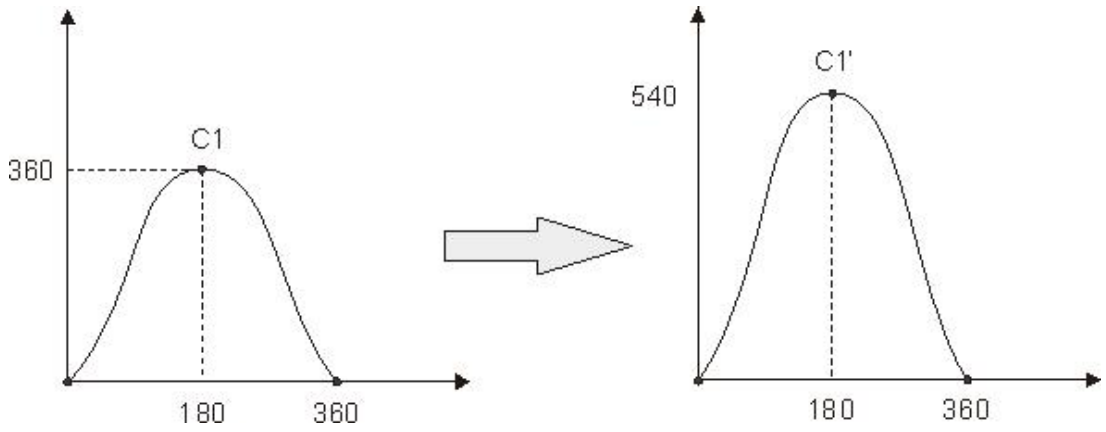
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
CamWr	DMC_CamWritePoint	
CamWr_Ex	BOOL	FALSE
CamWr_CT	USINT	1
CamWr_CP	UINT	2

Ім'я змінної	Тип даних	Початкове значення
CamWr_MP	LREAL	180,0
CamWr_SP	LREAL	540,0
CamWr_SV	LREAL	0,0
CamWr_SA	LREAL	0,0
CamWr_Done	BOOL	
CamWr_Bsy	BOOL	
CamWr_Err	BOOL	
CamWr_ErrID	WORD	
CamRe	DMC_CamReadPoint	
CamRe_Ex	BOOL	FALSE
CamRe_CT	USINT	1
CamRe_CHP	BOOL	TRUE
CamRe_CPN	UINT	2
CamRe_Done	BOOL	
CamRe_Bsy	BOOL	
CamRe_Err	BOOL	
CamRe_ErrID	WORD	
CamRe_MP	LREAL	
CamRe_SP	LREAL	
CamRe_SV	LREAL	
CamRe_SA	LREAL	



2. Кулачкова крива



Кулачкова крива C1 змінюється на C1'.

- На кривій кулачка є три точки кулачка. Коли CamWr_Ex змінюється на TRUE, виконується DMC_CamWritePoint. Коли CamWr_Done змінюється на TRUE, це означає, що запис інформації про кулачок завершено. Це друга точка кулачка, про яку записується інформація.
- Коли CamRe_Ex змінюється на TRUE, виконується DMC_CamReadPoint. Коли параметр *CamChangedPoint* (змінна: CamRe_CHP) змінюється на FALSE, інформація про кулачок, яку зчитує інструкція, є інформацією про кулачок перед записом, як показано в наведеній нижче таблиці.

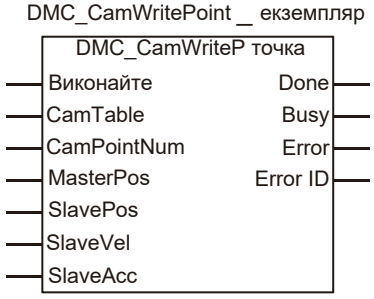
№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
2	180,0	360,0	0,0	0,0

- Коли параметр *CamChangedPoint* (змінна: CamRe_CHP) змінюється на TRUE, інформація про cam-point, яку зчитує інструкція, є інформацією про cam-point після виконання запису, як показано в наступній таблиці.

№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
2	180,0	540,0	0,0	0,0

11.4.8 DMC_CamWritePoint

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamWritePoint використовується для налаштування параметрів однієї точки кулачка.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
CamPointNum	Номер встановленої точки кулачка.	UINT	1~2048 (0)	Коли *Execute* змінюється з FALSE на TRUE.
MasterPos	Положення головної осі встановленої точки кулачка.	LREAL	0, позитивне число	
SlavePos	Положення веденої осі точки кулачка, яка встановлена.	LREAL	0, позитивне число	
SlaveVel	Швидкість підлеглої осі точки кулачка, яка встановлена.	LREAL	0, додатне число	
SlaveAcc	Прискорення веденої осі точки кулачка, яка встановлена.	LREAL	0, додатне число	

- **Вихідні параметри**

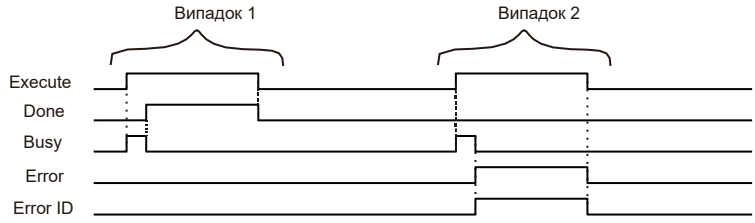
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

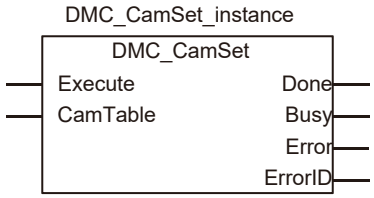
Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

- **Функції**

DMC_CamWritePoint використовується для встановлення параметрів точки кулачка в електронній таблиці кулачка. Нова крива кулачка не буде ефективною одразу після завершення налаштування, доки не буде виконано інструкцію DMC_CamSet. Зверніться до прикладу програмування в розділі 11.4.9 DMC_CamSet для прикладу використання цієї інструкції.

11.4.9 DMC_CamSet

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamSet використовується для того, щоб зробити модифіковану інформацію про точку кулачка ефективною.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

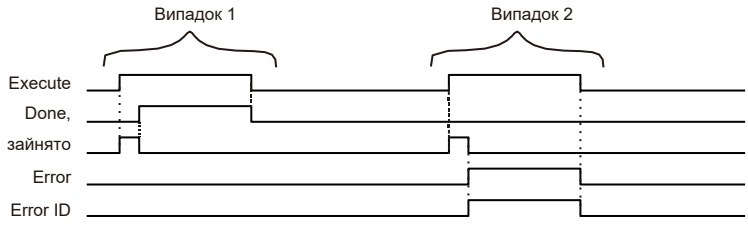
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

● Функції

DMC_CamSet використовується для ефективної інформації про встановлену точку кулачка. Спочатку використовуйте інструкцію DMC_CamWritePoint, щоб установити відповідну інформацію про кулачок в електронній таблиці кулачка. Потім виконайте інструкцію DMC_CamSet, щоб зробити нову інформацію про точку кулачка ефективною.

Якщо інструкція DMC_CamSet виконується після виконання інструкції MC_CamIn, крива кулачка після модифікації буде ефективною в наступному циклі.

Якщо інструкція DMC_CamSet виконується перед виконанням інструкції MC_CamIn, змінена крива кулачка набуде чинності негайно.

● Запобіжні заходи

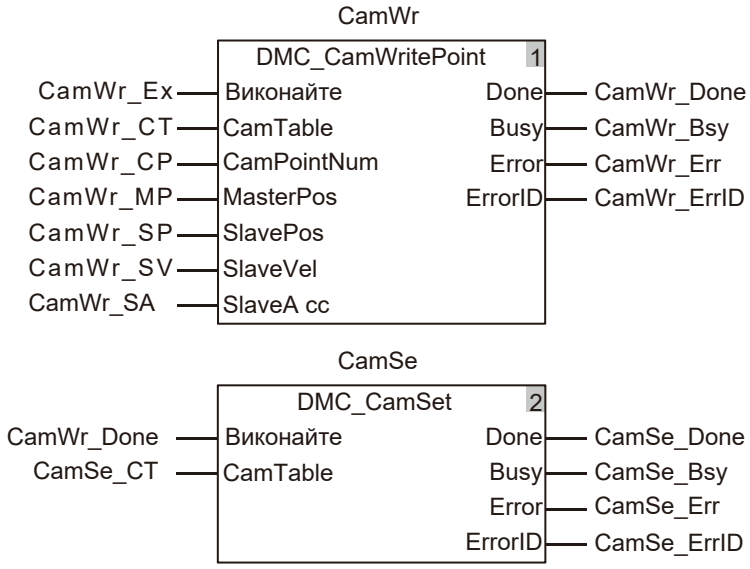
Якщо DMC_CamSet використовується для ефективної модифікованої кривої кулачка, переконайтеся, що крива кулачка викликається для використання щонайбільше однією інструкцією MC_CamIn. Якщо кулачкова крива викликається для використання декількома інструкціями MC_CamIn, час для введення в дію встановленої кулачкової кривої не є впевненим після виконання DMC_CamSet.

Приклад програмування

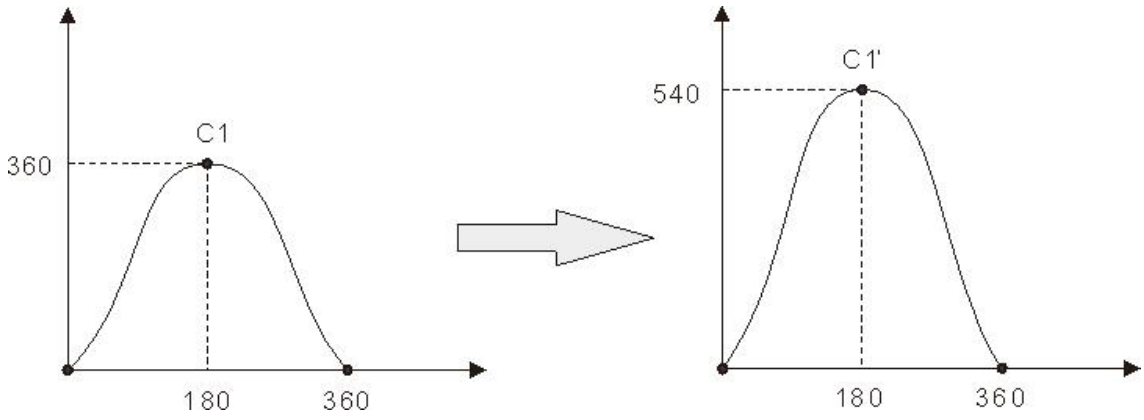
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
CamWr	DMC_CamWritePoint	
CamWr_Ex	BOOL	FALSE
CamWr_CT	USINT	1
CamWr_CP	UINT	2
CamWr_MP	LREAL	180,0
CamWr_SP	LREAL	540,0
CamWr_SV	LREAL	0,0
CamWr_SA	LREAL	0,0
CamWr_Done	BOOL	
CamWr_Bsy	BOOL	
CamWr_Err	BOOL	
CamWr_ErrID	WORD	
CamSe	DMC_CamSet	
CamSe_CT	USINT	1
CamSe_Done	BOOL	

Ім'я змінної	Тип даних	Початкове значення
CamSe_Bsy	BOOL	
CamSe_Err	BOOL	
CamSe_ErrID	WORD	



2. Кулачкова крива



На кривій кулачка є три точки кулачка. Змініть криву C1 на криву C1'.

❖ Інформація про камеру перед зміною:

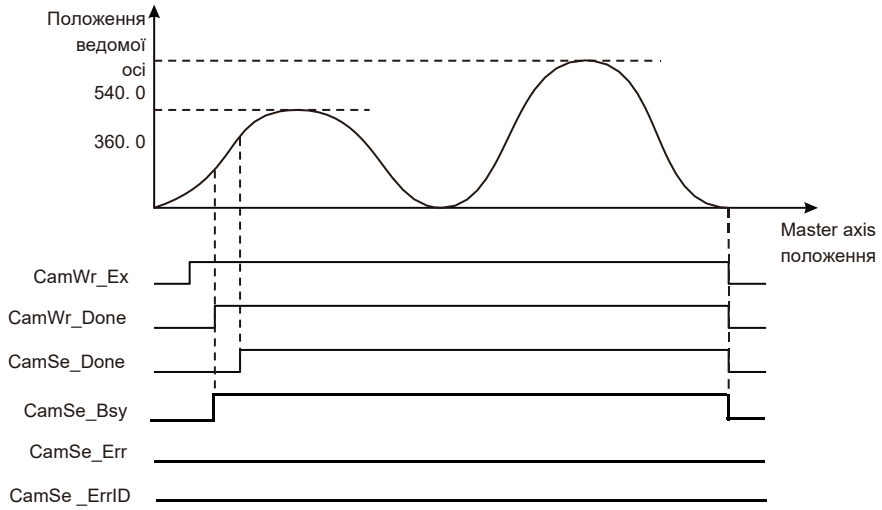
№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
1	0,0	0,0	0,0	0,0
2	180,0	360,0	0,0	0,0
3	360,0	0,0	0,0	0,0

На кривій вище видно, що друга точка кулачка змінена. Крива кулачка змінюється шляхом виконання спочатку DMC_CamWritePoint, а потім виконання DMC_CamSet у програмі вище.

❖ Інформація про камеру після модифікації:

№	Положення головної осі	Положення веденої осі	швидкість	Прискорення
1	0,0	0,0	0,0	0,0
2	180,0	540,0	0,0	0,0
3	360,0	0,0	0,0	0,0

3. Діаграма послідовності:

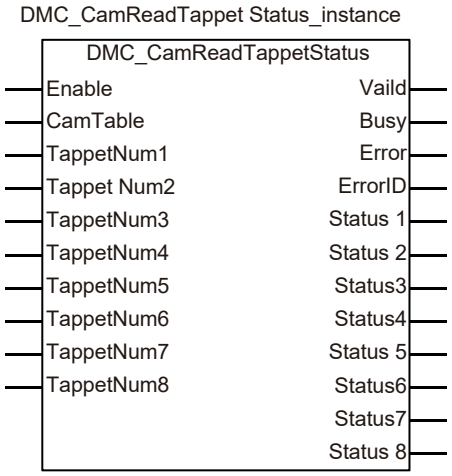


◆ Коли CamWr_Ex змінюється з FALSE на TRUE, DMC_CamWritePoint виконується. Коли CamWr_Done змінюється на TRUE, налаштування інформації про точку кулачка завершено.

◆ Коли CamWr_Done змінюється з FALSE на TRUE, DMC_CamSet виконується. Коли CamSe_Done змінюється на TRUE, виконання DMC_CamSet завершено. Крива кулачка після модифікації набуде чинності в наступному циклі після завершення поточного циклу кулачка.

11.4.10 DMC_CamReadTappetStatus

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamReadTappetStatus використовується для зчитування стану кількох точок штовхача.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли *Enable* має значення TRUE.
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.
TappetNum1	Номер першої точки штовхача	UINT	1~128 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
TappetNum2	Номер другої точки штовхача	UINT	1~128 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.
TappetNum3	Номер третьої точки штовхача	UINT	1~128 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
TappetNum4	Номер четвертої точки штовхача	UINT	1~128 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.
TappetNum5	Номер п'ятої точки штовхача.	UINT	1~128 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
TappetNum6	Номер шостої точки натискання	UINT	1~128 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
TappetNum7	Номер сьомої точки штовхача.	UINT	1~128 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.
TappetNum8	Номер восьмої точки штовхача.	UINT	1~128 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.

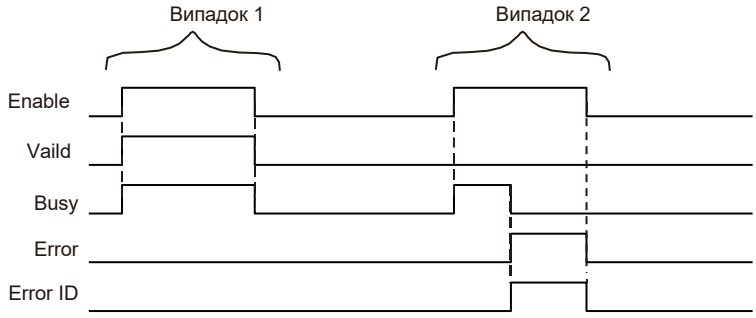
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Status1	Статус першої точки штовхання.	BOOL	TRUE/FALSE
Status2	Стан другої точки штовхача.	BOOL	TRUE/FALSE
Status3	Стан третьої точки штовхання.	BOOL	TRUE/FALSE
Status4	Стан четвертої точки штовхання.	BOOL	TRUE/FALSE
Status5	Стан п'ятої точки натискання.	BOOL	TRUE/FALSE
Status6	Стан шостої точки натискання.	BOOL	TRUE/FALSE
Status7	Стан сьомої точки натискання.	BOOL	TRUE/FALSE
Status8	Стан восьмої точки натискання.	BOOL	TRUE/FALSE

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли виконання інструкції завершено.	◆ Коли *Enable* змінюється на FALSE.
Busy	◆ Коли *Enable* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Enable* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і Busy змінюються на TRUE. Коли *Enable* змінюється на FALSE, *Valid* і *Busy* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Enable* змінюється з TRUE на FALSE, а потім значення *ErrorID* очищається до 0.

- **Функції**

DMC_CamReadTappetStatus використовується для зчитування стану восьми точок штовхача.

Інструкція DMC_CamReadTappetStatus виконується для зчитування стану точок штовхача, коли головна вісь проходить через точки штовхача в позитивному або негативному напрямку. Статус кожної точки штовхача визначається налаштуванням кожної точки штовхача. Статус кожної точки штовхача зміниться на FALSE, коли головна вісь проходить кінцеву точку кулачка в позитивному напрямку або коли головна вісь проходить початкову точку кулачка в негативному напрямку.

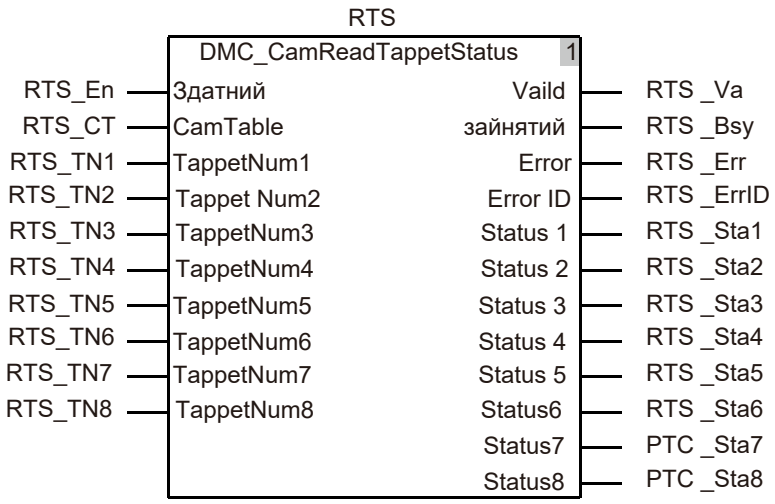
Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
RTS	DMC_CamReadTappetStatus	
RTS_En	BOOL	FALSE
RTS_CT	USINT	1
RTS_TN1	UINT	1
RTS_TN2	UINT	2
RTS_TN3	UINT	3
RTS_TN4	UINT	4
RTS_TN5	UINT	5
RTS_TN6	UINT	6
RTS_TN7	UINT	7
RTS_TN8	UINT	8
RTS_Va	BOOL	
RTS_Bsy	BOOL	
RTS_Err	BOOL	
RTS_ErrID	WORD	
RTS_Sta1	BOOL	
RTS_Sta2	BOOL	
RTS_Sta3	BOOL	
RTS_Sta4	BOOL	
RTS_Sta5	BOOL	

Ім'я змінної	Тип даних	Початкове значення
RTS_Sta6	BOOL	
RTS_Sta7	BOOL	
RTS_Sta8	BOOL	

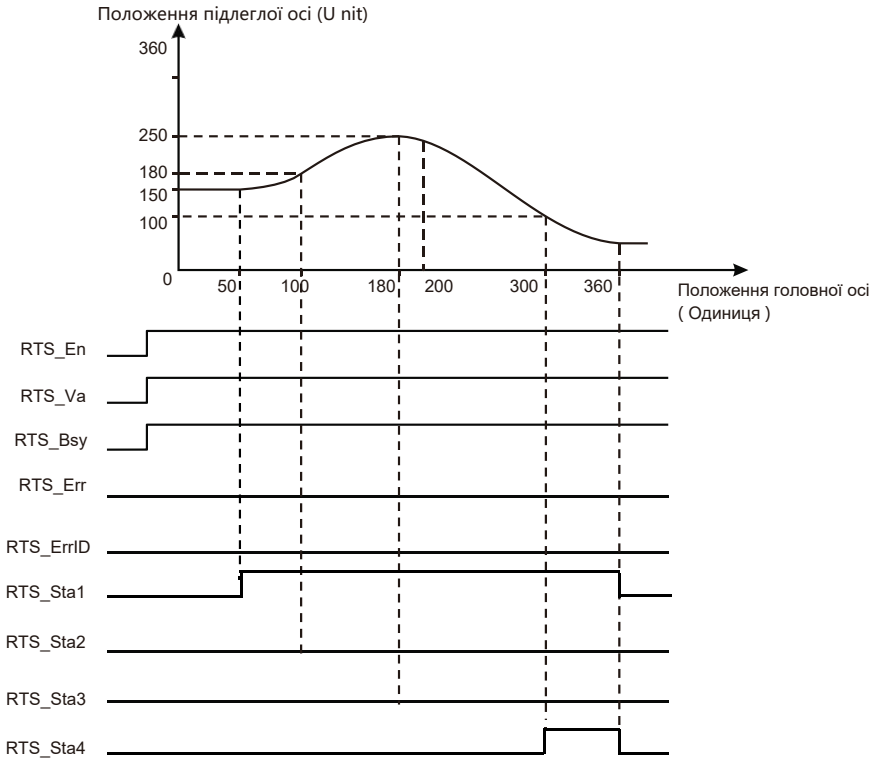
11



1. Таблиця точок штовхача

Індекс	Положення головної осі	Положення веденої осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
1	50,0	150,0	PositiveOn	NegativeOff
2	100,0	180,0	PositiveDisable	NegativeOff
3	180,0	250,0	PositiveOff	NegativeOff
4	300,0	100,0	PositiveInvert	NegativeOff

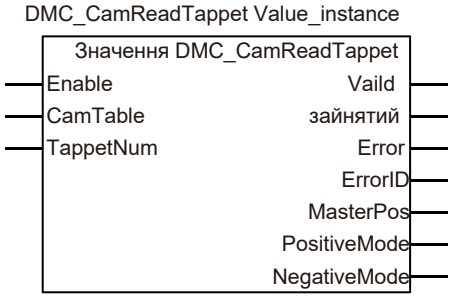
2. Крива положення та діаграма послідовності



- Додайте точки штовхача до встановленої кривої кулачка, як показано на малюнку вище. Точка натискання 1 встановлюється на PositiveOn, коли вона передається в позитивному напрямку. Точка натискання 2 встановлюється на PositiveDisable, коли її передають у позитивному напрямку. Точка штовхача 3 встановлюється на PositiveOff, коли вона передається в позитивному напрямку. Точка штовхача 4 встановлюється на PositiveInvert, коли її передають у позитивному напрямку.
- Коли вісь рухається вперед і RTS_En змінюється з FALSE на TRUE, виконується DMC_CamReadTappetStaus. (Оскільки перша точка штовхача вибирає PositiveOn,) RTS_Sta1 змінюється з FALSE на TRUE, коли вісь проходить першу точку штовхача на кривій кулачка. (Оскільки друга точка натискання вибирає PositiveDisable, RTS_Sta2 має значення FALSE, коли другу точку натискання пройдено. (Оскільки третя точка штовхання вибирає PositiveOff, RTS_Sta3 має значення FALSE, коли третя точка штовхання пройдена. (Оскільки четверта точка перемикання вибирає PositiveInvert, RTS_Sta4 змінюється з FALSE на TRUE, коли четверта точка перемикання пройдена.
- RTS_Sta1 і RTS_Sta4 змінюються на FALSE, коли головна вісь проходить кінцеву точку кривої кулачка.

11.4.11 DMC_CamReadTappetValue

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamReadTappetValue читає параметри однієї точки штовхання.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	Коли Enable має значення TRUE.
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Enable* має значення TRUE.
TappetNum	Номер точки штовхача для зчитування	UINT	1~128 (Потрібно встановити значення змінної)	Коли Enable має значення TRUE.

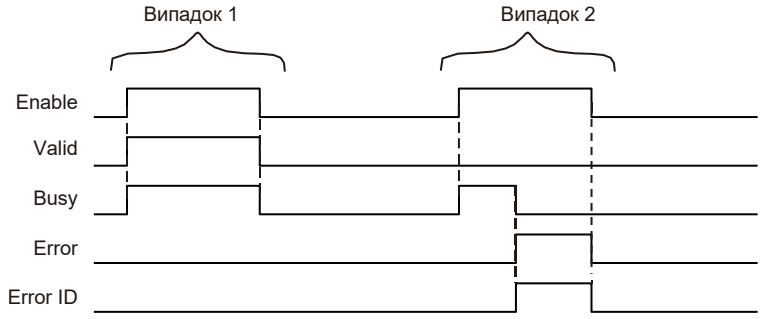
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
MasterPos	Відображає положення головної осі.	LREAL	
Позитивний режим	Вибір режиму стану для точки штовхання, коли вісь проходить через точку штовхання в позитивному напрямку.	PositiveMode_Type	0: позитивне відключення 1: Позитивний 2: Позитив Вимкнено 3: Позитивне інвертування
NegativeMode	Вибір режиму стану для точки штовхання, коли вісь проходить через точку штовхання в негативному напрямку.	NegativeMode_Type	0: негативне відключення 1: Негативний 2: Негатив Вимкнено 3: негативний інверт

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли виконання інструкції завершено.	◆ Коли *Enable* змінюється на FALSE.
Busy	◆ Коли *Enable* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Enable* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE. Коли *Enable* змінюється на FALSE, *Valid* і *Busy* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Enable* змінюється з TRUE на FALSE, а потім значення *ErrorID* очищається до 0.

● **Функції**

Інформація про точку штовхання містить положення головної осі та режими, коли точка штовхання переміщається в позитивному та негативному напрямку. Коли вісь рухається вперед, точка штовхача може вибрати PositiveDisable, PositiveOn, PositiveOff або PositiveInvert. Коли вісь рухається в зворотному напрямку, точка штовхача може вибрати NegativeDisable, NegativeOn, NegativeOff або NegativeInvert. Значення режимів наведено в наступній таблиці.

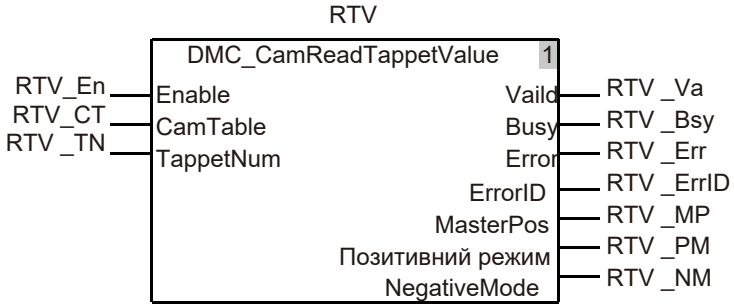
Режим	функція	Пояснення
PositiveDisable	Вимкнено	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, не змінюється.
PositiveOn	ON	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, є ON.
PositiveOff	OFF	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, OFF.
PositiveInvert	Інвертувати	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде в позитивному напрямку. В іншому випадку стан точки штовхання, яка зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхання пройде в позитивному напрямку.
NegativeDisable	Вимкнено	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, не змінюється.
NegativeOn	ON	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, є ON.
NegativeOff	OFF	Коли головна вісь проходить точку штовхача у негативному напрямку, стан точки штовхання, яка зчитується, OFF.

Режим	функція	Пояснення
NegativeInvert	Інвертувати	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде у негативному напрямку. В іншому випадку стан точки штовхання, яка зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхача пройде в негативному напрямку.

Приклад програмування

1. **Таблиця змінних і програма**

Ім'я змінної	Тип даних	Початкове значення
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	2
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. Таблиця точок штовхача

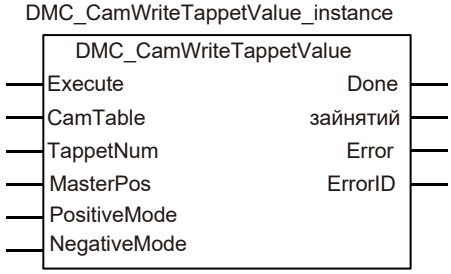
Індекс	Положення головної осі	Положення веденої осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
1	108,0	235,0	PositiveOn	NegativeInvert
2	200,0	250,0	PositiveOff	NegativeOff
3	300,0	192,0	PositiveInvert	NegativeOn

❖ Коли RTV_En змінюється з FALSE на TRUE, виконується DMC_CamReadTappetValue. Коли RTV_Va змінюється на TRUE, виконання інструкції завершується, і інформація про точку штовхача зчитується наступним чином.

Індекс	Положення головної осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
2	200,0	PositiveOff	NegativeOff

11.4.12 DMC_CamWriteTappetValue

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamWriteTappetValue використовується для встановлення параметрів для точки штовхання.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
TappetNum	Номер точки штовхача	UINT	1~128 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
MasterPos	Головне положення точки штовхача	LREAL	0, додатне число	Коли *Execute* змінюється з FALSE на TRUE.
Позитивний режим	Режим точки штовхача, коли вісь рухається вперед і проходить її.	Позитивний режим _Тип	0 : Позитивне відключення 1 : Позитивний 2 : Позитивне вимкнено 3 : Позитивне інвертування	Коли *Execute* змінюється з FALSE на TRUE.
NegativeMode	Режим точки штовхача, коли вісь рухається назад і проходить її.	NegativeMode _Тип	0 : Негативне відключення 1 : Негативний увімк. 2 : Негативний вимкнений 3 : Негативний інверт	Коли *Execute* змінюється з FALSE на TRUE.

- **Вихідні параметри**

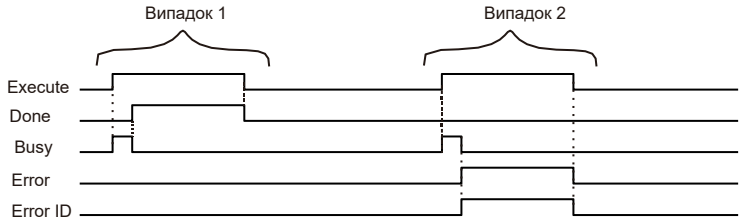
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* скидається на 0.

- **Функції**

DMC_CamWriteTappetValue використовується для встановлення параметрів для точки штовхання, включаючи положення головної осі та режим статусу точки штовхання, коли точка штовхання проходить у позитивному та негативному напрямку.
Як правило, налаштування точки штовхання виконується в кулачку, створеному на основі програмного забезпечення. Для динамічної зміни налаштування точки штовхання використовуйте інструкцію DMC_CamWriteTappetValue.
Після завершення налаштування точки штовхання використовуйте DMC_CamReadTappetStatus, щоб прочитати статус точки штовхання, коли головна вісь проходить через точку штовхання.
Коли вісь рухається вперед, точка штовхача може вибрати такий режим, як PositiveDisable, PositiveOn, PositiveOff або PositiveInvert. Коли вісь рухається назад, точка штовхача може вибрати такий режим, як NegativeDisable, NegativeOn, NegativeOff або NegativeInvert.

Значення режимів наведено в наступній таблиці.

Режим	функція	Пояснення
PositiveDisable	Вимкнено	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, не змінюється.
PositiveOn	ON	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, є ON.
PositiveOff	OFF	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, OFF.
PositiveInvert	Інвертувати	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде в позитивному напрямку. В іншому випадку стан точки штовхача який зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхання пройде в позитивному напрямку.

Режим	функція	Пояснення
NegativeDisable	Вимкнено	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, не змінюється.
NegativeOn	ON	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, є ON.
NegativeOff	OFF	Коли головна вісь проходить точку штовхача у негативному напрямку, стан точки штовхання, яка зчитується, OFF.
NegativeIvert	Інвертувати	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде у негативному напрямку. В іншому випадку стан точки штовхача який зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхання пройде в негативному напрямку.

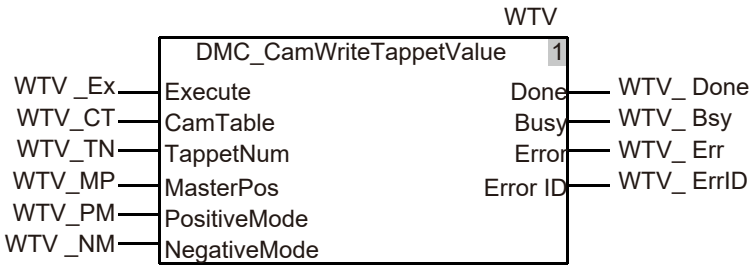
- **Запобіжні заходи**

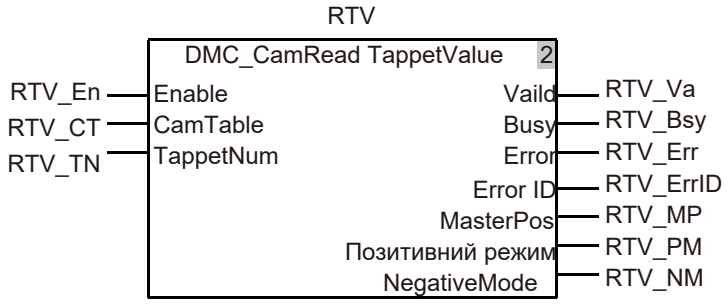
  Переконайтеся, що крива кулачка викликається для використання щонайбільше однією інструкцією MC_CamIn, якщо DMC_CamWriteTappetValue використовується для встановлення параметрів для однієї точки штовхача в кривій кулачка.
  Якщо криву кулачка викликається декількома інструкціями MC_CamIn, точку штовхання, яку потрібно змінити, яка використовується в програмах, буде змінено, коли DMC_CamWriteTappetValue використовується для зміни параметрів однієї точки штовхача.

Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
WTV	DMC_CamWriteTappetValue	
WTV_Ex	BOOL	FALSE
WTV_CT	USINT	1
WTV_TN	UINT	2
WTV_MP	LREAL	200,0
WTV_PM	LREAL	PositiveOff
WTV_NM	LREAL	NegativeOff
WTV_Done	BOOL	
WTV_Bsy	BOOL	
WTV_Err	BOOL	
WTV_ErrID	WORD	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	2
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	





2. Таблиця точок штовхача

Індекс	Положення головної осі	Положення веденої осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
1	108,0	235,0	ON	Інвертувати
2	200,0	250,0	OFF	OFF
3	300,0	192,0	Інвертувати	ON

❖ Коли WTV_Ex змінюється з FALSE на TRUE, виконується DMC_CamWriteTappetValue. Коли WTV_Done змінюється на TRUE, виконання інструкції завершується, а інформація про другу точку штовхання записується, як показано нижче.

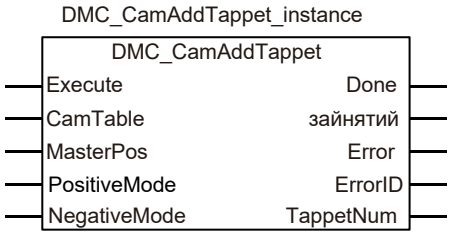
Індекс	Положення головної осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
2	250,0	PositiveOff	NegativeOff

❖ Коли RTV_En змінюється з FALSE на TRUE, виконується DMC_CamReadTappetValue. Коли RTV_Done змінюється на TRUE, виконання інструкції завершується, і інформація про одну точку штовхача, яка зчитується, відображається, як показано нижче.

Індекс	Положення головної осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
2	250,0	PositiveOff	NegativeOff

11.4.13 DMC_CamAddTappet

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamAddTappet використовується для додавання точки штовхання.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
MasterPos	Положення головної осі точки штовхача	LREAL	0, додатне число	Коли *Execute* змінюється з FALSE на TRUE.
Позитивний режим	Режим стану точки штовхача, коли вісь рухається вперед.	PositiveMode _Type	0: позитивне відключення 1: Позитивний 2: Позитив Вимкнено 3: Позитивне інвертування	Коли *Execute* змінюється з FALSE на TRUE.
NegativeMode	Режим стану точки штовхача, коли вісь рухається назад.	NegativeMode _Type	0: негативне відключення 1: Негативний 2: Негатив Вимкнено 3: негативний інверт	Коли *Execute* змінюється з FALSE на TRUE.

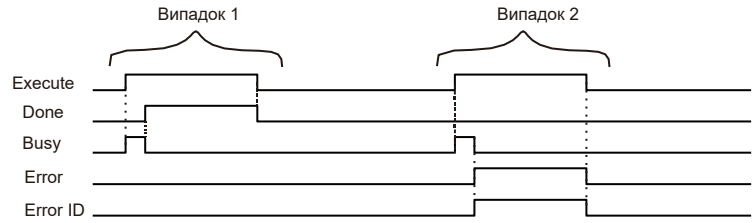
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
TappetNum	Виводить номер доданої точки штовхача.	UINT	1~128

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

● **Функції**

DMC_CamAddTappet використовується для додавання точки штовхання шляхом встановлення положення головної осі та режиму стану точки штовхання, коли точка штовхання проходить у позитивному та негативному напрямку. Кількість доданих точок штовхання накопичується на основі кількості наявних точок штовхання. Наприклад, якщо раніше найбільша кількість точок штовхача становила 3, то після виконання інструкції найбільша кількість точок штовхача дорівнює 4.
Коли вісь рухається вперед, додана точка штовхача може вибрати такий режим, як PositiveDisable, PositiveOn, PositiveOff або PositiveInvert. Коли вісь рухається назад, додана точка штовхача може вибрати такий режим, як NegativeDisable, NegativeOn, NegativeOff або NegativeInvert.
Значення режимів наведено в наступній таблиці.

Режим	функція	Пояснення
PositiveDisable	Вимкнено	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, не змінюється.
PositiveOn	ON	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, є ON.
PositiveOff	OFF	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, OFF.
PositiveInvert	Інвертувати	Коли головна вісь проходить через точку штовхання в позитивному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде в позитивному напрямку. В іншому випадку стан точки штовхання, яка зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхання пройде в позитивному напрямку.
NegativeDisable	Вимкнено	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, не змінюється.
NegativeOn	ON	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, є ON.
NegativeOff	OFF	Коли головна вісь проходить точку штовхача у негативному напрямку, стан точки штовхання, яка зчитується, OFF.

Режим	функція	Пояснення
NegativeInvert	Інвертувати	Коли головна вісь проходить через точку штовхання у негативному напрямку, стан точки штовхання, яка зчитується, буде OFF, якщо стан точки штовхання є УВІМКНЕНИМ до того, як точка штовхання пройде у негативному напрямку. В іншому випадку стан точки штовхання, яка зчитується, буде ON, якщо стан точки штовхання OFF до того, як точка штовхання пройде в негативному напрямку.

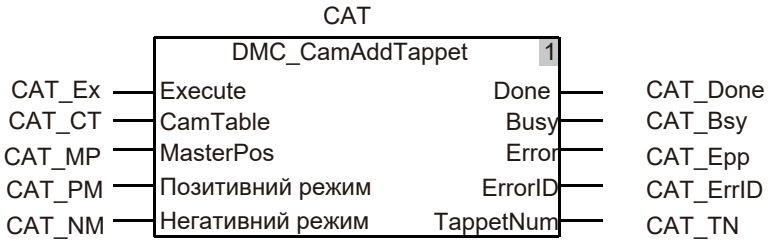
● **Запобіжні заходи**

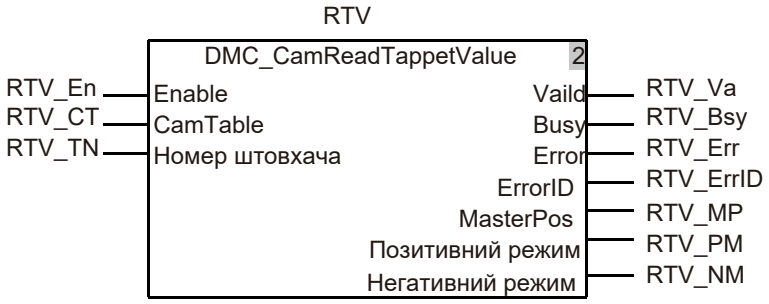
Переконайтеся, що крива кулачка викликається для використання щонайбільше однією інструкцією MC_CamIn, якщо DMC_CamAddTappet використовується для додавання однієї точки штовхача до кривої кулачка.
Якщо крива кулачка викликається декількома інструкціями MC_CamIn, додану точку штовхача можна використовувати в програмах, коли DMC_CamAddTappet використовується для додавання однієї точки штовхача.

Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
CAT	DMC_CamAddTappet	
CAT_Ex	BOOL	FALSE
CAT_CT	USINT	1
CAT_MP	LREAL	200,0
CAT_PM	LREAL	PositiveOff
CAT_NM	LREAL	NegativeOff
CAT_Done	BOOL	
CAT_Bsy	BOOL	
CAT_Err	BOOL	
CAT_ErrID	WORD	
CAT_TN	UINT	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	4
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	





2. Таблиця точок штовхача

Індекс	Положення головної осі	Положення веденої осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
1	108,0	235,0	ON	Інвертувати
2	200,0	250,0	OFF	OFF
3	300,0	192,0	Інвертувати	ON

Є три точки штовхача, які були додані до встановленої кривої кулачка. Тепер додайте четверту точку штовхання за допомогою інструкції DMC_CamAddTappet.

❖ Коли CAT_Ex змінюється з FALSE на TRUE, DMC_CamAddTappet виконується. Коли CAT_Done змінюється на TRUE, виконання інструкції завершується, а інформація про четверту точку штовхання, яка додається, відображається, як показано нижче.

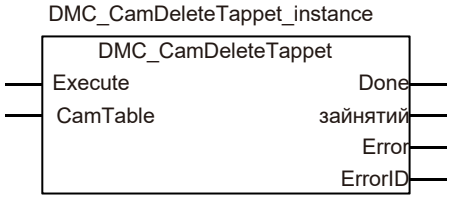
Індекс	Положення головної осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
4	250,0	PositiveOff	NegativeOff

❖ Коли RTV_En змінюється з FALSE на TRUE, виконується DMC_CamReadTappetValue. Коли RTV_Done змінюється на TRUE, виконання інструкції завершено, і інформація про точку штовхача, яка зчитується, відображається, як показано нижче.

Індекс	Положення головної осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
4	250,0	PositiveOff	NegativeOff

11.4.14 DMC_CamDeleteTappet

FB/FC	Пояснення	Застосовна модель
FB	DMC_CamDeleteTappet використовується для видалення точки натискання.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	-
CamTable	Номер таблиці кулачків, на основі якого будується взаємозв'язок кулачка між головною віссю та веденою віссю.	USINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

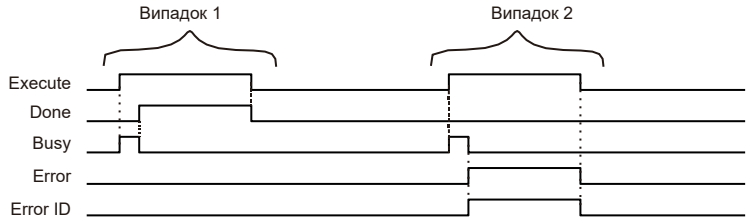
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка у виконанні інструкції.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання інструкції завершено.	◆ Коли *Execute* змінюється на FALSE.
Busy	◆ Коли *Execute* має значення TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли будь-який із вхідних параметрів для інструкції є недопустимим.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● Діаграма часу оновлення вихідних даних



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли виконання інструкції завершено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE і значення *ErrorID* скидається на 0.

● Функції

DMC_CamDeleteTappet використовується для видалення точки натискання, яка є точкою натискання з найбільшим числом. Щоразу, коли інструкція виконується один раз, одна точка натискання з найбільшої кількості буде видалена.

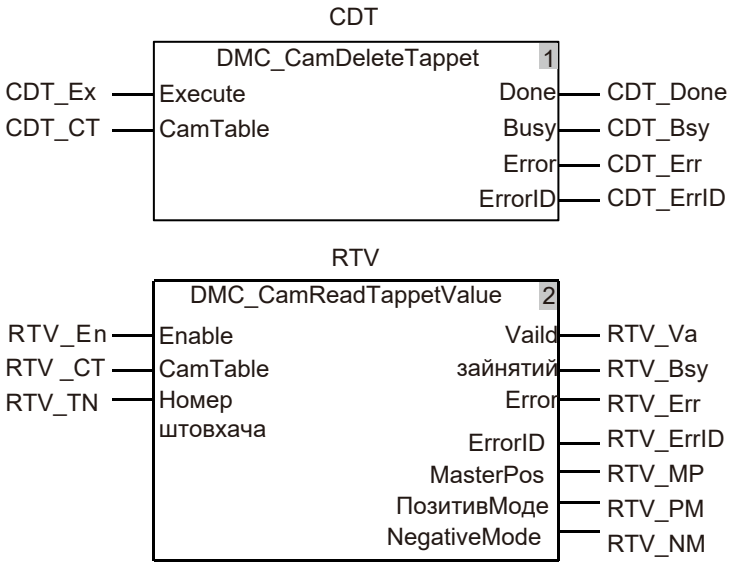
● Запобіжні заходи

Переконайтеся, що крива кулачка викликається для використання щонайбільше однією інструкцією MC_CamIn, якщо DMC_CamDeleteTappet використовується для видалення однієї точки штовхача на кривій кулачка. Якщо криву кулачка викликається декількома інструкціями MC_CamIn, точку штовхання, яку потрібно видалити, яка використовується в програмах, буде видалено, коли DMC_CamDeleteTappet використовується для видалення однієї точки штовхання.

Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
CDT	DMC_CamDeletTappet	
CDT_Ex	BOOL	FALSE
CDT_CT	USINT	1
CDT_Done	BOOL	
CDT_Bsy	BOOL	
CDT_Err	BOOL	
CDT_ErrID	WORD	
RTV	DMC_CamReadTappetValue	
RTV_En	BOOL	FALSE
RTV_CT	USINT	1
RTV_TN	UINT	3
RTV_Va	BOOL	
RTV_Bsy	BOOL	
RTV_Err	BOOL	
RTV_ErrID	WORD	
RTV_MP	LREAL	
RTV_PM	PositiveMode_Type	
RTV_NM	NegativeMode_Type	



2. Таблиця точок штовхача

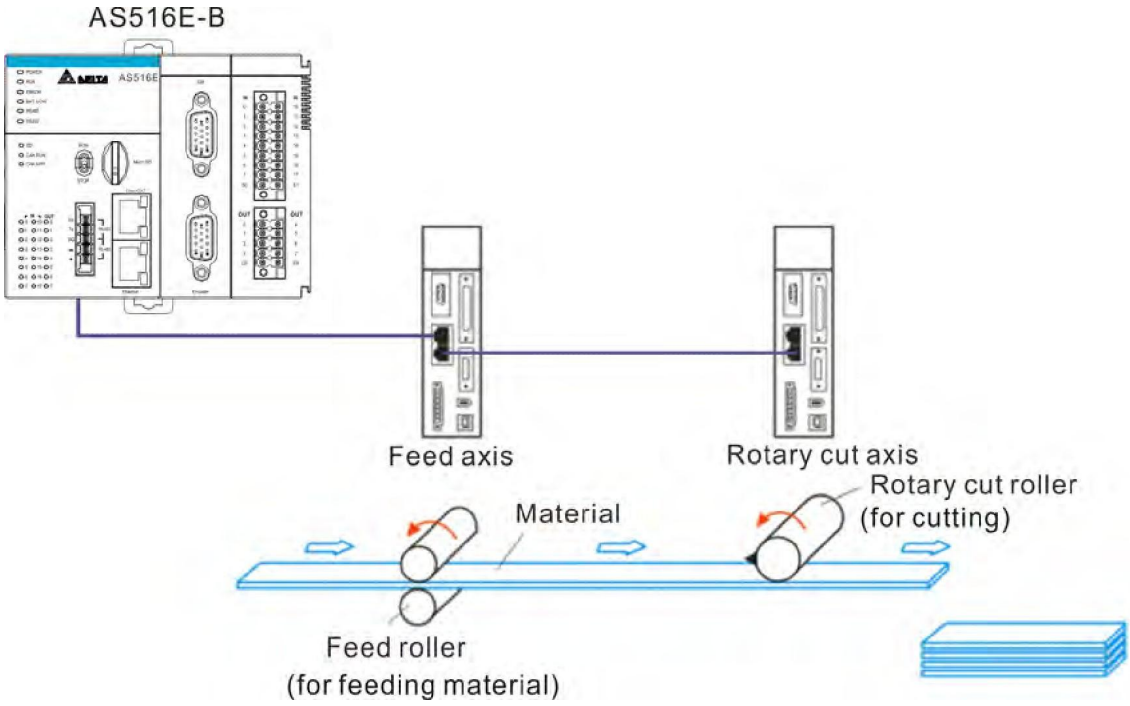
Індекс	Положення головної осі	Положення веденої осі	Пройшло в позитивному напрямку	Пройшло в негативному напрямку
1	108,0	235,0	PositiveOn	NegativeInvert
2	200,0	250,0	PositiveOff	NegativeOff
3	300,0	192,0	PositiveInvert	NegativeOn

- ❖ Коли CDT_Ex змінюється з FALSE на TRUE, DMC_CamDeletTappet виконується. Коли CDT_Done змінюється на TRUE, виконання інструкції завершується, а остання точка натискання видаляється.
- ❖ Коли RTV_En змінюється з FALSE на TRUE, виконується DMC_CamReadTappetValue. Оскільки третя точка штовхача не існує, інструкція сповіщає код помилки 16#5505 (вказує на відсутність такої точки штовхання).

11.5 Інструкція із застосування

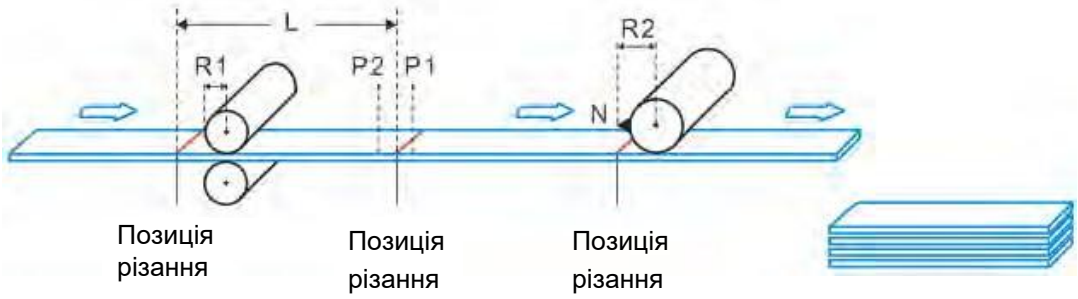
11.5.1 Технологія ротаційного різання

Ротаційне різання — це технологія вертикального різання матеріалу при передачі. Ніж проводить різання по поверхні різу періодично з обертанням поворотної осі різу.



Примітка:
Вісь подачі призначена для керування роликом подачі; поворотна вісь різання призначена для керування обертовим ріжучим роликом за допомогою ножа, встановленого на обертовому ріжучому валику. Функція ротаційного різання зазвичай використовується для різання тонкого матеріалу або матеріалу середньої тонкості та може застосовуватися в пакувальних машинах, різальних машинах, перфораційних машинах, друкарських машинах тощо.

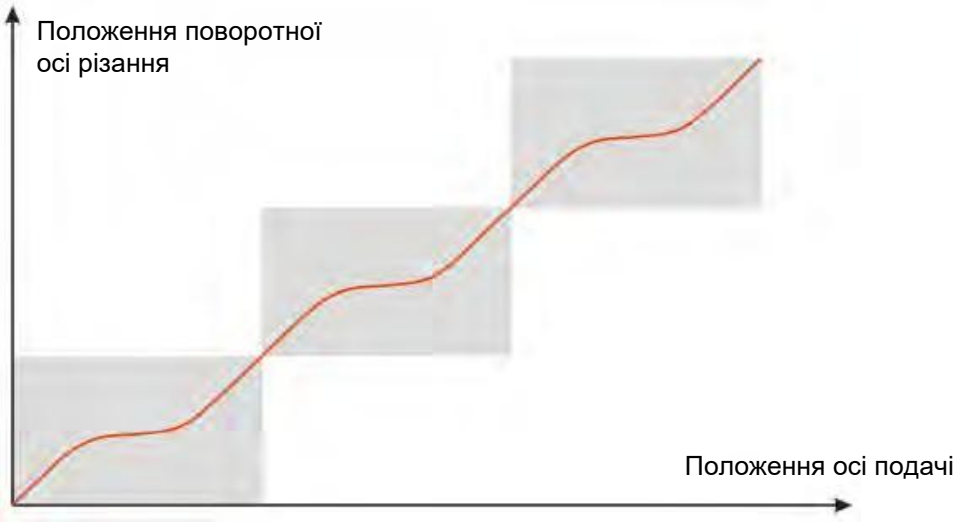
11.5.2 Параметри ротаційного різання



Параметр на малюнку	Пояснення	Відповідне ім'я параметра інструкції
L	Довжина різання оброблюваного матеріалу	APF_RotaryCut_Init.CutLength
R1	Радіус осі подачі, тобто радіус довжини подавального ролика.	APF_RotaryCut_Init.FeedRadius
R2	Радіус поворотної осі, тобто відстань від центру поворотного валика до долота.	APF_RotaryCut_Init.RotaryRadius
N	Кількість ножів поворотного ролика. На малюнку вище кількість ножів 1.	APF_RotaryCut_Init.KnifeNum
P1	Вихідне положення синхронної області.	APF_RotaryCut_Init.SyncStartPos
P2	Кінцеве положення синхронної області.	APF_RotaryCut_Init.SyncStopPos

11.5.3 Функція керування функцією поворотного різання

Функція ротаційного різання є різновидом спеціальної електронної функції кулачка. Малюнок кривої кулачка показаний нижче для безперервного різання.



- **особливості**
  1. Користувачі можуть вільно встановлювати довжину різання відповідно до технологічних вимог, і довжина різання може бути меншою або більшою за окружність різця.
  2. У зоні SYNC обертальна вісь різання та вісь подачі рухаються з певною швидкістю. (Їх швидкості зазвичай однакові.) А різання матеріалу ведеться в зоні SYN.
  3. Контролер підтримує обертовий ролик з кількома ножами.
  4. Вісь подачі може здійснювати постійний рух, прискорення, уповільнення та нерегулярний рух, оскільки вісь обертального різання рухається відповідно до фази осі подачі після ввімкнення функції обертального різання.
  5. Коли зв'язок обертального різання розривається, ніж зупиняється в нульовій точці системи, тобто в положенні входу для обертального різання.

11.5.4 Вступ до кулачкової кривої з функцією поворотного різання

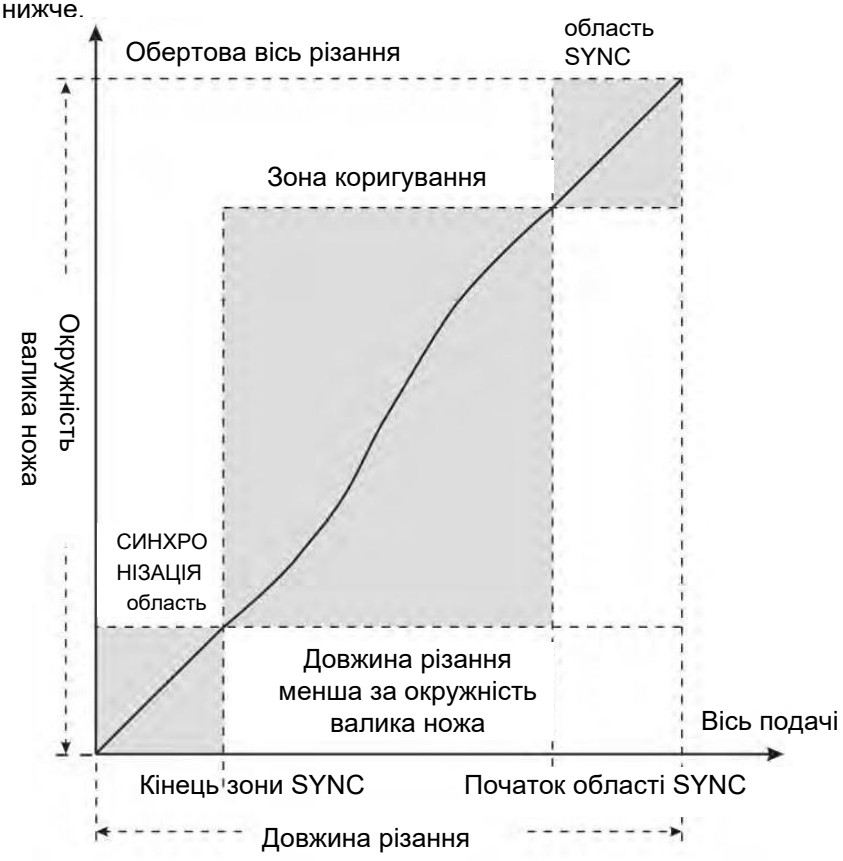
Криву кулачка з функцією поворотного різання можна розділити на зону SYNC і зону регулювання.

Зона SYNC : вісь подачі та вісь обертання здійснюють рух із фіксованим співвідношенням швидкості (лінійна швидкість ножа зазвичай дорівнює швидкості поверхні різу), а різання матеріалу відбувається в зоні SYNC.

Зона регулювання : через різну довжину різання необхідно відповідним чином відрегулювати положення. Зона регулювання може бути в наступних трьох ситуаціях на основі різної довжини різання.

1. Короткий розкрій матеріалу

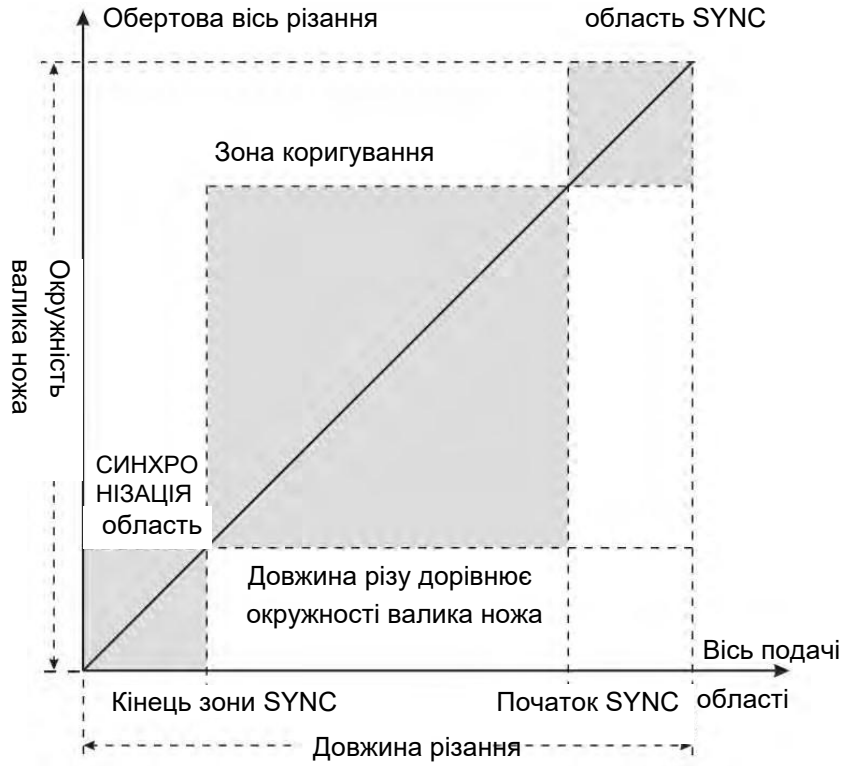
Якщо довжина різання менша за окружність ролика ножа, крива обертального різання для будь-якого циклу показана нижче.



Для різання короткого матеріалу вісь обертового різу повинна спочатку прискорюватися в зоні регулювання, а потім сповільнюватися до синхронної швидкості.

2. Рівнодовжина нарізка

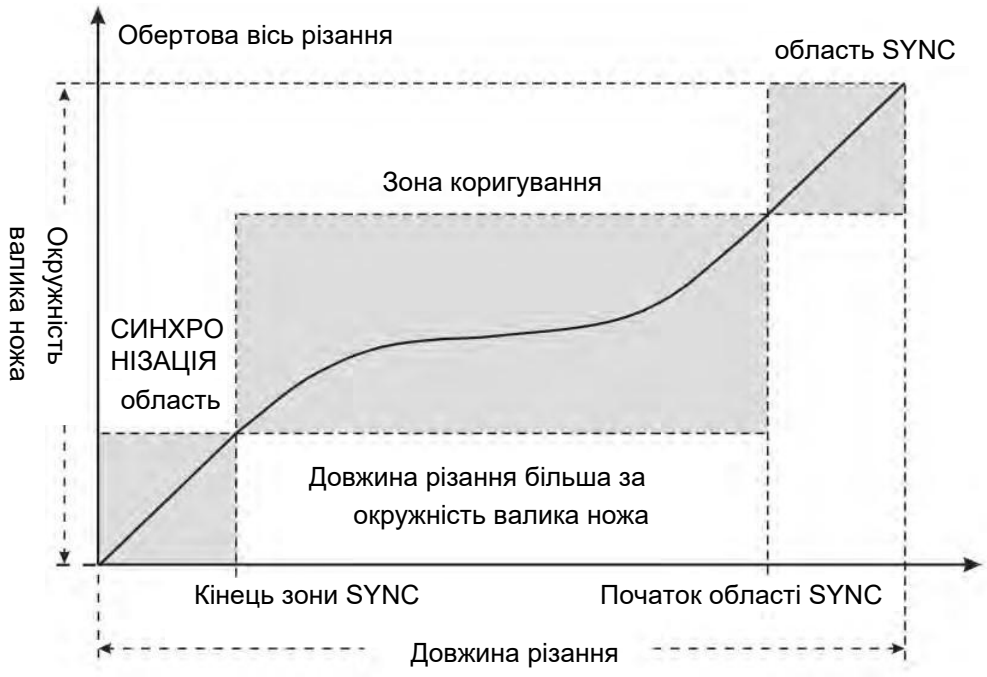
Коли довжина різання дорівнює окружності валика ножа, крива обертального різання для будь-якого циклу показана нижче.



У цій ситуації вісь подачі та вісь обертального різання в зоні SYNC та зоні без SYNC зберігають синхронну швидкість. Обертова вісь різання не потребує регулювання.

3. Розкрій довгого матеріалу

Якщо довжина різання перевищує окружність ножового валика, нижче показана крива обертального різання для будь-якого циклу.



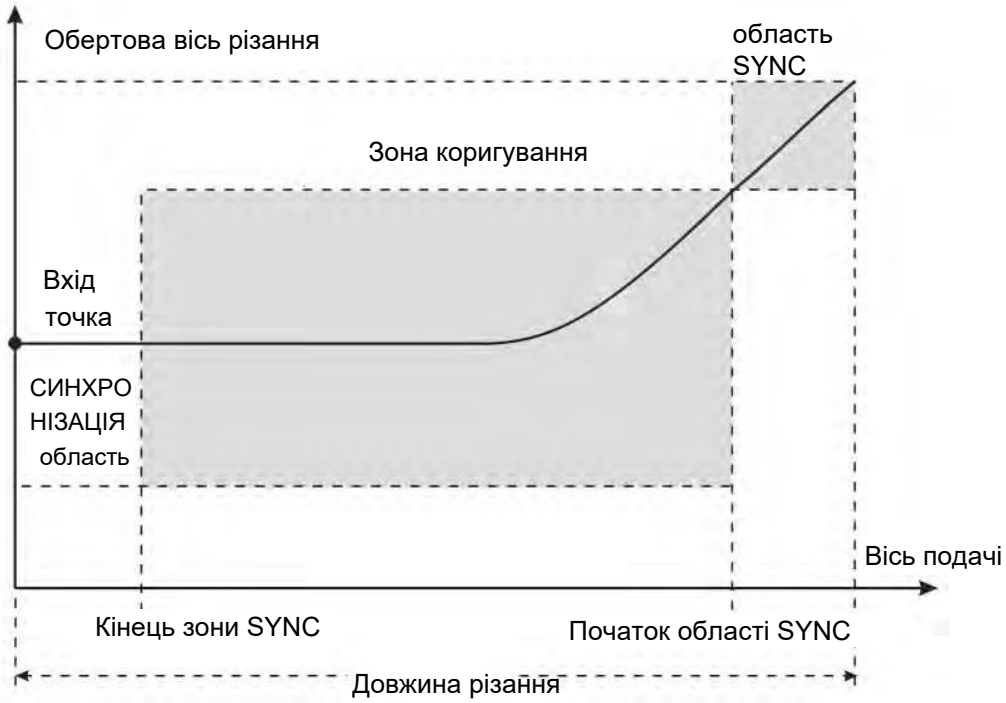
У цій ситуації обертова вісь різання повинна сповільнюватися спочатку в зоні регулювання, а потім прискорюватися до синхронної швидкості. Якщо довжина різання набагато більша за окружність обертового ріжучого ролика, ролик може сповільнитися до 0 і деякий час залишатися нерухомим; а потім розігнатися до синхронної швидкості. The

чим довжина різання, тим довше залишається ролик.

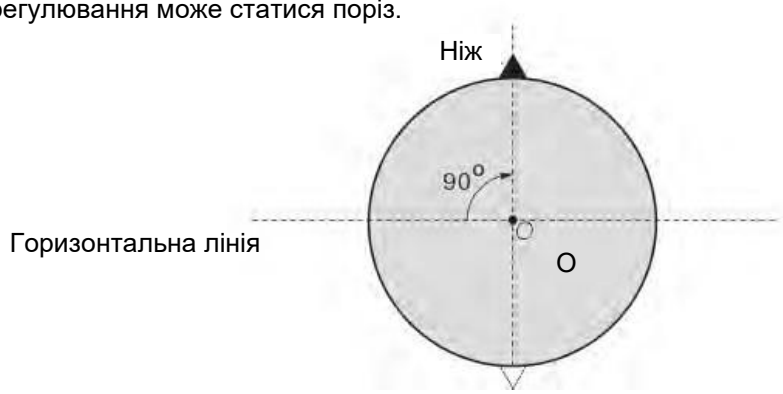
Крім того, коли функція обертового різання запускається або припиняється, криві кулачка використовуються різні.

4. Крива входу

Це крива обертального різання, коли запускається функція ротаційного різання.

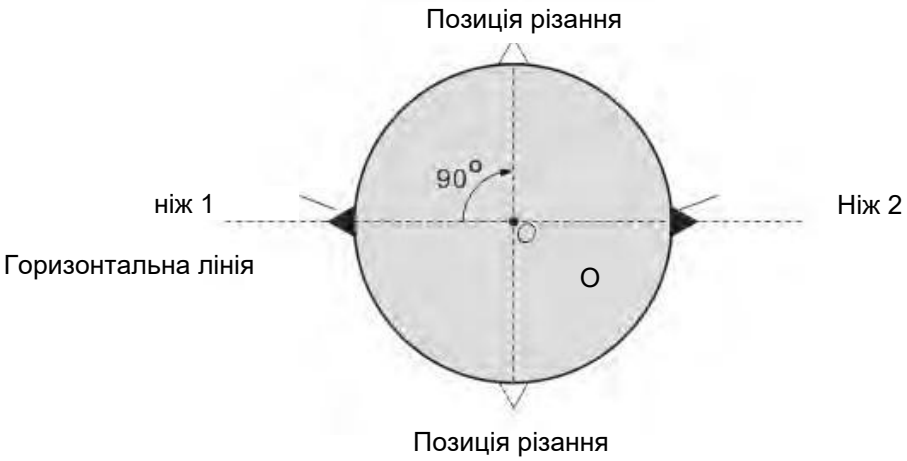


Крива — це крива входу функції обертального різання. Коли функція обертального різання запущена, вісь обертального різання слідуватиме за віссю подачі, щоб обертатися відповідно до кривої. Позиція входу заснована на поворотній осі різання. Для одного ножа позиція різання знаходиться безпосередньо під обертовим ріжучим роликом, якщо на наступному малюнку вхідна позиція знаходиться над обертовим ріжучим валиком. Перш ніж запустити функцію обертового різання, ніж потрібно повернути до верхньої частини обертового ролика. Інакше в зоні регулювання може статися поріз.



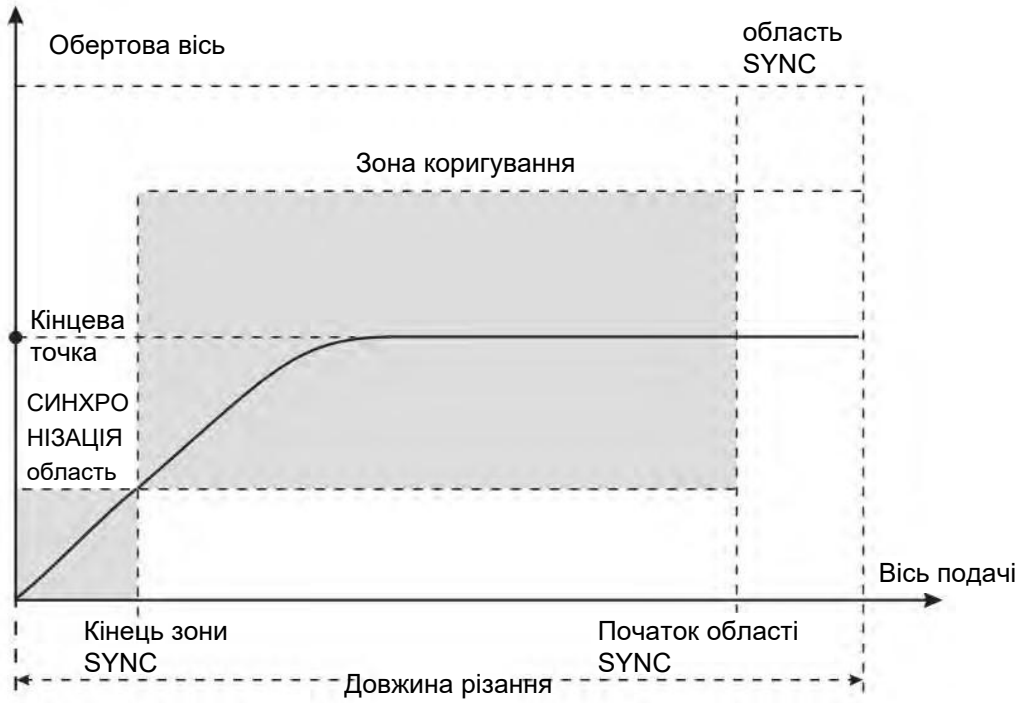
Позиція різання

Коли обертовий ролик встановлено з декількома ножами, відстані між ножами мають бути однаковими, а положення різання знаходиться в центрі відстані між ножами. Дивіться малюнок із двома ножами нижче.



5. Кінцева крива

Це крива обертального різання, коли функція ротаційного різання відірвана.



Після запуску інструкції «APF_RotaryCut_Out» система використовуватиме криву, щоб вісь поворотного різання вийшла зі стану обертального різання. Згодом ніж зупиняється в кінцевому положенні, як показано на малюнку вище.

Кінцеве положення базується на осі обертання. Для одинарного ножа кінцеве положення є входом, і воно також знаходиться прямо над обертовим ріжучим роликом.

11.5.5 Інструкції з ротаційного різання

11.5.5.1 APF_RotaryCut_Init

FB/FC	Пояснення	Застосовна модель
FB	APF_RotaryCut_Init використовується для ініціалізації радіуса осі обертання та осі подачі, довжини різання, синхронної площі тощо.	AS516E-B AS532EST-B AS564EST-B

Екземпляр APF_Rotary Cu t_Init_



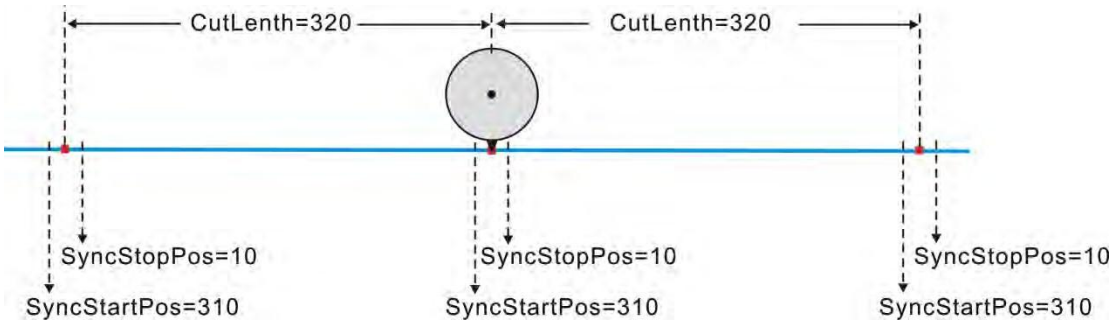
- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли Execute змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Радіус поворотної осі	Радіус осі обертового різання, тобто відстань від центру різального ролика до ножа.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
RotaryAxisKnifeNum	Кількість ножів обертової осі різання, тобто кількість ножів, встановлених на обертовому різальному ролику	USINT	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
Радіус осі подачі	Радіус осі подачі; тобто радіус довжини подаючого ролика	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
CutLength	Довжина різання матеріалу	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
SyncStartPos	Початкова позиція області синхронізації, тобто відповідна позиція осі подачі, коли починається область синхронізації.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
SyncStopPos	Кінцеве положення області синхронізації, тобто відповідне положення осі подачі, коли область синхронізації закінчується.	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
RotStartPos	Зарезервований	-	-	-
FedStartPos	Зарезервований	-	-	-
RotaryCutID	Номер для групи інструкцій обертального різання; група інструкцій обертального різання використовує той самий номер. Діапазон налаштувань: 1~8.	USINT	1~8 (Значення змінної має бути встановлено)	Коли *Execute* змінюється з FALSE на TRUE

Примітки:

1. Значення «SyncStartPos» в області SYNC завжди більше, ніж значення «SyncStopPos» в області SYNC. Як показано на малюнку нижче, довжина різання становить 320; «SyncStartPos» — 310, а «SyncStopPos» — 10.



2. Обмеження для області SYNC полягає в тому, що вона не повинна перевищувати половину довжини різання. На малюнку вище зона SYNC дорівнює 20, а половина довжини різання – 160.
3. Параметри довжини у функції: RotaryAxisRadius, FeedAxisRadius, CutLenth, SyncStartPos і SyncStopPos з однаковими одиницями вимірювання. Наприклад, якщо одиницею для одного з параметрів є CM (сантиметр), одиницями для інших параметрів також повинні бути CM.

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли інструкція виконана.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли ініціалізація завершена.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **функція**

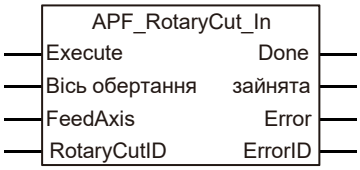
Перед тим, як буде встановлено співвідношення обертального різання, інструкція використовується для ініціалізації радіуса осі обертального різання та осі подачі, довжини різання, площі SYNC та інших параметрів. Після успішного виконання інструкції відповідні параметри будуть завантажені для використання у встановленому відношенні обертального різання.

Після того, як буде встановлено зв’язок ротаційного різання, інструкцію можна використовувати для зміни параметрів ротаційного різання. Після завершення виконання інструкції нові параметри запрацюють у наступному циклі.

11.5.5.2 APF_RotaryCut_In

FB/FC	Пояснення	Застосовна модель
FB	APF_RotaryCut_In використовується для встановлення зв'язку обертального різання та вказівки номера осі обертального різання та осі подачі відповідно до вимог застосування.	AS516E-B AS532EST-B AS564EST-B

APF_Rotary Cut_In_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Вісь обертання	Номер осі поворотної осі	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
FeedAxis	Номер осі осі подачі.	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
RotaryCutID	Номер для групи інструкцій обертального різання; група інструкцій обертального різання використовує той самий номер. Діапазон налаштувань: 1~8.	USINT	1~8 (Змінна потрібно встановити значення)	Коли Execute змінюється з FALSE на TRUE

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли інструкція виконана.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

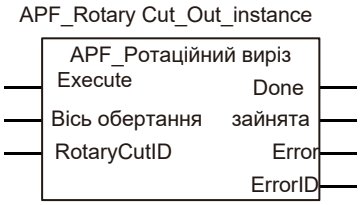
Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли з'єднання між віссю поворотного різання та віссю подачі завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **функція**

APF_RotaryCut_In використовується для побудови співвідношення обертального різання та вказівки номера осі обертального різання та осі подачі відповідно до вимог програми. Вісь обертального різання слідуватиме за віссю подачі для руху на основі кривої обертального різання після успішного виконання інструкції.

11.5.5.3 APF_RotaryCut_Out

FB/FC	Пояснення	Застосовна модель
FB	APF_RotaryCut_Out використовується для роз'єднання вже встановленого співвідношення обертального різання між віссю обертального різання та віссю подачі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

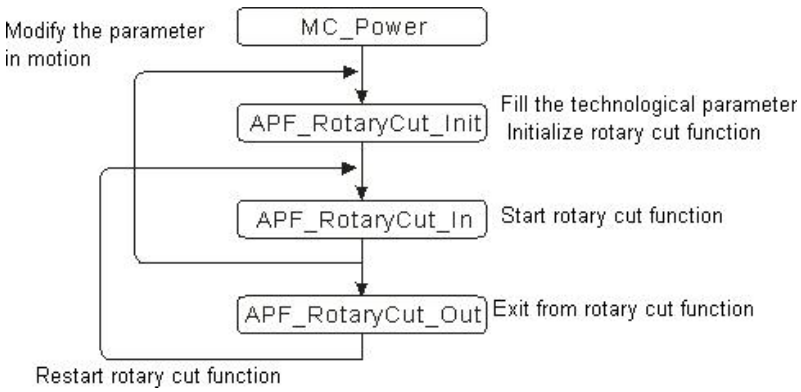
Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	TRUE чи FALSE (ЛОЖЬ)	
Вісь обертання	Номер осі поворотної осі	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли Execute змінюється з FALSE на TRUE
RotaryCutID	Номер для групи інструкцій обертального різання; група інструкцій обертального різання використовує той самий номер. Діапазон налаштувань: 1~8.	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE

- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли інструкція виконана.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

Примітки:

1. Схема послідовності керування функцією обертового різання



2. Коли виконується функція обертального різання, вісь обертального різання може виконувати лише інструкції APF_RotaryCut_Out і MC_Stop, а інші інструкції недійсні.

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Після завершення роз'єднання ротаційно-різового зв'язку.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **функція**

APF_RotaryCut_Out використовується для роз’єднання вже встановленого співвідношення обертального різання між віссю обертального різання та віссю подачі. Після того, як взаємозв’язок обертального різання буде роз’єднано, ніж осі обертального різання зупиниться у вихідному положенні та більше не рухатиметься за віссю подачі.

Інструкція не впливає на рух осі подачі.

11.5.6 Приклад застосування інструкцій з ротаційного різання

У цьому розділі пояснюється налаштування параметрів обертального різання, встановлення та відключення співвідношення обертального різання. Нижче наведено приклад програмування.

Перегляньте ключові параметри в прикладі, як показано в таблиці нижче

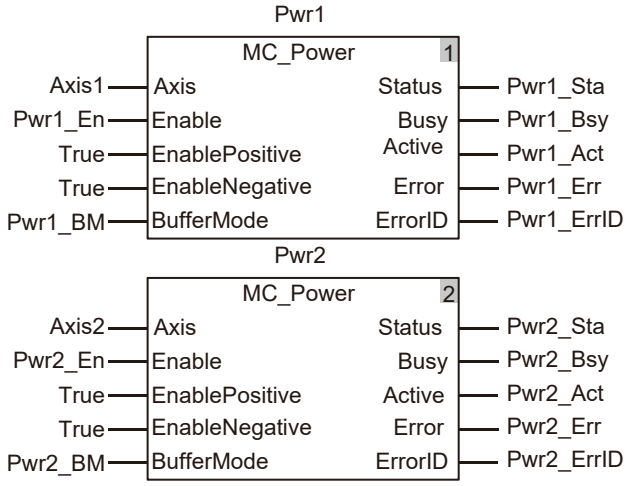
Назва параметра	Поточне значення
Вісь обертання	2
FeedAxis	1
Радіус поворотної осі	10 (Одиниця: одиниці)
RotaryAxisKnifeNum	1
Радіус осі подачі	20 (Одиниця: одиниці)
CutLenth	30 (Одиниця: одиниці)
SyncStartPos	19 (Одиниця: одиниці)
SyncStopPos	1 (Одиниця: одиниця)

Приклад програмування

1. Оскільки Pwr1_En має значення TRUE, сервопривод з адресою вузла 1 вмикає «Servo On»; оскільки Pwr2_En має значення TRUE, сервопривід адреси вузла 2 вмикає «Servo On».

Таблиця змінних і програма

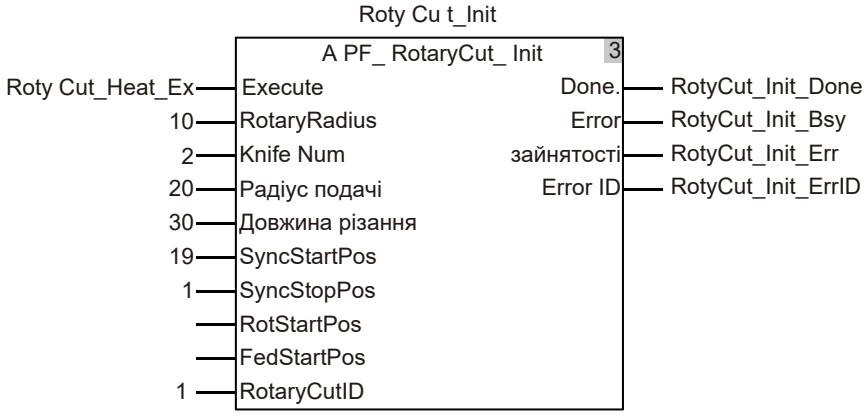
Ім'я змінної	Тип даних	Початкове значення
Pwr1	MC_Power	
Axis1	USINT	1
Pwr1_En	BOOL	TRUE
Pwr1_BM	MC_Buffer_Mode	0
Pwr1_Sta	BOOL	TRUE
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
Pwr2	MC_Power	
Axis2	USINT	1
Pwr2_En	BOOL	TRUE
Pwr2_BM	MC_Buffer_Mode	0
Pwr2_Sta	BOOL	TRUE
Pwr2_Bsy	BOOL	
Pwr2_Act	BOOL	
Pwr2_Err	BOOL	
Pwr2_ErrID	WORD	



2. Встановіть параметри технології ротаційного різання. Радіус осі обертання різання становить 10, кількість ножів осі обертання різання становить 1, радіус осі подачі становить 20, а довжина різання осі подачі становить 30. Початкова позиція області SYNC 19, кінцева позиція області SYNC дорівнює 1, а номер групи ротаційного різання – 1. Якщо RotyCut_Init_Ex має значення TRUE, параметри технології ротаційного різання буде ініціалізовано.

Таблиця змінних і програма

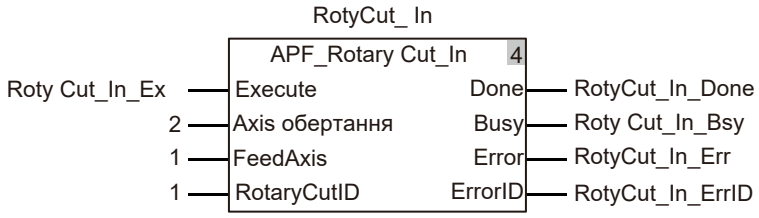
Ім'я змінної	Тип даних	Початкове значення
RotyCut_Init	APF_RotaryCut_Init	
RotyCut_Init_Ex	BOOL	TRUE
RotyCut_Init _Done	BOOL	TRUE
RotyCut_Init _Bsy	BOOL	
RotyCut_Init _Err	BOOL	
RotyCut_Init _ErrID	WORD	



3. Коли RotyCut_In_Ex має значення TRUE, починає встановлюватися зв'язок ротаційного різання. Якщо RotyCut_In_Done має значення TRUE, це вказує на те, що зв'язок обертального різання між віссю обертального різання та віссю подачі виконано успішно. Серво 1 є віссю подачі, а серво 2 є віссю поворотного різання.

Таблиця змінних і програма

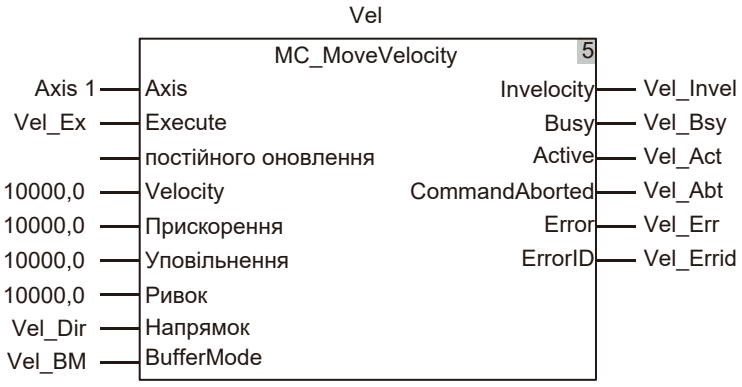
Ім'я змінної	Тип даних	Початкове значення
RotyCut_In	APF_RotaryCut_In	
RotyCut_In_Ex	BOOL	TRUE
RotyCut_In _Done	BOOL	TRUE
RotyCut_In _Bsy	BOOL	
RotyCut_In _Err	BOOL	
RotyCut_In _ErrID	WORD	



4. Коли Vel _Ex має значення TRUE, вісь подачі починає виконувати інструкцію швидкості. На даний момент вісь обертального різання виконує дію обертального різання на основі фази осі подачі.

Таблиця змінних і програма

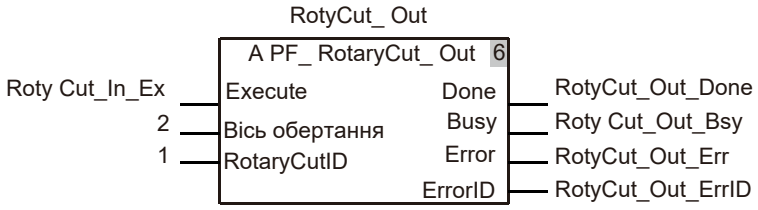
Ім'я змінної	Тип даних	Початкове значення
Vel	MC_MoveVelocity	
Axis1	USINT	1
Vel _Ex	BOOL	TRUE
Vel _Dir	MC_DIRECTION	1
Vel _BM	MC_Buffer_Mode	0
Vel _Invel	BOOL	
Vel _Bsy	BOOL	
Vel _Act	BOOL	
Vel _Abt	BOOL	
Vel _Err	BOOL	
Vel _ErrID	WORD	



5. Коли RotyCut_Out_Ex має значення TRUE, вісь обертального різання починає відриватися від осі подачі. Якщо RotyCut_Out_Done має значення TRUE, це вказує на те, що вісь поворотного різання успішно відривається. Після того, як вісь обертального різання відірветься від осі подачі, вона повернеться до точки входу, і рух осі подачі більше не впливатиме на вісь обертального різання.

Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
RotyCut_Out	APF_RotaryCut_Out	
RotyCut_Out_Ex	BOOL	TRUE
RotyCut_Out_Done	BOOL	TRUE
RotyCut_Out_Bsy	BOOL	
RotyCut_Out_Err	BOOL	
RotyCut_Out_ErrID	WORD	



11.6 Інструкції з коду G

11.6.1 ЧПУ Вступ

Як багатоосьовий контролер руху, контролер руху підтримує стандартну функцію ЧПУ і може статично виконувати G-коди для досягнення простого числового керування верстатами та роботом. Крім того, його також можна застосувати у випадках, коли G-коди використовуються для позиціонування та планування шляху.

Програмне забезпечення CANopen Builder забезпечує функцію редагування коду CNC G; користувач може редагувати G-коди в редакторі CNC або імпортувати G-коди, перетворені іншим програмним забезпеченням для проектування, у цей редактор. Коли G-коди вводяться в список кодів, тривимірна діаграма G-кодів виводиться у вікні попереднього перегляду.

Усі G-коди будуть завантажені на контролер під час завантаження програми.

G-коди потрібно викликати в програмі керування рухом після редагування. За допомогою інструкції DMC_CartesianCoordinate можна керувати віссю сервоприводу для інтерполяції положення.

11.6.2 Формат введення коду G

Формати коду G, які підтримує контролер руху, перераховані в наступній таблиці.

код G	функція	Кількість підтримуваних осей	Формат
G0	Швидке позиціонування	8	Формат 1: G0 X_ Y_ Z_ A_ B_ C_ P_ Q_
G1	Лінійна інтерполяція	8	Формат 1: G1 X_ Y_ Z_ A_ B_ C_ P_ Q_ E_ F_
G2	Дуга кола/спіральна інтерполяція за годинниковою стрілкою	8	Формат 1: G2 X_ Y_ Z_ A_ B_ C_ P_ Q_ I_ J_ (I_ K_ / J_ K_)T_ E_ F_ Формат 2: G2 X_ Y_ Z_ A_ B_ C_ P_ Q_ R_ T_ E_ F_
G3	Дуга кола/спіральна інтерполяція проти годинникової стрілки	8	Формат 1: G3 X_ Y_ Z_ A_ B_ C_ P_ Q_ I_ J_ (I_ K_ / J_ K_)T_ E_ F_ Формат 2: G3 X_ Y_ Z_ A_ B_ C_ P_ Q_ R_ T_ E_ F_
G4	Інструкція відкласти	--	Формат 1: G4 K_
G17	Площина XY для кругової інтерполяції	--	Формат 1: G17
G18	Площина XZ для кругової інтерполяції	--	Формат 1: G18
G19	Площина YZ для кругової інтерполяції	--	Формат 1: G19
G90	Абсолютний режим	--	Формат 1: G90
G91	Відносний режим	--	Формат 1: G91
G50	Точна зупинка	--	Формат 1: G50
G51	Круглий перехід шляху	--	Формат 1: G51 D_
G52	Плавний перехід шляху	--	Формат 1: G52
M0~M99	Код M	--	Формат 1: M_ D_

Примітка:

Місце з підкресленням означає значення параметра, який потрібно встановити.

Коли потрібно ввести один G-код у програму ЧПУ у CANopen Builder, N_ потрібно поставити ліворуч від G-коду. N_ означає номер рядка коду G у програмі ЧПУ.

Кожен рядок має лише один вхід G-коду.

Формат введення G-кодів у програмному забезпеченні CANopen Builder такий.

N0 G0 X100 Y100

11.6.3 Пояснення форматів коду G

- **Одиниця коду G**

  Одиниця позиції осі X_, Y_, Z_, A_, B_, C_, P_, Q_ у коді G узгоджується з одиницею параметра осі. Будь ласка, встановіть ту саму фізичну одиницю для кожної осі.
  Наприклад, одиницею виміру є мм. Таким чином, G0 X100.5 Y300 Z30.6 вказує на те, що осі X, Y, Z переміщуються на 100,5 мм, 300 мм і 30,6 мм відповідно.

- **Параметр коду G пропущено**
  1. Один або кілька елементів серед X_, Y_, Z_, A_, B_, C_, P_, Q_ в інструкції G0 можна пропустити.
  2. Один або кілька елементів серед X_, Y_, Z_, A_, B_, C_, P_, Q_, E_, E_, F_ в інструкції G1 можна пропустити.
  3. Один або кілька елементів серед X_, Y_, Z_, A_, B_, C_, P_, Q_, E_, E_, F_ в інструкції G2 і G3 можна пропустити, за винятком I_, J_, K_, R_.
  4. Параметри праворуч від інструкції G4, G51 не можна пропускати.
  5. D_ можна опустити для коду M.
  6. Тільки один код G можна записати в одному рядку в області редагування ЧПУ у програмному забезпеченні CANopen Builder.

- **Спеціальна функція коду G**
  - **Використання регістра %ML для представлення ключових значень у коді G**

    X_, Y_, Z_, A_, B_, C_, P_, Q_, E_, F_, I_, J_, K_, R_, T_, E_, F_ усі можуть використовувати регістр %ML. «%» з «%ML» видаляється, а «$» додається праворуч і ліворуч від «ML». T має тип ULINT, а інші мають тип LREAL.

    Приклад: N0 G0 X$ML0$ Y$ML1$ Z$ML2$ (%ML0=100.0 · %ML1=200.0 · %ML2=300.0)

    Пояснення: після виконання коду G вісь X переміщується на 100 одиниць; вісь Y переміщується на 200 одиниць, а вісь Z — на 300 одиниць.

  - **Перехід коду G**

    Режим переходу коду G можна змінити за допомогою G50/G51/G52. Нижче наведено режими переходу, які можна використовувати для G0/G1/G2/G3.

	G50 (Точна зупинка)	G51 (Круговий перехід)	G52 (Плавний перехід шляху)
G0	Можна використовувати	Режим переходу недійсний, а ефект руху такий самий, як у G50.	Режим переходу недійсний, а ефект руху такий самий, як у G50.
G1	Можна використовувати	Можна використовувати	Можна використовувати
G2	Можна використовувати	Можна використовувати	Режим переходу можна використовувати, коли пряма лінія або дуга кола та дуга кола дотичні або близькі до точки дотику.

	G50 (Точна зупинка)	G51 (Круговий перехід)	G52 (Плавний перехід шляху)
G3	Можна використовувати	Можна використовувати	Режим переходу можна використовувати, коли пряма лінія або дуга кола та дуга кола дотичні або близькі до точки дотику.

- **За замовчуванням**
  1. *Відносний, абсолютний режим за замовчуванням* : режим за замовчуванням є абсолютним режимом і може бути встановлений за допомогою G90/G91.
  2. *Літак за замовчуванням:* Площиною за замовчуванням є площина XY, яку можна перемикати через G17/G18/G19.
  3. *Режим переходу:* Площина за замовчуванням є режимом точної зупинки, і її можна перемикати за допомогою G50/G51/G52.
  4. *Значення за замовчуванням, пов'язані з G0:* Швидкість, прискорення, уповільнення та ривок – це швидкість, прискорення, уповільнення та ривок кожної осі в параметрах групи осей. Їх можна встановити за допомогою інструкції DMC_SetG0Para.
  5. *Стандартні параметри G1/G2/G3:* Швидкість, прискорення, уповільнення та ривок – це швидкість, прискорення, уповільнення та ривок термінального приводу. Їх можна встановити за допомогою інструкції DMC_SetG1Para та змінити за допомогою параметра E, F. E+ і E- можна ввести в код G, щоб встановити різну швидкість прискорення та уповільнення.

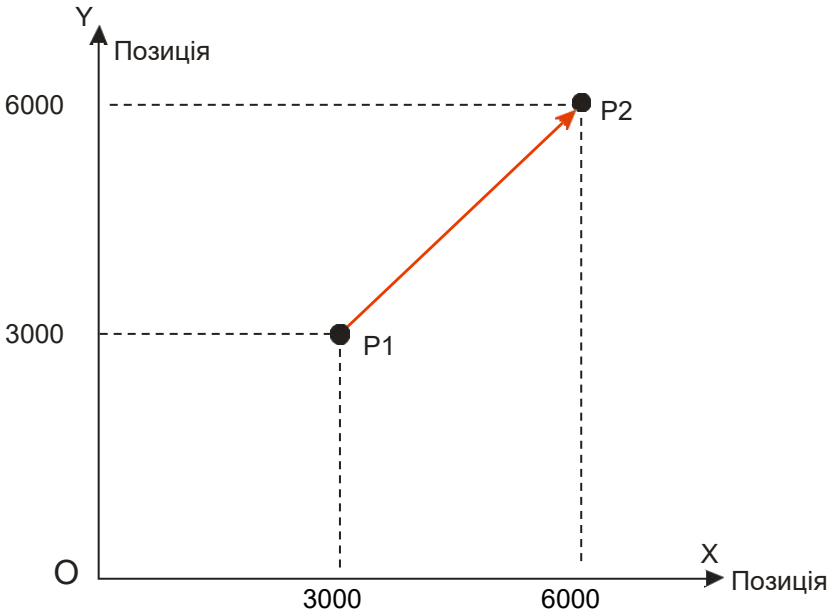
     Приклад: G1 X10000 Y32105.6 E+20000 E-90000

     Пояснення: коли інструкція виконується, різак рухається з прискоренням 20000 одиниць/с ^{2} для прискорення та з уповільненням 90000 одиниць/с ^{2} для зменшення швидкості.

11.6.4 Функції коду G

11.6.4.1 G90 (абсолютний режим)

- Функція: після виконання G90 кінцеве положення кожної осі в коді G базується на 0 одиницях, і G91 можна використовувати для перемикання у відносний режим у процесі. За замовчуванням це абсолютний режим для програми ЧПУ.
- Формат: N_G90
- Пояснення параметра:
- N_ : номер рядка коду G у програмі ЧПУ
- приклад:
  Початкові позиції осей X і Y становлять 3000 одиниць, і обидва параметри осі є значеннями за замовчуванням. G-коди, які потрібно виконати, такі:
  N0 G90
  N1 G0 X6000 Y6000
  Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



11.6.4.2 G91 (відносний режим)

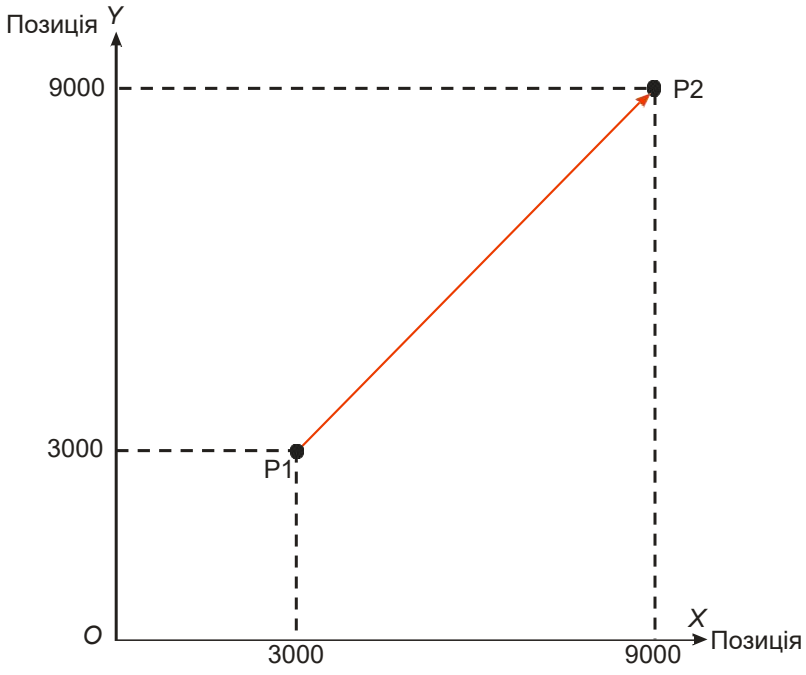
- Функція: після виконання G91 кінцеве положення кожної осі в G-коді підраховується інкрементним методом, починаючи з поточного положення, і G90 може використовуватися для перемикання в абсолютний режим у процесі.
- Формат: N_G91
- Пояснення параметра:
- N_: Номер рядка коду G у програмі ЧПУ
- приклад:

  Початкові позиції осей X і Y становлять 3000 одиниць, і параметри осі є значеннями за замовчуванням. G-коди, які потрібно виконати, такі:

  N0 G91

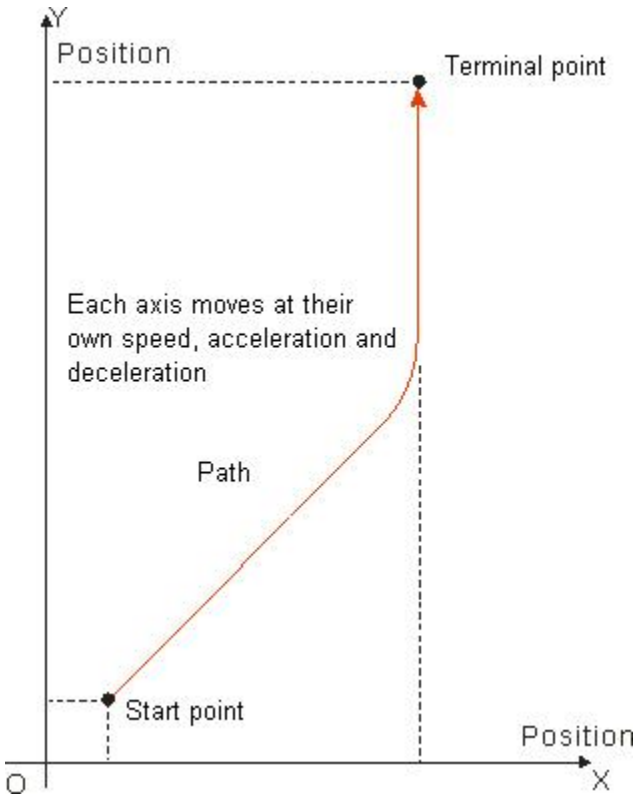
  N1 G0 X6000 Y6000

  Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



11.6.4.3 G0 (швидке позиціонування)

- функція:
  Кожна вісь рухається з поточної позиції в кінцеву позицію з заданою швидкістю. Можна керувати максимум 8 осями, і кожна вісь рухається незалежно одна від одної.
  Нижче показано фігуру шляху руху.



- Формат: N_G0 X_Y_Z_A_B_C_P_Q_
- Пояснення параметра:
  N_: Номер рядка коду G у програмі ЧПУ.
  X_: укажіть кінцеве положення осі X, одиниця: одиниця, тип даних: LREAL.
  Y_: укажіть кінцеве положення осі Y, одиниця: одиниця, тип даних: LREAL.
  Z_: укажіть кінцеве положення осі Z, одиниця: одиниця, тип даних: LREAL.
  A_: укажіть кінцеве положення осі A, одиниця: одиниця, тип даних: LREAL.
  B_: укажіть кінцеву позицію осі B, одиниця: одиниця, тип даних: LREAL.
  C_: укажіть кінцеву позицію осі C, одиниця: одиниця, тип даних: LREAL.
  P_: вкажіть кінцеве положення осі P, одиниця: одиниця, тип даних: LREAL.
  Q_: вкажіть кінцеве положення осі Q, одиниця: одиниця, тип даних: LREAL.

- Пояснення до інструкції:
  1. G0 може керувати однією або декількома осями, а іншу вісь можна опустити.
  2. Швидкість, прискорення, уповільнення та ривок кожної осі в русі залежать від пов’язаних із віссю параметрів у параметрах групи осей. Їх можна встановити за допомогою інструкції DMC_SetG0Para.
  3. Абсолютний режим визначається G90: кінцева позиція G0 заснована на 0 одиницях.
  4. Відносний режим, визначений G91: Кінцева позиція G0 є додатковим значенням, починаючи з поточної позиції.

■ Приклад абсолютного режиму:

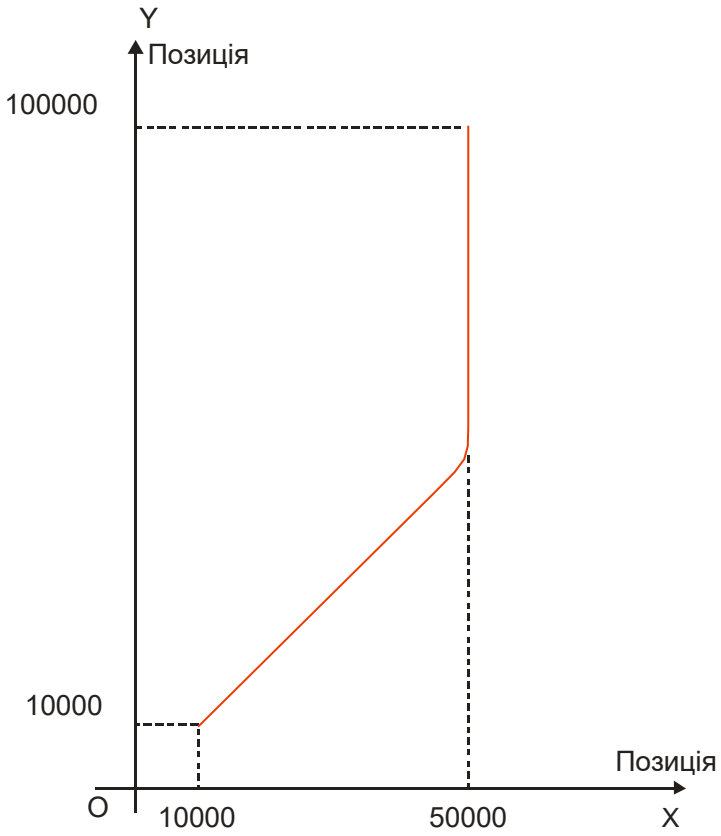
Початкові позиції осей X, Y обидва становлять 10 000 одиниць, і параметри їхніх осей є значеннями за замовчуванням.

Коди G, які потрібно виконати:

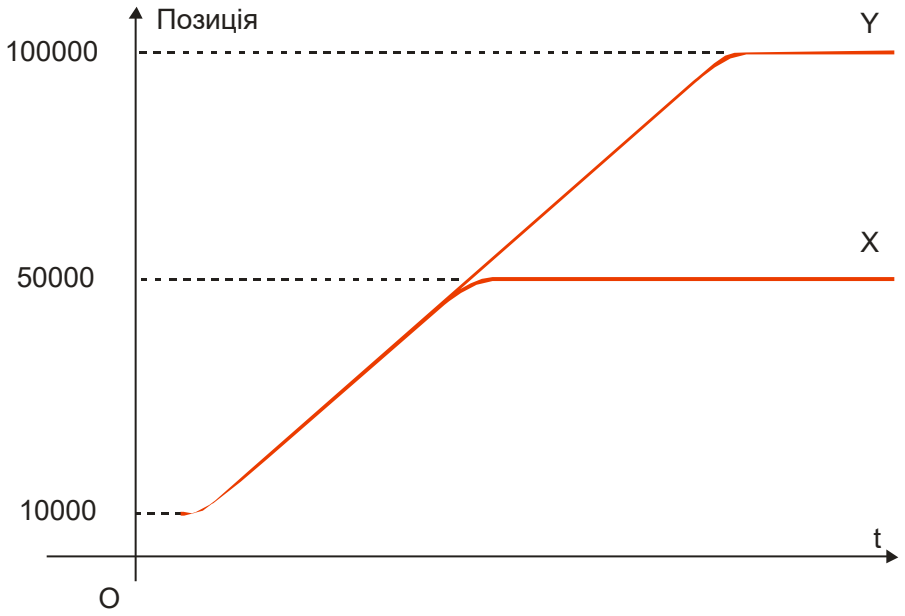
N0 G90

N1 G0 X50000 Y100000

➢ Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



➢ Після виконання G-кодів крива «Позиція/Час» для всього процесу руху показана нижче:



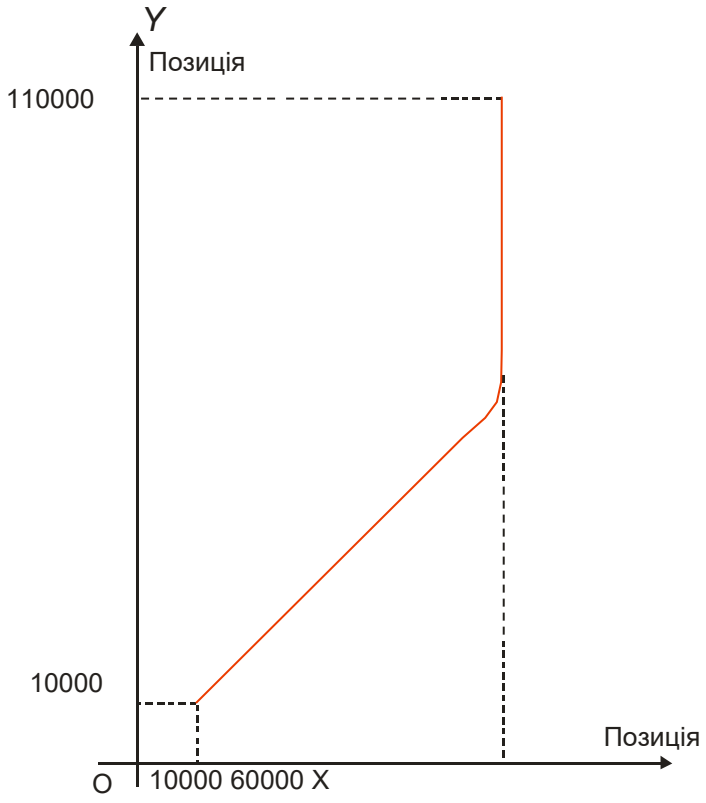
■ Приклад відносного режиму:

Початкові позиції осей X, Y обидва становлять 10 000 одиниць, і параметри їхніх осей є значеннями за замовчуванням. G-коди, які потрібно виконати:

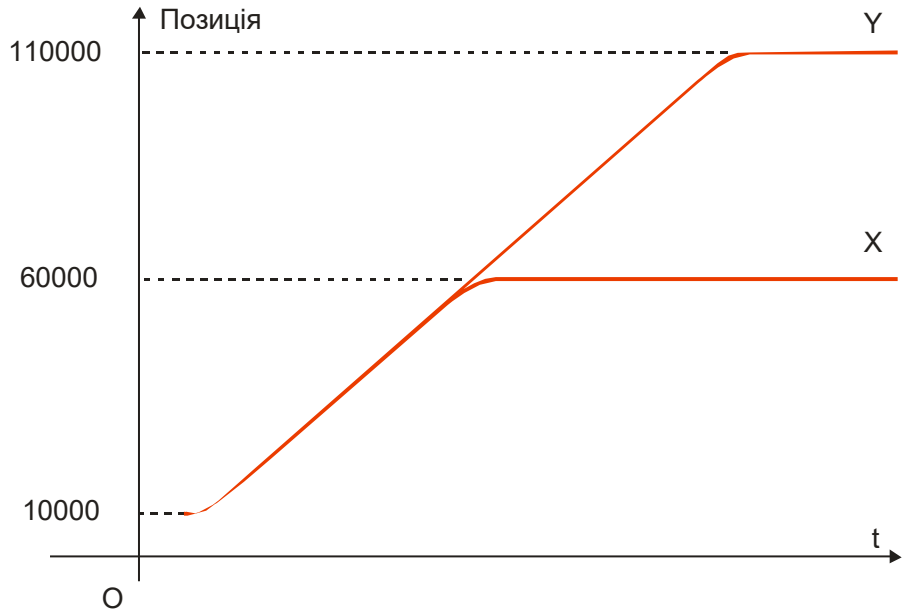
N0 G91

N1 G0 X50000 Y100000

➢ Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:

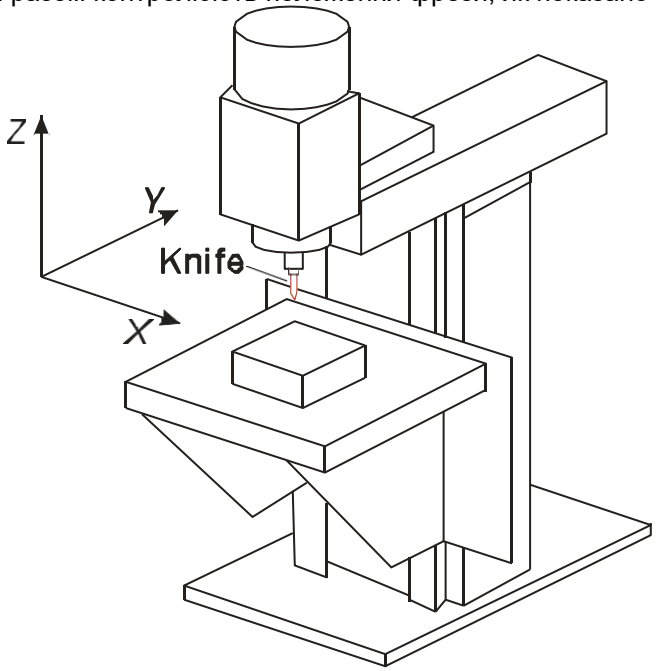


➢ Після виконання G-кодів крива «Позиція/Час» для всього процесу руху показана нижче:



11.6.4.4 G1 (лінійна інтерполяція)

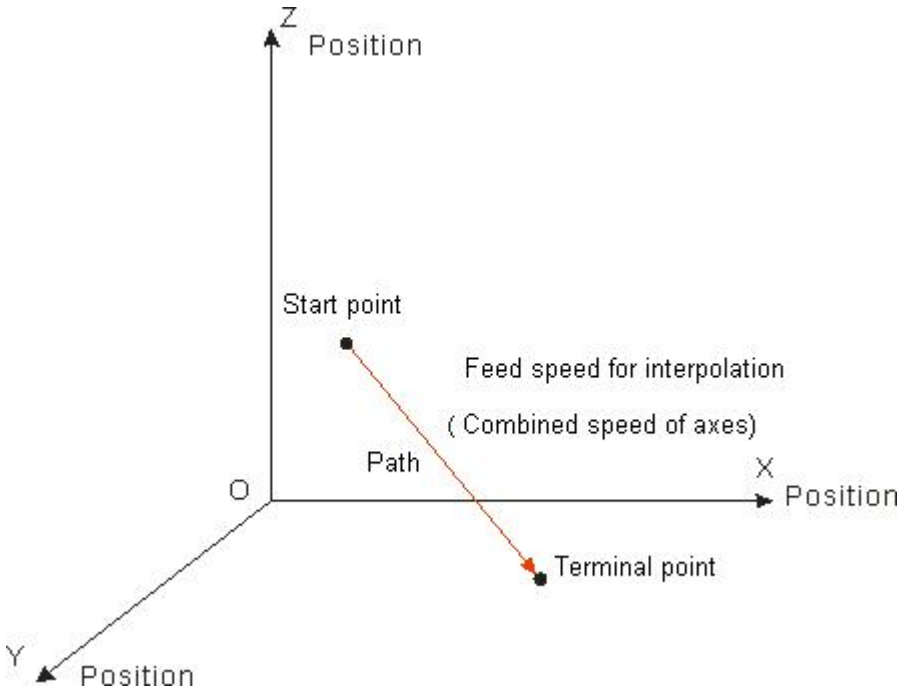
- функція:
  Фреза починає рух з однієї точки і рухається прямо до цільової позиції з заданою швидкістю. Ця інструкція може контролювати до 8 осей, і всі осі запускаються або зупиняються одночасно. Три осі разом контролюють положення фрези, як показано на малюнку нижче.



Вертикальний
фрезерний верстат

11

Фігура шляху руху:



- Формат: N_G1 X_Y_Z_A_B_C_P_Q_ E_ E_F_
- Пояснення параметра:

  N_ : номер рядка коду G у програмі ЧПУ

  X_ : вкажіть кінцеве положення осі X, одиниця: одиниця, тип даних: LREAL.

  Y_ : вкажіть кінцеве положення осі Y, одиниця: одиниця, тип даних: LREAL.

  Z_ : вкажіть кінцеве положення осі Z, одиниця: одиниця, тип даних: LREAL.

  A_ : вкажіть кінцеве положення осі A, одиниця: одиниця, тип даних: LREAL.

  B_ : вкажіть кінцеву позицію осі B, одиниця: одиниця, тип даних: LREAL.

  C_ : вкажіть кінцеву позицію осі C, одиниця: одиниця, тип даних: LREAL.

  P_ : вкажіть кінцеве положення осі P, одиниця: одиниця, тип даних: LREAL.

  Q_ : вкажіть кінцеве положення осі Q, одиниця: одиниця, тип даних: LREAL.

  E_ : Вкажіть прискорення та уповільнення різака. Додатне число відноситься до прискорення; від'ємне число відноситься до уповільнення; одиниця: одиниця/секунда ^{2}; тип даних: LREAL.

  F_ : Укажіть швидкість подачі фрези, одиниця: одиниця/секунда, тип даних: LREAL.

  Коли різець рухається з постійною швидкістю, сумарна швидкість усіх осей у коді G дорівнює значенню F. Метод розрахунку наведено нижче.

  Коли існують дві осі, $F = \sqrt{V_1^2 + V_2^2}$.

  Коли існує три осі, $F = \sqrt{V_1^2 + V_2^2 + V_3^2}$.

  Для більшої кількості осей значення F можна обчислити таким же чином, як описано вище.

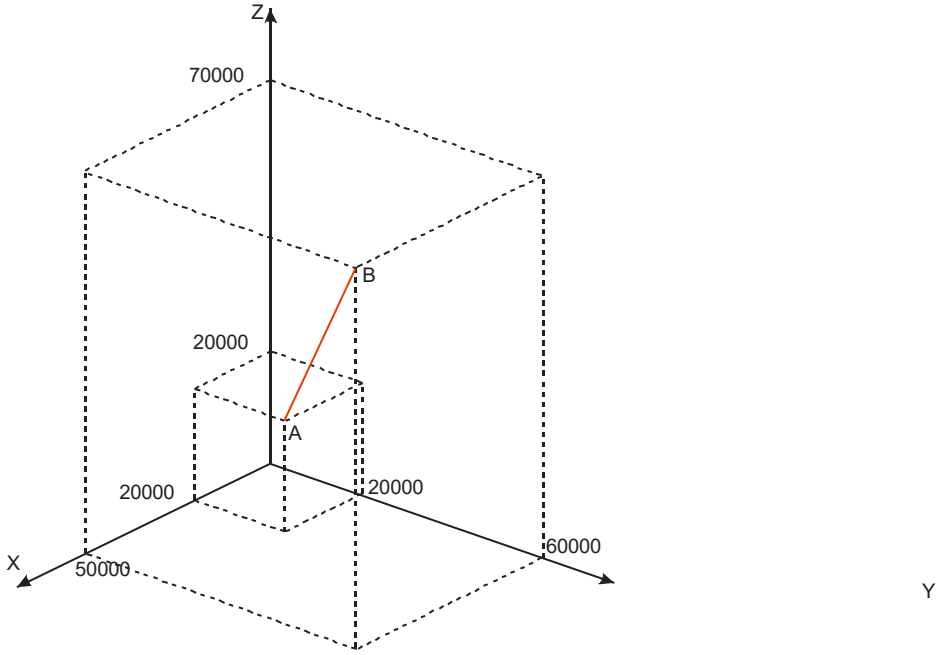
- Пояснення до інструкції:

  1. G1 може керувати однією або декількома осями, а іншу вісь можна опустити.

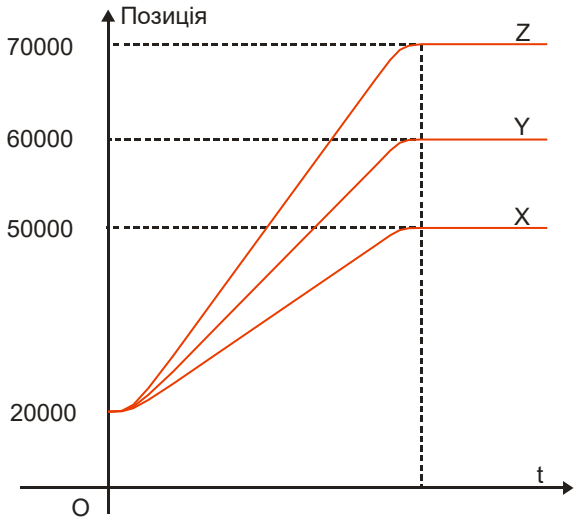
2. І E, і F можна опустити.
   Якщо в області редагування ЧПУ є лише один рядок коду, а E, F пропущені, швидкість, прискорення, уповільнення та ривок визначаються параметрами групи осей. Їх можна встановити через DMC_SetG1Para. Якщо є кілька рядків кодів, а E і F у коді G1 пропущені, швидкість, прискорення та уповільнення різця базуються на дійсних E та F у попередніх рядках кодів. Якщо в попередніх рядках G-кодів не вказано E та F, переважатимуть параметри групи осей.
3. Абсолютний режим, визначений G90: кінцева позиція G1 заснована на 0 одиницях.
4. Відносний режим, визначений G91: Кінцева позиція G1 є додатковим значенням, починаючи з поточної позиції.

➢ Приклад абсолютного режиму:

Початкові позиції осей X, Y, Z — це 20 000 одиниць, а параметри осей — значення за замовчуванням. G-коди, які потрібно виконати:
N0 G90
N1 G1 X50000 Y60000 Z70000
Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



➢ Після виконання G-кодів крива «Позиція/Час» для всього процесу руху показана нижче:



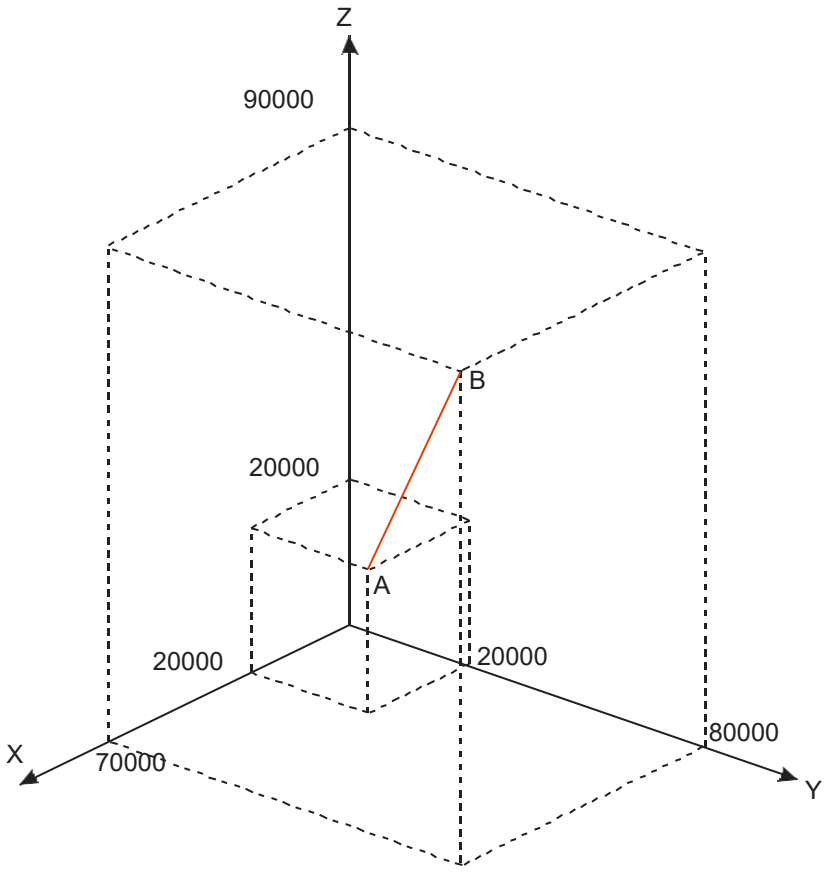
■ Приклад відносного режиму:

Початкові позиції осей X, Y, Z становлять 20 000 одиниць, а параметри осей — значення за замовчуванням. G-коди, які потрібно виконати:

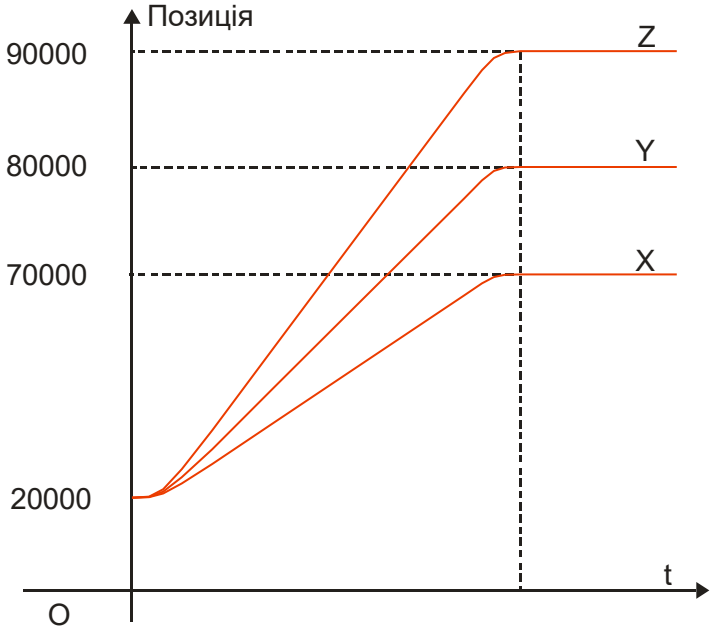
N0 G91

N1 G1 X50000 Y60000 Z70000

➢ Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



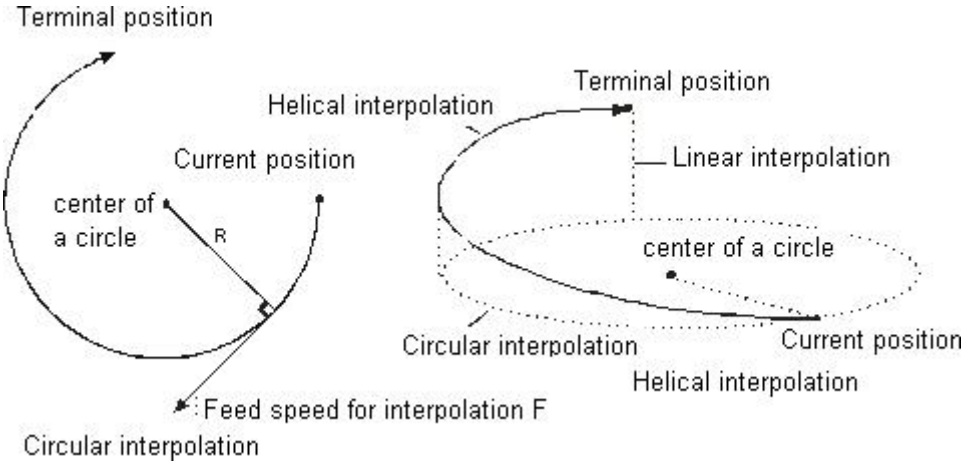
➢ Після виконання G-кодів крива «Позиція/Час» для всього процесу руху показана нижче:



11.6.4.5 G2 (кругова/спіральна інтерполяція за годинниковою стрілкою)

- функція:

  Кругова інтерполяція: різець виконує різання обробленого об'єкта за годинниковою стрілкою зі швидкістю подачі, заданою параметром F, на дузі кола з фіксованим радіусом або фіксованим центром кола зазначеної площини.
  Гвинтова інтерполяція: фреза рухається за годинниковою стрілкою по дузі кола зазначеної площини, яка є круговою інтерполяцією, і одночасно рухається у вертикальному напрямку зазначеної площини зі швидкістю подачі, заданою параметром F, яка є лінійною інтерполяцією.

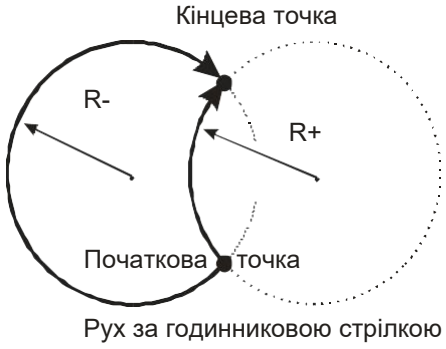


- Формат:
  Формат 1: N_G2 X_Y_Z_A_B_C_P_Q_I_J_(I_K_/J_K_)T_ E_E_F_
  Формат 2: N_G2 X_Y_Z_A_B_C_P_Q_R_T_ E_E_F_

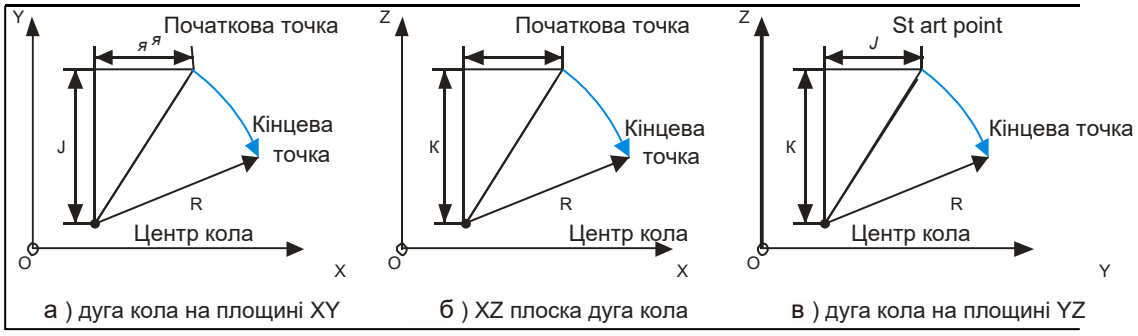
- Пояснення параметра:
  N_: Номер рядка коду G у програмі ЧПУ
  X_Y_Z_: вкажіть кінцеві позиції осей X, Y і Z, що відповідають кінцевій точці дуги кола;
  Одиниця: одиниця, тип даних: LREAL.
  A_B_C_P_Q_: Укажіть кінцеве положення кожної доданої осі,
  Одиниця: одиниця, тип даних: LREAL.

I_J_: вкажіть координатне положення центру кола площини XY, одиниця: одиниця, тип даних: LREAL.
I_K_: Вкажіть координатне положення центру кола площини XZ, Одиниця: одиниця, тип даних: LREAL.
J_K_: вкажіть координатне положення центру кола площини YZ, одиниця: одиниця, тип даних: LREAL.
T_: вкажіть кількість повних кіл, одиниця: коло, тип даних: ULINT.
E_: Вкажіть прискорення та уповільнення різця. Додатне число відноситься до прискорення; від'ємне число відноситься до уповільнення; Одиниця: одиниця/секунда ^{2}; тип даних: LREAL.
F: вкажіть швидкість подачі різця, одиниця: одиниця/секунда, тип даних: LREAL.

- Пояснення до інструкції:
  1. Дві осі серед осей X, Y і Z виконують кругову інтерполяцію на площині, визначеній інструкцією G17/G18/G19. 3-я вісь виконує лінійну інтерполяцію у вертикальному напрямку на вказаній площині.
  2. Додані осі A, B, C, P і Q роблять лінійну інтерполяцію. Лінійна інтерполяція та кругова інтерполяція запускаються або припиняються одночасно.
  3. І E, і F можна опустити. Якщо в області редагування ЧПУ є лише один рядок коду, а E, F пропущені, швидкість, прискорення та уповільнення визначаються параметрами групи осей. Їх можна встановити за допомогою інструкції DMC_SetG1Para.
     Якщо існує декілька рядків кодів, а E та F у коді G2 пропущені, швидкість, прискорення та уповільнення різця базуються на дійсних E та F у попередніх рядках кодів над рядком, де є G2. Якщо в попередніх рядках кодів G не вказано E та F, «максимальна швидкість», «максимальне прискорення» та «максимальне уповільнення» серед параметрів групи осей матимуть переважну силу.
  4. В абсолютному режимі для G90 кінцева точка дуги кола є абсолютним значенням координати щодо 0 одиниць у їхніх власних напрямках як еталон. У відносному режимі для G91 кінцева точка дуги кола є додатковим значенням відносно початкової точки дуги кола.
  5. Незалежно від того, в абсолютному чи відносному режимі, координати центру кола I_J_(I_K_/J_K_) завжди є відносними координатами з початковою точкою як посиланням
  6. T – кількість повних кіл; шлях є довжиною дуги, коли T=0; це кількість повних кіл плюс довжина дуги, коли T є константою.
  7. На відміну від формату 1, формат 2 визначає дугу кола через початкову точку, кінцеву точку та радіус. Якщо вхідне значення праворуч від параметра R є додатним числом (R+), дуга кола є другорядною дугою менше 180 градусів; якщо вхідне значення праворуч від параметра R є від'ємним числом (R-), дуга кола є великою дугою більше ніж на 180 градусів.
  8. Наступні повні лінії являють собою шлях руху, коли G2 вибирає R+ і R-, а стрілки на дузі вказують напрямок руху.



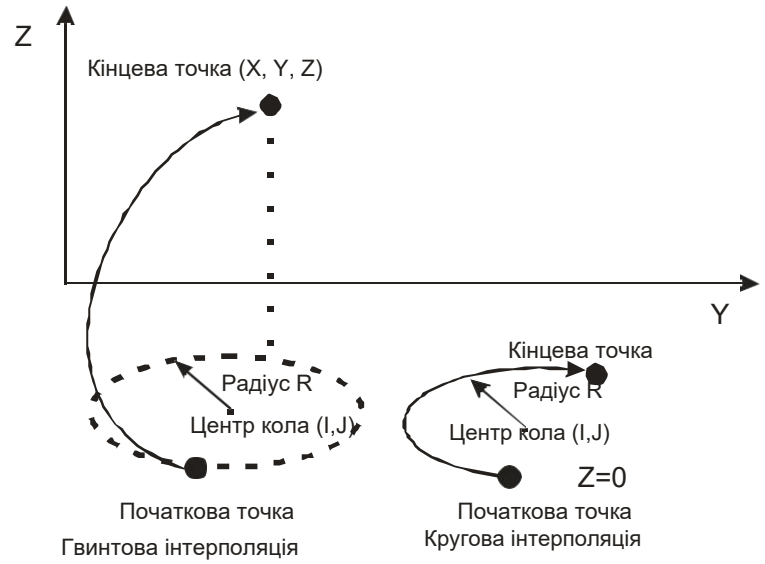
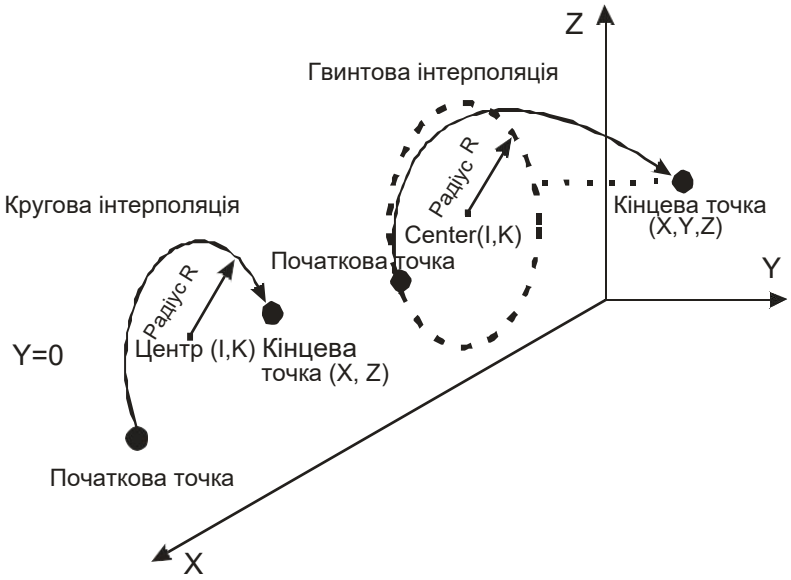
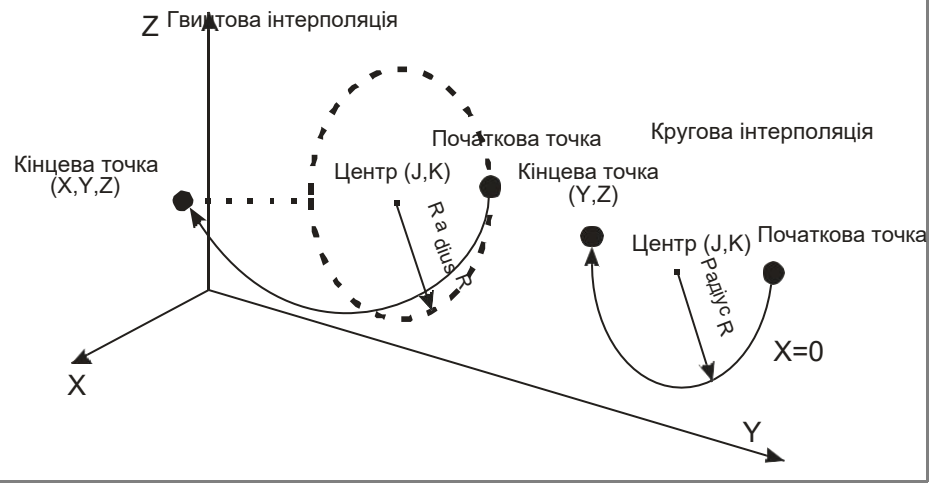
- Координатні співвідношення на різних площинах:



а) дуга кола на площині XY б) XZ плоска дуга кола в) дуга кола на площині YZ

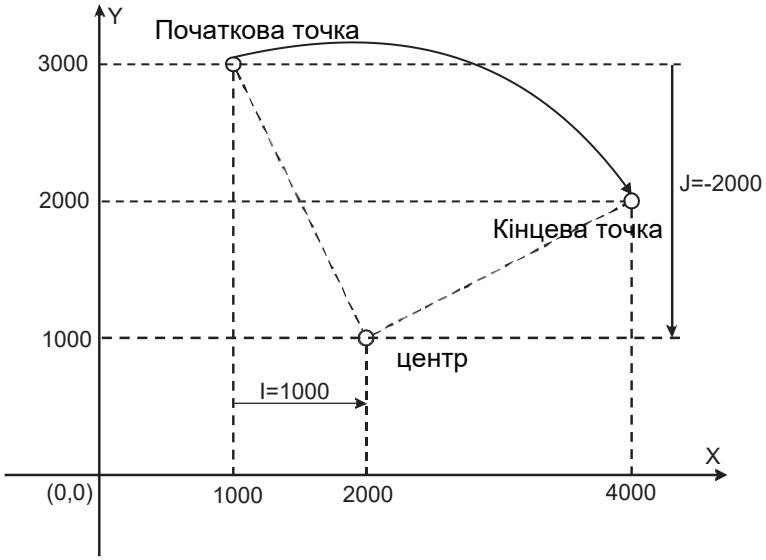
Зверніть увагу на співвідношення між координатними площинами та I, J, K. Тільки два з I, J і K існують в одній інструкції дуги кола. Які два існують, залежить від відповідної площини, наприклад, на площині XY з'являються лише I та J.

Координатну площину можна встановити за допомогою G17, G18 і G19. Кругові та спіральні шляхи руху для G2 на різних координатних площинах показані нижче.

код G	функція	Фігура шляху
G17	Площина XY: якщо немає варіацій початкової та кінцевої точок, що відповідають координатам осі Z, траєкторія руху є круговою інтерполяцією. В іншому випадку це спіральна інтерполяція.	
G18	Літак XZ: Якщо немає варіацій для початкової та кінцевої точок, що відповідають координатам осі Y, траєкторія руху є круговою інтерполяцією. В іншому випадку це спіральна інтерполяція.	
G19	Літак YZ: Якщо немає варіацій початкової та кінцевої точок, що відповідають координатам осі X, траєкторія руху є круговою інтерполяцією. В іншому випадку це спіральна інтерполяція.	

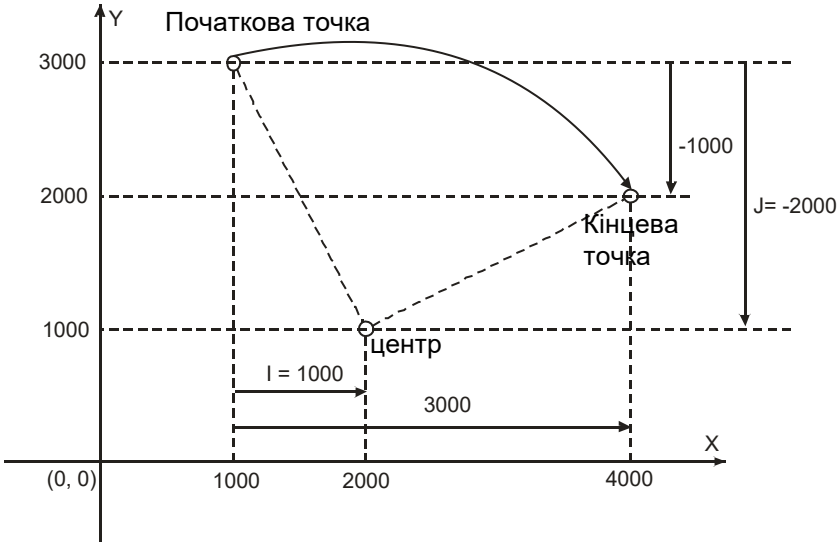
Приклад 1

- Задайте центр кола та проведіть кругову інтерполяцію в абсолютному режимі
  - Поточна позиція (1000, 3000), параметри осі: значення за замовчуванням, G-коди для виконання: N00 G90
    N01 G17
    N02 G2 X4000 Y2000 I1000 J-2000 E5000 F5000
  - Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



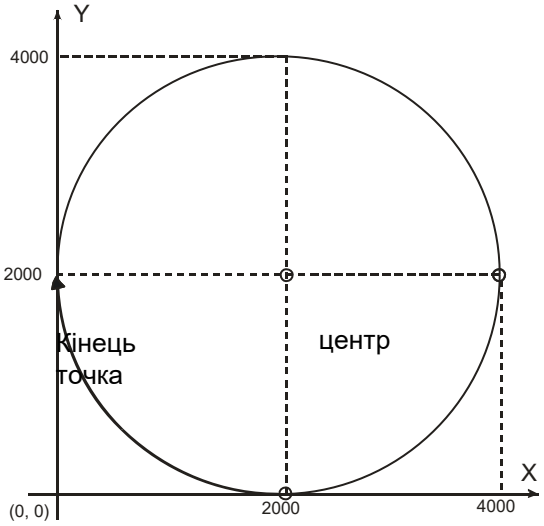
приклад 2:

- Задайте центр кола та проведіть кругову інтерполяцію у відносному режимі
  - Поточна позиція (1000, 3000), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати: N00 G91
    N01 G17
    N02 G2 X3000 Y-1000 I1000 J-2000
  - Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



приклад 3:

- Вкажіть центр кола та проведіть кругову інтерполяцію з T у відносному режимі
  - Поточна позиція (2000, 0), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:
    N00 G91
    N01 G17
    N02 G2 X-2000 Y2000 I0 J2000 T3
  - Після виконання G-кодів шлях дуги кола становить 3 кола плюс 1/4 кола завтовшки. Крива Y/X для всього процесу руху показана нижче:



Приклад 4: Початкова точка

- **Гвинтова інтерполяція з центром, визначеним площиною XY**

  Поточна позиція (0, 0), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:

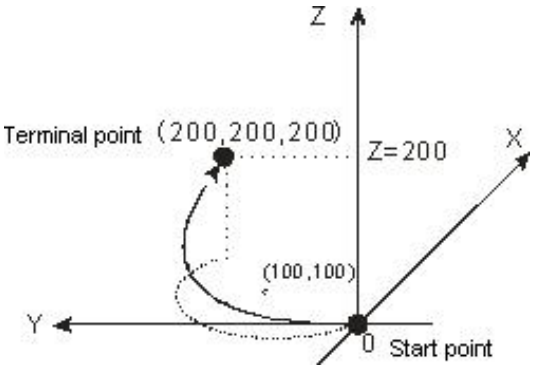
  N00 G17

  N01 G91

  N02 G2 X200 Y200 Z200 I100 J100 E+10000 E-20000 F1000

- **Пояснення до інструкції:**

  Поки G2 виконується, вісь розглядає 0 як початкову точку, а параметри координат осі – як кінцеві точки; дуга кола малюється за годинниковою стрілкою; остаточний шлях руху є гвинтовою кривою.
  Проекцією на площину XY є половина кола з центром (100,100).



Приклад 5:

- **Форма пропуску**

  G-коди, які потрібно виконати:

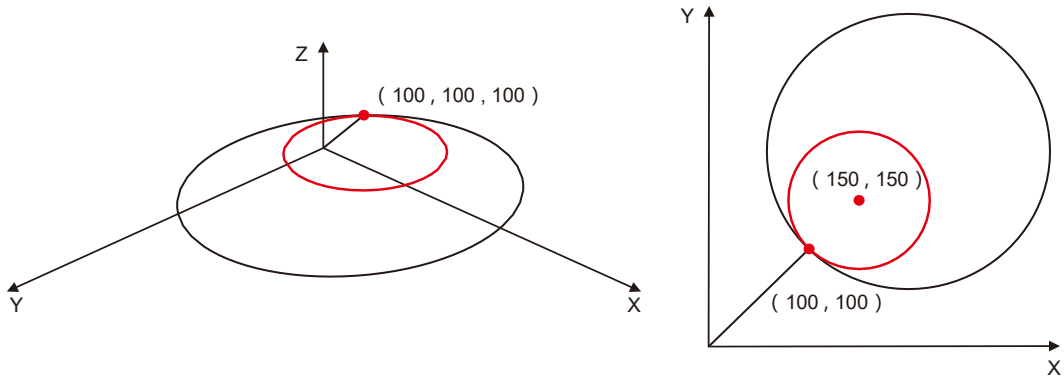
  N00 G0 X0 Y0 Z0

  N01 G1 X100 Y100 Z100

  N02 G2 I100 J100

  N03 G91

  N04 G2 I50 J50



- **Пояснення до інструкції:**

  1. Положення осі (100, 100, 100) після завершення виконання рядка інструкції N01;
  2. У рядку інструкції N02 є лише параметри I та J, а для інших пропущених значень параметрів вони базуються на дійсних значеннях останньої інструкції. Іншими словами, інструкція N02 еквівалентна: N02 X100 Y100 Z100 I100 J100. Отже, початкова і кінцева точки дорівнюють (100, 100, 100), а траєкторія руху є повним колом.
  3. Рядком інструкції N03 є G91, а наведені нижче координати відносні.

     Оскільки X, Y і Z пропущені в рядку інструкції N04, кінцева позиція є абсолютною позицією (100, 100, 100). Таким чином, шлях N04 є повним колом із початковою точкою (100,100,100) і кінцевою точкою (100,100,100), а центром є (150,150).

Приклад 6:

- Гвинтова інтерполяція з радіусом, визначеним площиною XY (Поточна позиція: 0)

  G-коди, які потрібно виконати:

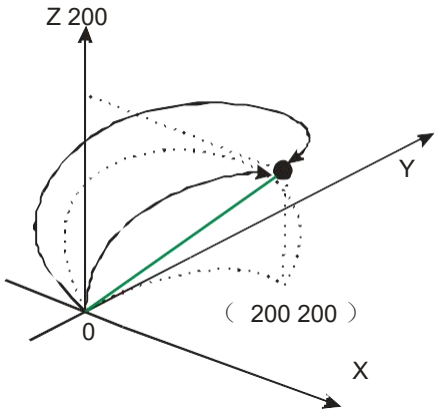
  N00 G2 X200 Y200 Z200 R-200

  N01 G0 X0 Y0 Z0

  N02 G2 X200 Y200 Z200 R200

- **Пояснення до інструкції:**

  Шлях руху є основною дугою, поки виконується перший код G2, і другорядною дугою, поки виконується другий код G2.



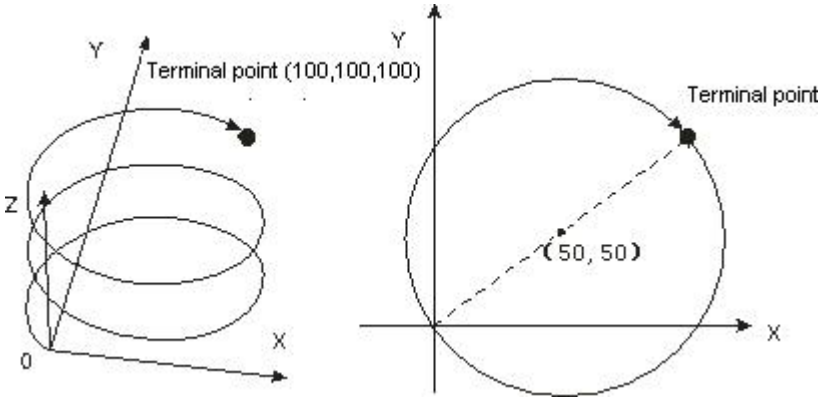
Приклад 7:

- Гвинтова інтерполяція з T і центром, вказаним на площині XY (Поточна позиція: 0) G-коди, які потрібно виконати:

  N00 G2 X100 Y100 Z100 I50 J50 T2

- **Пояснення до інструкції:**

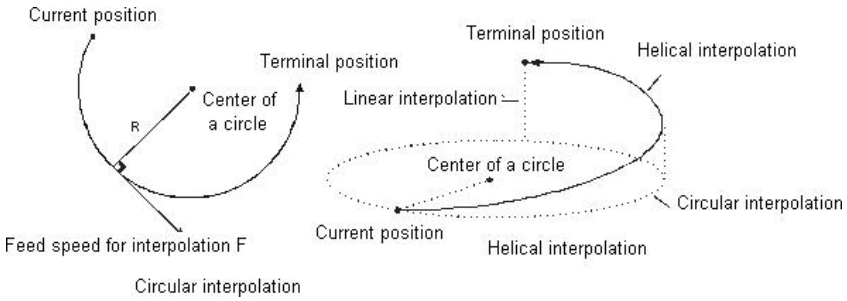
  Траєкторія руху — це гвинтова крива, а проекція на площину XY — повне коло з центром (50, 50).



11.6.4.6 G3 (кругова/спіральна інтерполяція проти годинникової стрілки)

- Пояснення функції:

  Кругова інтерполяція: різець виконує дугове різання обробленого об’єкта в напрямку проти годинникової стрілки зі швидкістю подачі, заданою параметром F, на дузі кола з фіксованим радіусом або фіксованим центром на вказаній площині.
  Гвинтова інтерполяція: фреза рухається проти годинникової стрілки по дузі кола зазначеної площини, яка є круговою інтерполяцією, і одночасно рухається у напрямку, вертикальному до вказаної площини, зі швидкістю подачі, заданою параметром F, що є лінійною інтерполяцією.



- Формат:
  Формат1: N_G3 X_Y_Z_A_B_C_P_Q_I_J_(I_K_/J_K_)T_ E_E_F_
  Формат2: N_G3 X_Y_Z_A_B_C_P_Q_R_T_ E_E_F_

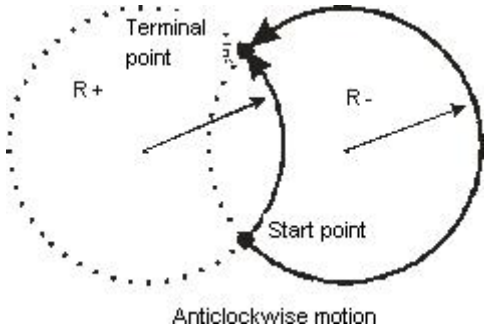
- Пояснення параметра:
  N_: Номер рядка коду G у програмі ЧПУ
  X_Y_Z_: вкажіть кінцеві позиції осей X, Y і Z, що відповідають кінцевій точці дуги кола; Одиниця: одиниця, тип даних: LREAL.
  A_B_C_P_Q_: Укажіть кінцеві позиції доданих осей, Одиниця: одиниця, тип даних: LREAL.
  I_J_: вкажіть координатне положення вказаного центру на площині XY, одиниця: одиниця, тип даних: LREAL. I_K_: вкажіть координатне положення зазначеного центру на площині XZ, одиниця: одиниця, тип даних: LREAL. J_K_: вкажіть координатне положення зазначеного центру на площині YZ, одиниця: одиниця, тип даних: LREAL.
  T_: вкажіть кількість повних кіл, одиниця: коло, тип даних: ULINT.
  E_: Вкажіть прискорення та уповільнення різця. Додатне число вказує на прискорення; від'ємне число вказує на уповільнення, одиниця: одиниця/секунда ^{2}, тип даних: LREAL. F: Укажіть швидкість подачі фрези, Одиниця: одиниця/секунда, тип даних: LREAL.

- Пояснення до інструкції:
  - Дві осі серед осей X, Y і Z роблять кругову інтерполяцію на площині, визначеній G17/G18/G19. 3-я вісь виконує лінійну інтерполяцію в напрямку, вертикальному до заданої площини.
  - Додані осі A, B, C, P і Q роблять лінійну інтерполяцію. Лінійна інтерполяція та кругова інтерполяція запускаються або припиняються одночасно.
  - I E, і F можна опустити. Якщо в області редагування ЧПУ є лише один рядок коду, а E, F пропущені, швидкість, прискорення та уповільнення визначаються параметрами групи осей. Якщо існує декілька рядків кодів, а E та F у коді G3 пропущені, швидкість, прискорення та уповільнення різця базуються на дійсних E та F у попередніх рядках кодів над рядком, де знаходиться G3. Якщо в попередніх рядках G-кодів не вказано E та F, «максимальна швидкість», «максимальне прискорення» та «максимальне уповільнення» серед параметрів групи осей матимуть переважну силу.
  - В абсолютному режимі для G90 кінцева точка дуги кола має абсолютні значення координат відносно одиниць 0 у відповідних напрямках як еталон. У відносному режимі для G91 кінцева точка дуги кола має додаткові значення відносно початкової точки дуги кола.
  - Незалежно від того, в абсолютному чи відносному режимі, координати центру кола I_J_(I_K_/J_K_) завжди є відносними координатами з початковою точкою в якості відліку.
  - T – кількість повних кіл; шлях є довжиною дуги, коли T=0; шлях - це відповідні повні кола плюс довжина дуги, коли T є константою.
  - На відміну від формату 1, формат 2 визначає довжину дуги кола через початкову точку, кінцеву точку та радіус. Якщо вхідне значення праворуч від параметра R є додатним числом (R+), дуга кола є другою дугою менше 180 градусів; якщо введене значення включено

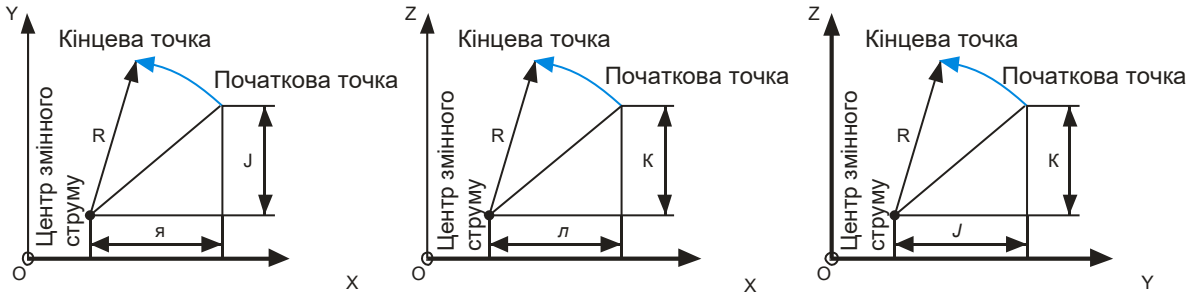
праворуч від параметра R є від'ємне число (R-), дуга кола є великою дугою більше 180 градусів.

- Наступні повні лінії являють собою шляхи руху, коли G3 вибирає R+ і R-, а стрілки на дугах вказують на напрямок руху.

11



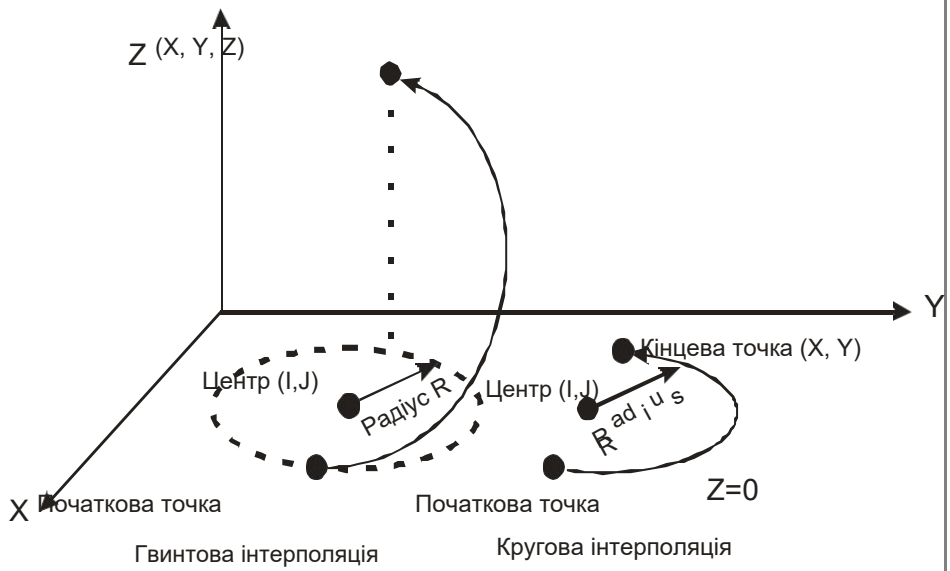
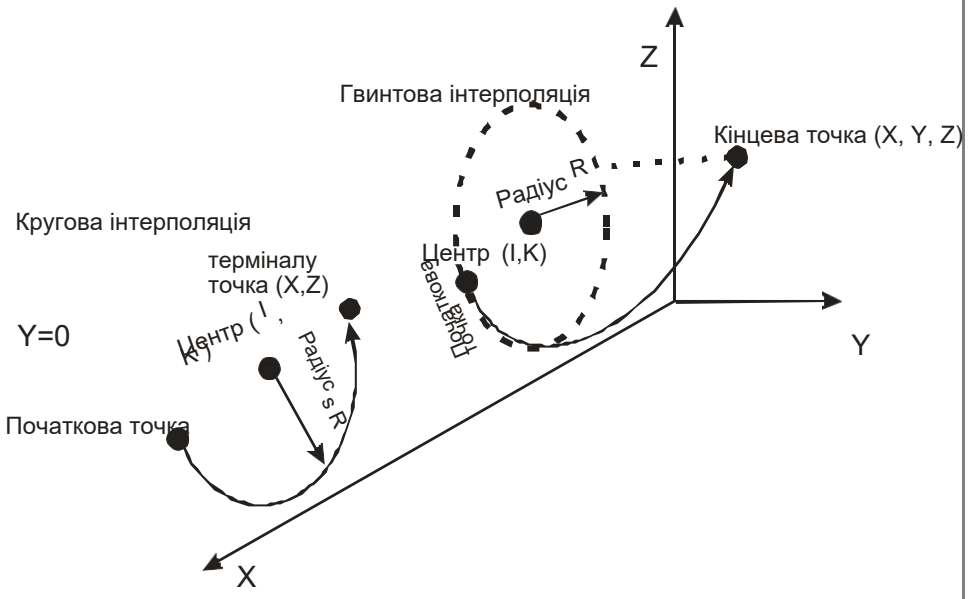
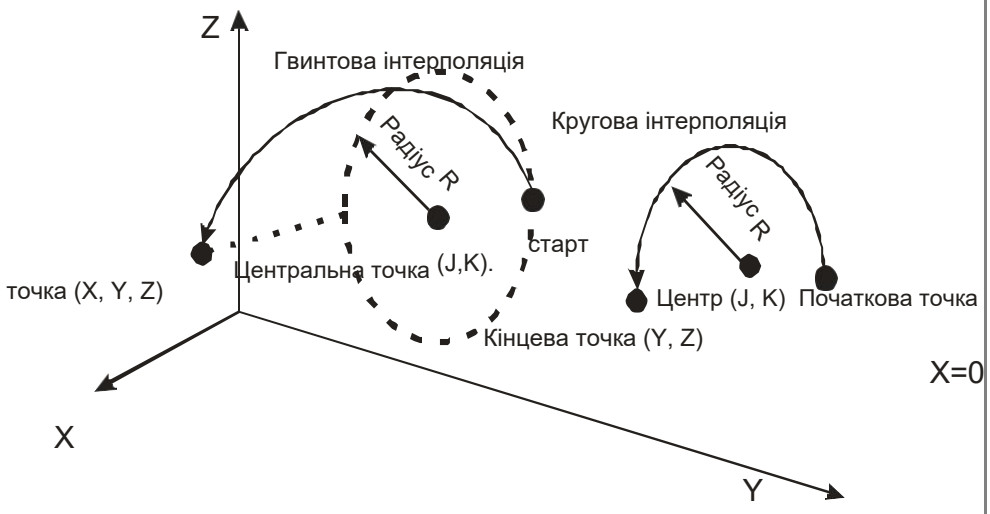
- Координатні співвідношення на різних площинах:



а) дуга кола на площині XY б) дуга кола на площині XZ С) дуга кола на площині YZ

Зверніть увагу на співвідношення між координатними площинами та I, J, K. Тільки два з I, J і K існують в одній інструкції дуги кола. Які два існують, залежить від відповідної площини, наприклад, на площині XY існують лише I та J.

Координатну площину можна встановити за допомогою G17, G18 і G19. Кругові та спіральні шляхи руху для G3 на різних координатних площинах показані нижче.

Г код	функція	Фігура шляху
G17	Площина XY: коли є певна варіація початкової точки та кінцева точка, що відповідає координатам осі Z, рух є спіральною інтерполяцією. В іншому випадку це кругова інтерполяція на площині XY.	
G18	Площина XZ: коли є варіація для початкова точка і кінцева точка відповідно до координат осі Y, рух є спіральною інтерполяцією. В іншому випадку це кругова інтерполяція на площині XZ.	
G19	Площина YZ: коли є варіація для початкова точка і кінцева точка відповідно до координат осі X, рух є спіральною інтерполяцією. В іншому випадку це кругова інтерполяція на площині YZ.	

Приклад 1

- Задайте центр кола та кругову інтерполяцію в абсолютному режимі

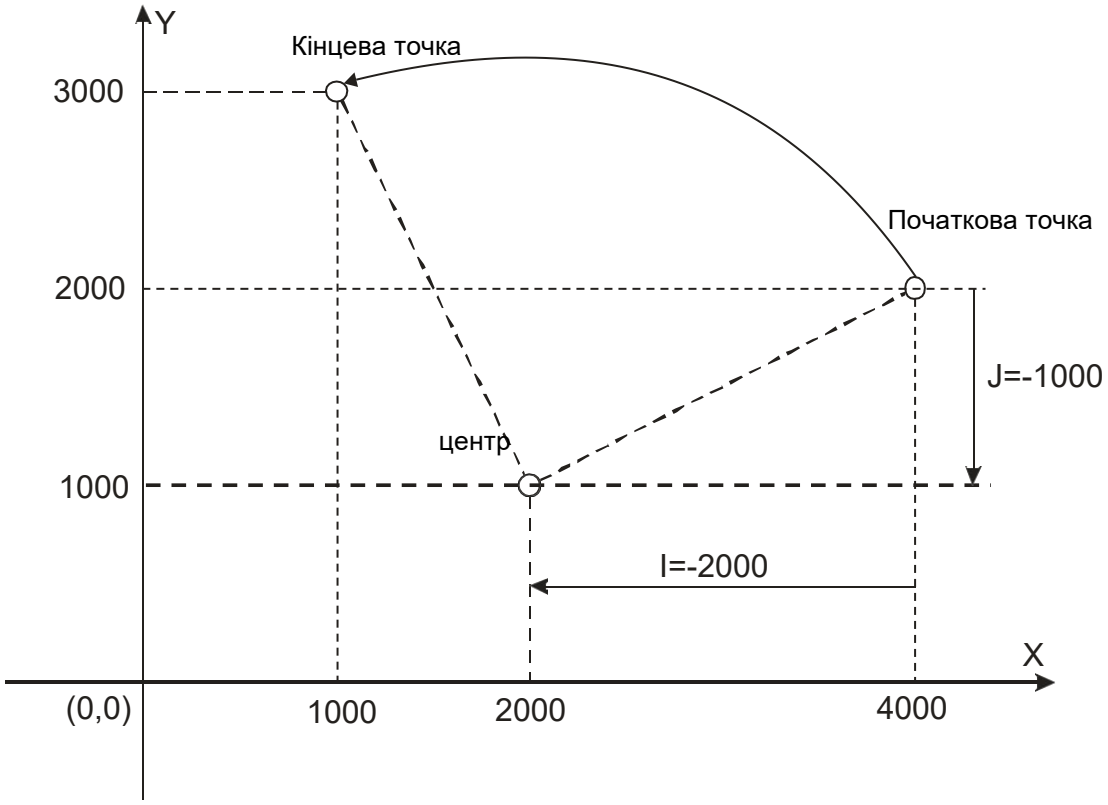
Поточна позиція (4000, 2000), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:

N0 G90

N1 G17

N2 G3 X1000 Y3000 I-2000 J-1000

Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



Приклад 2

- Задайте центр кола та кругову інтерполяцію у відносному режимі

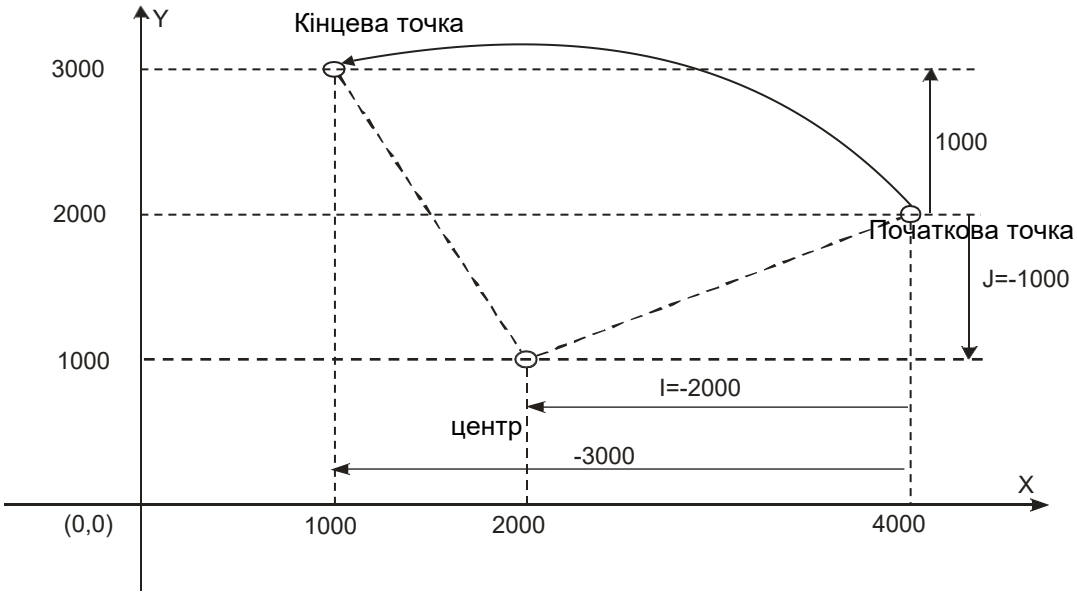
Поточна позиція (4000, 2000), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:

N0 G91

N1 G17

N2 G3 X-3000 Y1000 I-2000 J-1000

Після виконання G-кодів крива Y/X для всього процесу руху показана нижче:



Приклад 3

- Вкажіть центр кола та кругову інтерполяцію за допомогою T у відносному режимі

  Поточна позиція (2000, 0), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:

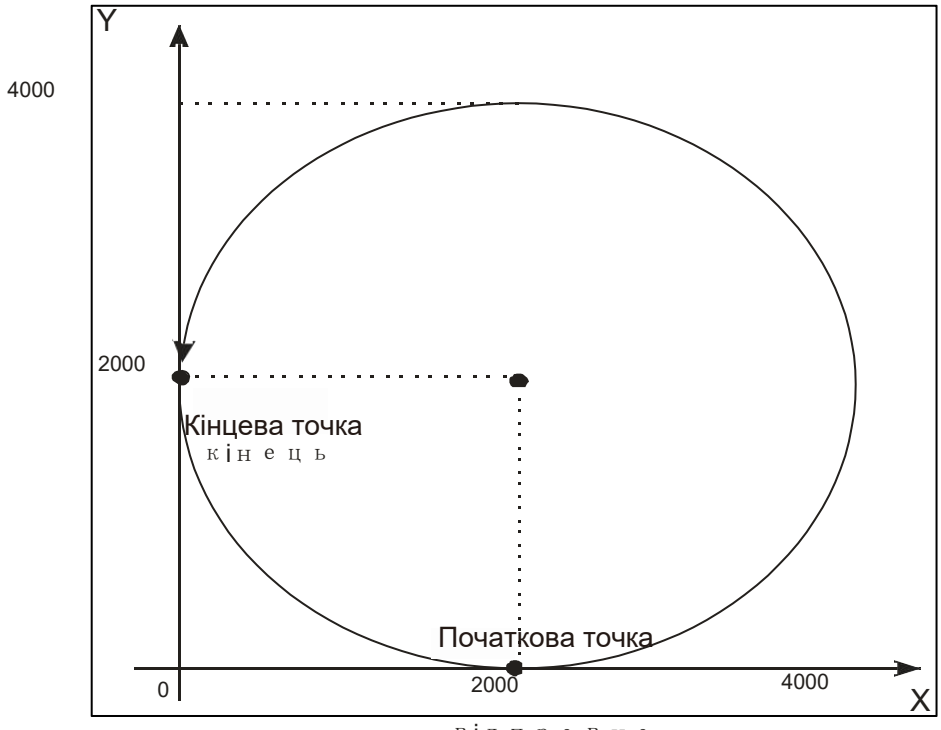
  N0 G91

  N1 G17

  N2 G3 X-2000 Y2000 I0 J2000 T3

- Пояснення до інструкції:

  Після виконання кодів G траєкторією руху є дуга на площині XY, а довжина дуги дорівнює (3+3/4) довжині кола).



Приклад 4

- Гвинтова інтерполяція з указаним центром кола

  Поточна позиція (0, 0), параметри осі: значення за замовчуванням, G-коди, які потрібно виконати:

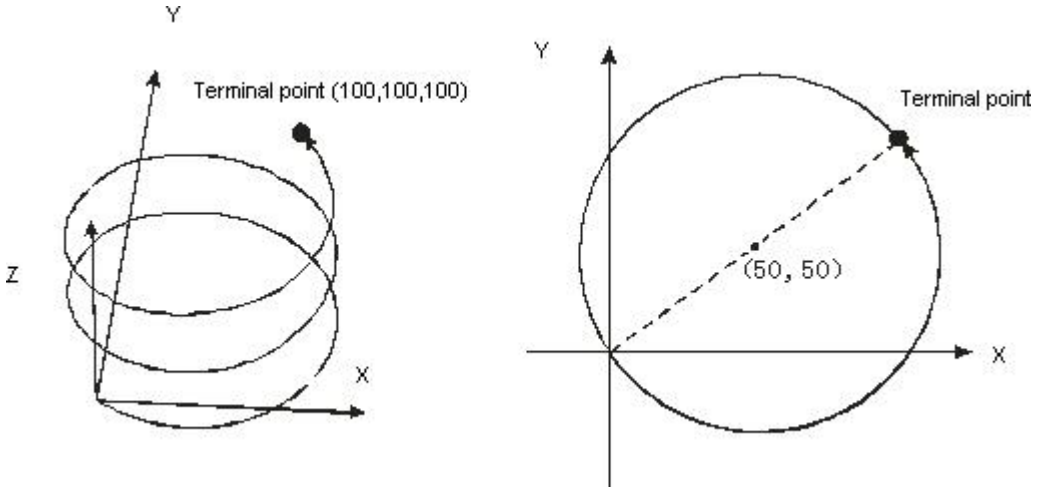
  N0 G17

  N1 G3 X100 Y100 Z100 I50 J50 T2

- Пояснення до інструкції:

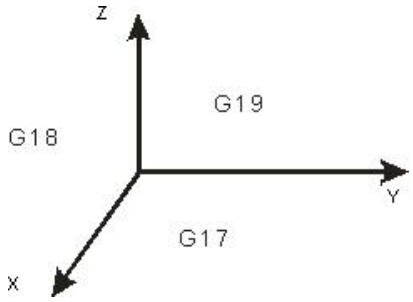
  Оскільки зміна осі Z становить 100, траєкторія руху є гвинтовою кривою, а проекція на площину XY є повним колом.

  Якщо немає змін для осі Z, траєкторією руху є дуга кола на площині XY з центром (50,50) і довжиною дуги, яка в 2,5 рази перевищує довжину окружності повного кола.



11.6.4.7 G17/G18/G19 (вкажіть кругову площину інтерполяції)

- функція:
  Три інструкції використовуються для вибору кругової інтерполяції або спіральної площини інтерполяції та не впливають на лінійну інтерполяцію.
  Під час виконання програми три робочі площини можна перемикати одна з одною.
  Якщо параметр площини не встановлено, початковим станом системи є площина XY (G17).
- Формат:
  N_G17 N_G18
  N_G19
- Пояснення параметра:
  N_: Номер рядка коду G у програмі ЧПУ
- Фігура площин зображена так:

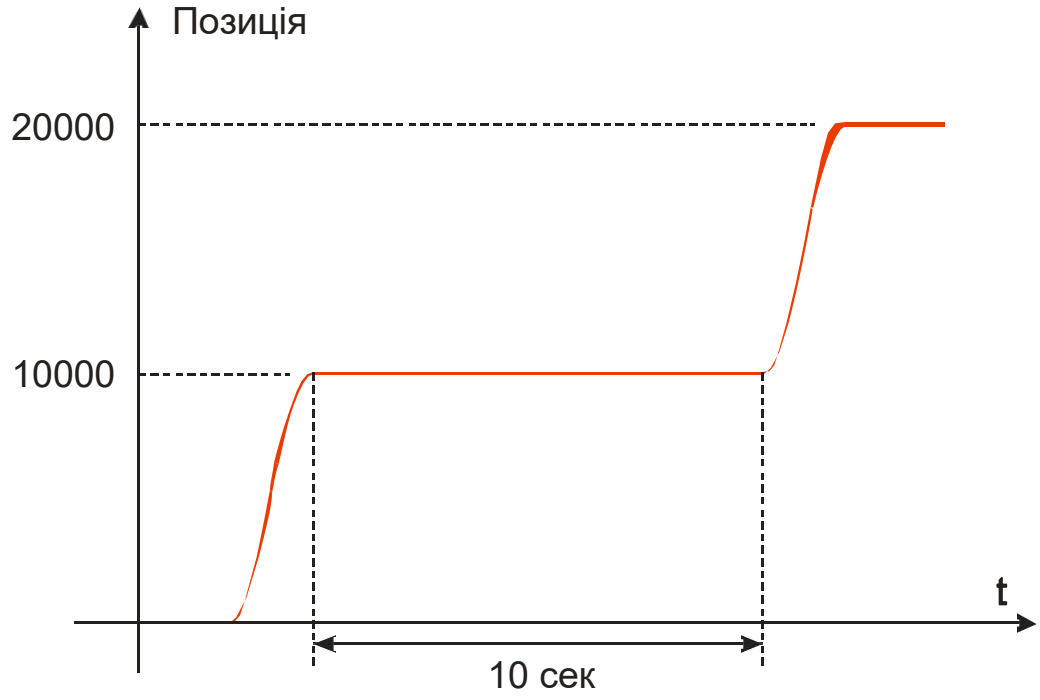


11.6.4.8 G4 (інструкція щодо зупинки)

- Функція: Інструкція щодо зупинки
- Формат: N_G4 K_
- Пояснення параметра:
  N_: Номер рядка коду G у програмі ЧПУ
  K_: Вкажіть час затримки, одиниця: секунда. Діапазон: 0,001 секунди ~100000 секунд
- Пояснення до інструкції:
  Після того, як токарний верстат завершить обробку на певному етапі, необхідно тимчасово зупинити рух фрези. У цей момент можна використовувати G4, щоб зупинити різець на певний період часу.

Приклад інструкції:

N00 G1 X10000
N01 G4 K10
N02 G1 X20000



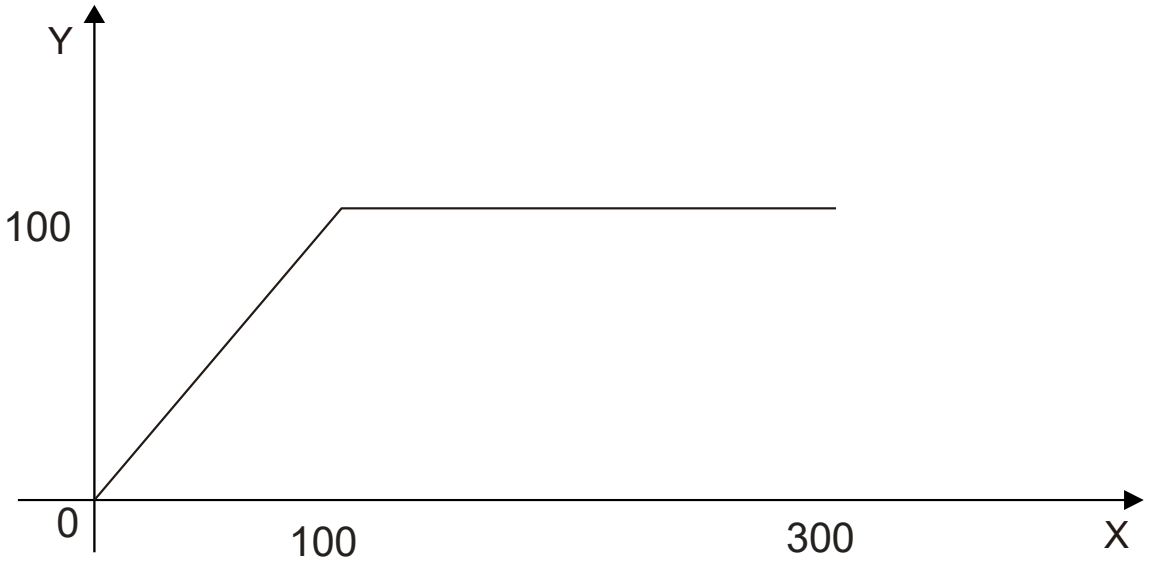
Після завершення виконання інструкції під номером N0 програма буде затримана на 10 секунд, після чого продовжить виконання інструкції під номером N2.

11.6.4.9 G50 (точна зупинка)

- Функція: змініть режим переходу на точну зупинку. І наступні режими переходу завжди є точною зупинкою. G51/G52 можна використовувати для перемикача в процесі виконання. Актуатор терміналу зменшить швидкість до 0 між кодами G.
- Формат: N_ G50
- Пояснення параметра:
  N_: Номер рядка G-коду в програмі NC

приклад

N0 G50
N1 G1 X100 Y100
N2 G1 X300 Y100



11.6.4.10 G51 (перехід по круговій траєкторії)

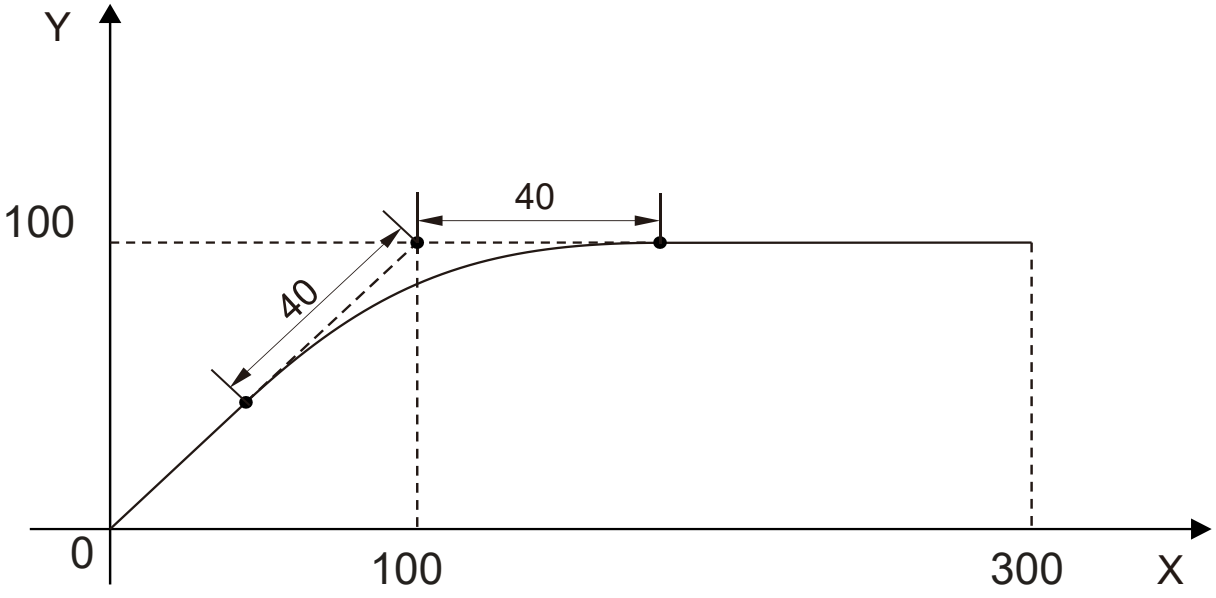
- Функція: змінити режим переходу на дуговий перехід. Наступні режими переходу завжди є дуговим переходом. G50/G52 можна використовувати для перемикання режимів переходу в процесі виконання. Термінальний привод не зменшуватиме свою швидкість між кодами G, а крива переходу є дугою. У цьому режимі швидкість першого коду G переважає, оскільки вся швидкість руху та F не можуть бути використані для зміни швидкості руху. Параметр VelOverride інструкції DMC_CartesianCoordinate можна встановити для керування швидкістю термінального приводу.
- Формат: N_ G51 D_
- Пояснення параметра:

  N_ : номер рядка коду G у програмі ЧПУ

  D_ : радіус дуги

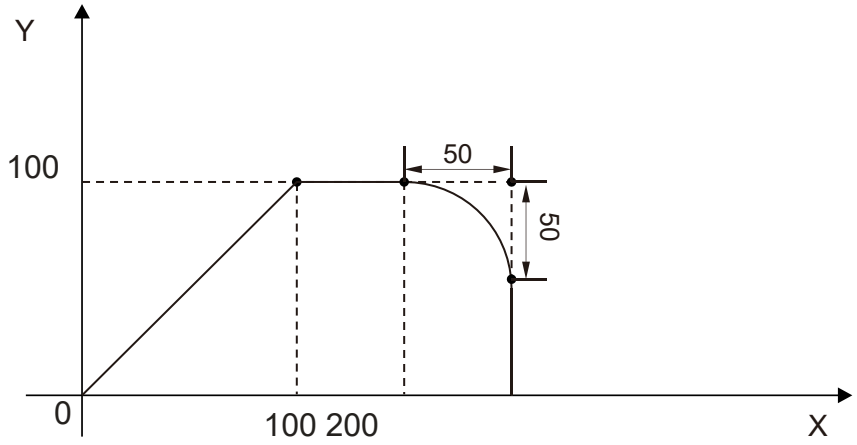
Приклад 1

N0 G51 D40
N1 G1 X100 Y100
N2 G1 X300 Y100



Приклад 2

```
N0 G1 X100 Y100
N1 G51 D50
N2 G1 X200 Y100
N3 G1 X200 Y0
```



11.6.4.11 G52 (плавний перехід контуру)

- Функція: щоб змінити режим переходу на плавний перехід траєкторії. Наступні режими переходу завжди є плавним переходом траєкторії. G50/G51 можна використовувати для перемикання режимів переходу в процесі виконання. Актуатор терміналу не зменшуватиме свою швидкість між кодами G. Він підходить для безперервної інтерполяції малих сегментів. У цьому режимі швидкість першого коду G переважає, оскільки вся швидкість руху та F не можуть бути використані для зміни швидкості руху. Параметр VelOverride інструкції DMC_CartesianCoordinate можна встановити для керування швидкістю термінального приводу.
- Формат: N_ G52
- Пояснення параметра:

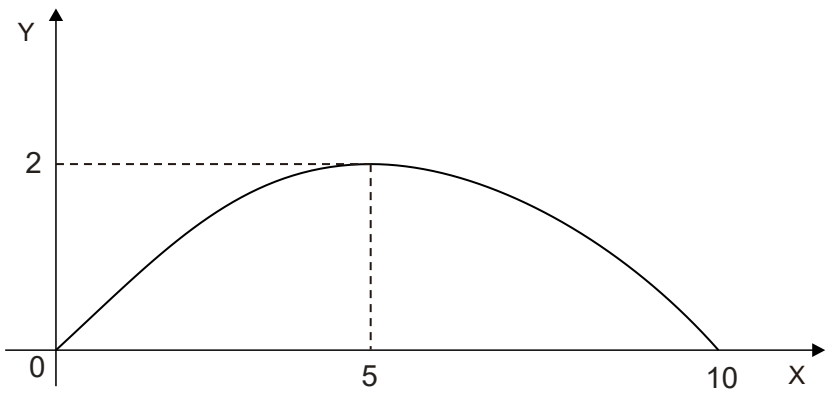
  N_ : номер рядка коду G у програмі ЧПУ

приклад

Намалюйте частину кривої sin.

N0 G52	N1 G1 X0 Y0 E5 E-5 F5
N2 G1 X0.1 Y0.06282151816	N3 G1 X0.2 Y0.1255810391
N4 G1 X0.3 Y0.1882166266	N5 G1 X0.4 Y0.2506664671
N6 G1 X0.5 Y0.3128689301	N7 G1 X0.6 Y0.3747626292
N8 G1 X0.7 Y0.4362864828	N9 G1 X0.8 Y0.4973797743
N10 G1 X0.9 Y0.5579822121	N11 G1 X1 Y0.6180339887
N12 G1 X1.1 Y0.6774758405	N13 G1 X1.2 Y0.7362491054
N14 G1 X1.3 Y0.7942957813	N15 G1 X1.4 Y0.8515585831
N16 G1 X1.5 Y0.9079809995	N17 G1 X1.6 Y0.9635073482
N18 G1 X1.7 Y1.018082832	N19 G1 X1.8 Y1.07165359
N20 G1 X1.9 Y1.124166756	N21 G1 X2 Y1.175570505
N22 G1 X2.1 Y1.225814107	N23 G1 X2.2 Y1.274847979
N24 G1 X2.3 Y1.322623731	N25 G1 X2.4 Y1.369094212
N26 G1 X2.5 Y1.414213562	N27 G1 X2.6 Y1.457937255
N28 G1 X2.7 Y1.500222139	N29 G1 X2.8 Y1.541026486
N30 G1 X2.9 Y1.580310025	N31 G1 X3 Y1.618033989

N32 G1 X3.1 Y1.654161149	N33 G1 X3.2 Y1.688655851
N34 G1 X3.3 Y1.721484054	N35 G1 X3.4 Y1.75261336
N36 G1 X3.5 Y1.782013048	N37 G1 X3.6 Y1.809654105
N38 G1 X3.7 Y1.835509251	N39 G1 X3.8 Y1.859552972
N40 G1 X3.9 Y1.881761538	N41 G1 X4 Y1.902113033
N42 G1 X4.1 Y1.920587371	N43 G1 X4.2 Y1.937166322
N44 G1 X4.3 Y1.951833524	N45 G1 X4.4 Y1.964574501
N46 G1 X4.5 Y1.975376681	N47 G1 X4.6 Y1.984229403
N48 G1 X4.7 Y1.991123929	N49 G1 X4.8 Y1.996053457
N50 G1 X4.9 Y1.999013121	N51 G1 X5 Y2
N52 G1 X5.1 Y1.999013121	N53 G1 X5.2 Y1.996053457
N54 G1 X5.3 Y1.991123929	N55 G1 X5.4 Y1.984229403
N56 G1 X5.5 Y1.975376681	N57 G1 X5.6 Y1.964574501
N58 G1 X5.7 Y1.951833524	N59 G1 X5.8 Y1.937166322
N60 G1 X5.9 Y1.920587371	N61 G1 X6 Y1.902113033
N62 G1 X6.1 Y1.881761538	N63 G1 X6.2 Y1.859552972
N64 G1 X6.3 Y1.835509251	N65 G1 X6.4 Y1.809654105
N66 G1 X6.5 Y1.782013048	N67 G1 X6.6 Y1.75261336
N68 G1 X6.7 Y1.721484054	N69 G1 X6.8 Y1.688655851
N70 G1 X6.9 Y1.654161149	N71 G1 X7 Y1.618033989
N72 G1 X7.1 Y1.580310025	N73 G1 X7.2 Y1.541026486
N74 G1 X7.3 Y1.500222139	N75 G1 X7.4 Y1.457937255
N76 G1 X7.5 Y1.414213562	N77 G1 X7.6 Y1.369094212
N78 G1 X7.7 Y1.322623731	N79 G1 X7.8 Y1.274847979
N80 G1 X7.9 Y1.225814107	N81 G1 X8 Y1.175570505
N82 G1 X8.1 Y1.124166756	N83 G1 X8.2 Y1.07165359
N84 G1 X8.3 Y1.018082832	N85 G1 X8.4 Y0.9635073482
N86 G1 X8.5 Y0.9079809995	N87 G1 X8.6 Y0.8515585831
N88 G1 X8.7 Y0.7942957813	N89 G1 X8.8 Y0.7362491054
N90 G1 X8.9 Y0.6774758405	N91 G1 X9 Y0.6180339887
N92 G1 X9.1 Y0.5579822121	N93 G1 X9.2 Y0.4973797743
N94 G1 X9.3 Y0.4362864828	N95 G1 X9.4 Y0.3747626292
N96 G1 X9.5 Y0.3128689301	N97 G1 X9.6 Y0.2506664671
N98 G1 X9.7 Y0.1882166266	N99 G1 X9.8 Y0.1255810391
N100 G1 X9.9 Y0.06282151816	N101 G1 X10 Y0



11.6.4.12 Код M

- Функція: Взаємодіє із загальними програмами.
- Формат: N_ M_ D_
- Пояснення параметра:

  N_ : номер рядка коду M у програмі ЧПУ

  M_ : номер коду M, діапазон: 0~99

  D_ : вихідний параметр, тип даних: LREAL.
- Пояснення до інструкції:

1. D_ можна опустити, і тоді параметри не виводяться після виконання коду M.
2. Два способи використання коду M: один полягає в написанні його поза рядком коду G; інший - написати його в рядку коду G.
3. Коли код M і код G не в одному рядку, наприклад, N0 M10 D10.02 ; N1 G1 X100 Y100. При досягненні рядка N0 виконання G-коду припиняється. Тим часом DMC_ReadMFunction є використовується для читання стану коду M, MFuntion має значення TRUE, а значення MFunctionValue дорівнює 10,02. Після скидання коду M за допомогою інструкції DMC_ResetMFunction виконання коду G продовжиться, а потім буде виконано рядок N1.
4. Коли код M і код G знаходяться в одному рядку, а код M можна розмістити лише після коду G, наприклад N0 G1 X100 M10 D10.02. Коли виконання надходить до рядка N0, виконується G1. Тим часом DMC_ReadMFunction використовується для читання стану коду M, MFuntion має значення TRUE, а значення MFunctionValue дорівнює 10,02. Наступне виконання продовжиться після завершення виконання G1, і інструкція скидання не буде потрібною для скидання коду M.

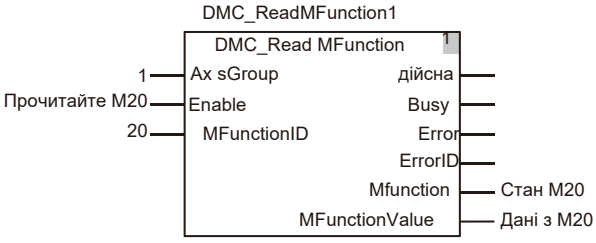
приклад

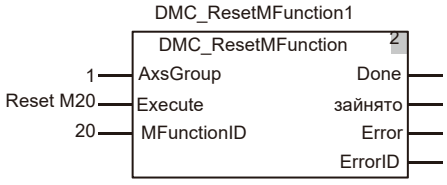
N0 G1 X100 Y50

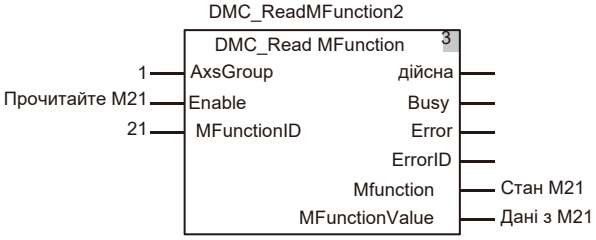
N1 M20

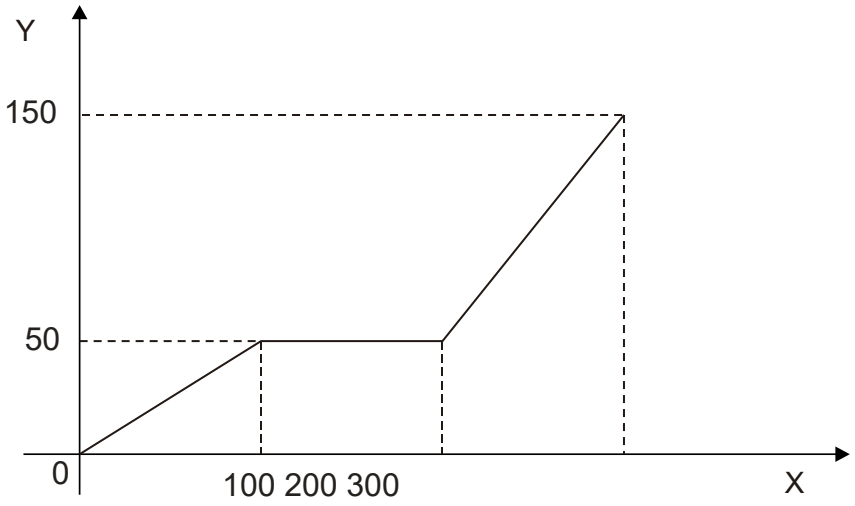
N2 G1 X200 Y50 M21 D3.14

N3 G1 X300 Y150









Коли дві змінні «Read M20» і «Read M21» мають значення TRUE, починається виконання коду G і виконавчий механізм терміналу зупиняється в положенні (100 · 50). На даний момент змінна «State of M20» має значення TRUE. Після

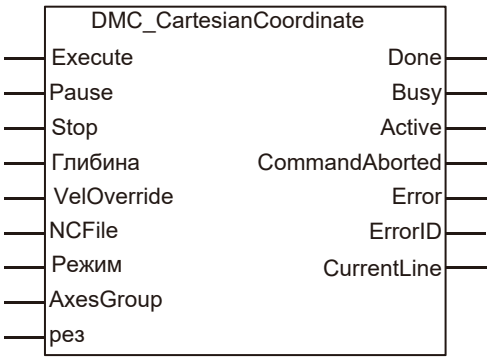
виконання інших дій завершено, змінна «Reset M20» змінюється на TRUE і код M скидається. Потім виконання коду G продовжується. Привід термінала починає рухатися до (200, 50). Тим часом змінна “state of M21” змінюється на TRUE, а значення змінної “Data from M21” дорівнює 3,14. У цей час привод терміналу не зупиниться. Після досягнення положення (200, 50) він продовжуватиме рухатися, доки не буде досягнуто положення (300, 150) і виконання не буде завершено.

11.6.5 Інструкції з коду G

11.6.5.1 DMC_CartesianCoordinate

FB/FC	Пояснення	Застосовна модель
FB	DMC_CartesianCoordinate використовується для керування рукою робота з декартовими координатами для виконання інтерполяції відповідно до G-коду.	AS516E-B AS532EST-B AS564EST-B

DMC_CartesianCoordinate_instance



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
Пауза	Коли змінюється *пауза* від FALSE до TRUE, робота-рука з декартовими координатами тимчасово припиняє виконання коду G.	BOOL	Правда чи брехня (FALSE)	
СТІЙ	Коли *значення Stop* змінюється з FALSE на TRUE, робота з декартовими координатами припиняє виконання коду G.	BOOL	Правда чи брехня (FALSE)	
VelOverride	Перевизначення швидкості (%)	LREAL	0~500 (0)	Коли *Execute* змінюється з FALSE на TRUE.
Глибина	Заповніть 1 для внутрішнього резервування	UINT	1	Коли *Execute* змінюється з FALSE на TRUE.
NCFile	Номер файлу NC	UINT	1~64 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Режим	Заповніть 0 для внутрішнього резервування	INT	0	Коли *Execute* зміни від

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
				FALSE на TRUE.
рез	Зарезервований			

11

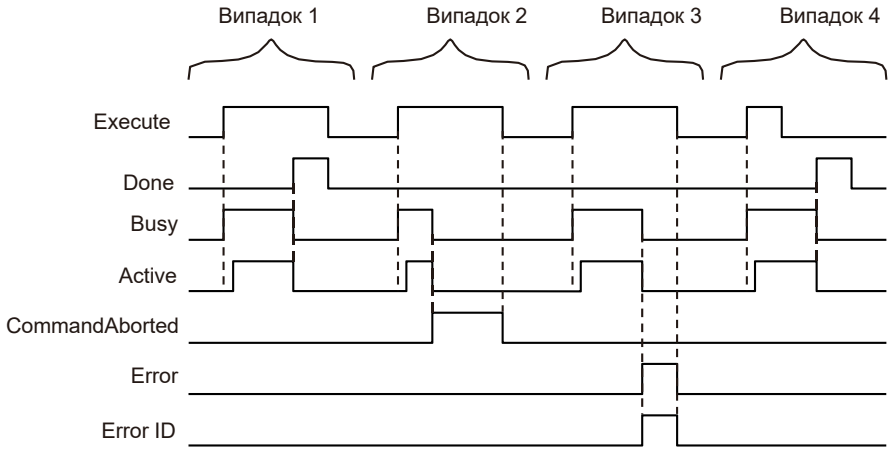
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли вісь контролюється інструкцією.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
CurrentLine	Номер рядка, де в даний момент виконується код G.	UDINT	

● **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли виконання коду G завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції. ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через крапку *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ TRUE, коли інструкція керує осями.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAborted	◆ TRUE, коли виконання інструкції переривається іншою інструкцією.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ *CommandAborted* змінюється на TRUE, коли виконання інструкції переривається іншою інструкцією після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один період *CommandAborted* змінюється на FALSE.
Error	◆ Вхідні параметри для інструкції є недопустимими або під час виконання інструкції сталася помилка.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а одним циклом пізніше *Active* змінюється на TRUE. Коли позиціонування завершено, *Done* змінюється на TRUE. Тим часом *«Зайнятий»* і *«Активний»* змінюються на FALSE.

Випадок 2 : якщо інструкція переривається MC_Stop або MC_Halt після того, як *Execute* змінюється з FALSE на TRUE, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли виникає помилка після того, як *Execute* змінюється з FALSE на TRUE, *Error* змінюється на TRUE, а *ErrorID* відображає відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 4 : коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

- **функція**

Інструкція DMC_CartesianCoordinate використовується для керування рукою робота з декартовими координатами для виконання інтерполяції відповідно до G-коду. Він застосовується до гравірувальних машин і швейних машин, які розглядають декартову координату робота як математичну модель. Прошивка V1.01 і вище підтримує цю функцію.

1. *Пауза* використовується для тимчасової зупинки виконання коду G. Після того, як *пауза* встановлена на TRUE, кінцевий привод зменшить свою швидкість до 0 відповідно до заданої швидкості уповільнення. Після завершення паузи *для паузи* встановлено значення FALSE. Термінальний привод прискорюватиметься із заданою швидкістю прискорення, доки не буде досягнуто цільової швидкості, і інтерполяція G-коду продовжиться.

1. *Зупинка* використовується для припинення виконання коду G. Якщо для *параметра «Зупинка»* встановлено значення «TRUE», виконавчий механізм терміналу негайно зупиниться, а тим часом інструкція «Готово» змінюється на «TRUE», а виконання G-коду припиняється .

1. *VelOverride* використовується для зміни швидкості кінцевого приводу в діапазоні від 0~500 з одиницею: %. «100» означає «100%». Швидкість кінцевого приводу після модифікації= Швидкість кінцевого приводу до модифікації x значення перевизначення.
Вісь буде прискорюватися або сповільнюватися до досягнення цільової швидкості після модифікації відповідно до швидкості прискорення та швидкості уповільнення коду G, який виконується в даний момент.

4. *NCFile* використовується для визначення номера файлу NC для виконання. Номер є ідентифікатором файлу ЧПУ, вбудованого в програмне забезпечення для програмування.

4. *AxesGroup* вказує номер групи осей, яка має виконати G-код.

4. Перед використанням інструкції DMC_CartesianCoordinate осі в групі осей повинні бути в стані нерухомості. В іншому випадку виникне помилка у виконанні інструкції.

7. Перед тим, як інструкція керуватиме рухом осей, осі слід відрегулювати за допомогою одноосьових інструкцій до положення, з якого спочатку починається виконання G-коду. Потім DMC_ AddAxisToGroup використовується для додавання окремих осей до групи осей. Після цього DMC_SetG0Para та DMC_SetG1Para використовуються для встановлення відповідних параметрів G0 та G1/G2/G3. Нарешті виконується інструкція DMC_CartesianCoordinate для керування осями для інтерполяції вздовж шляху, запланованого за допомогою G-кодів.

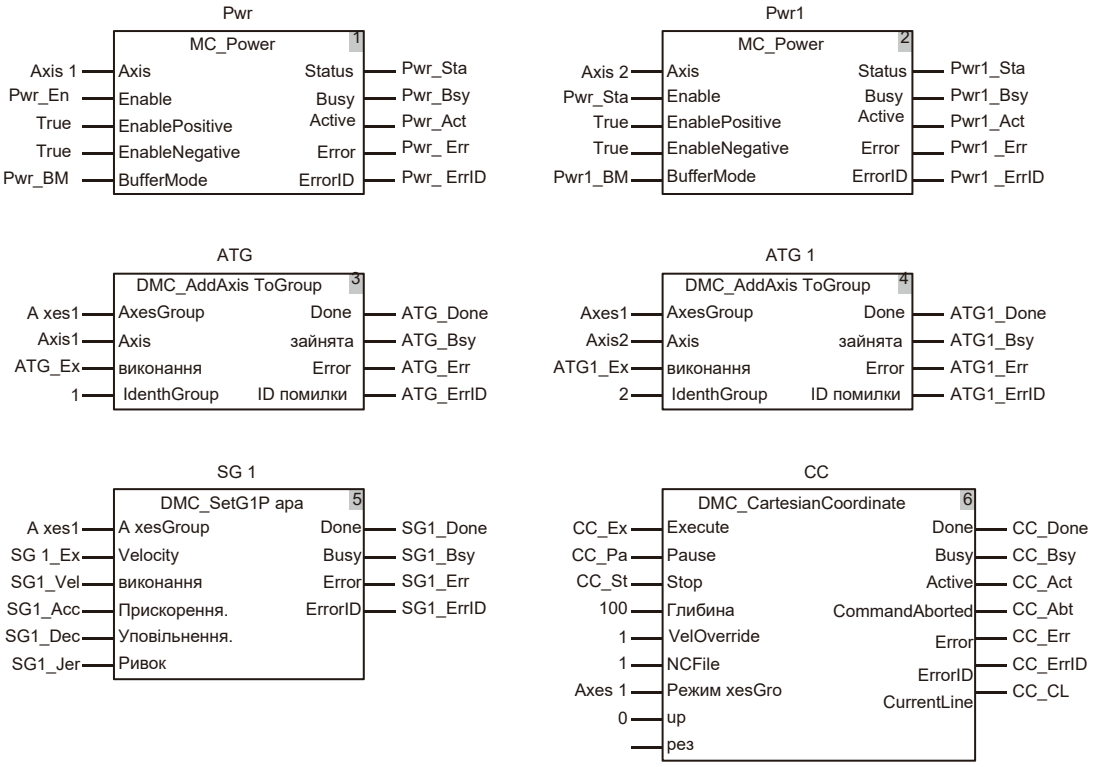
Приклад програмування

1. Таблиця змінних і програма

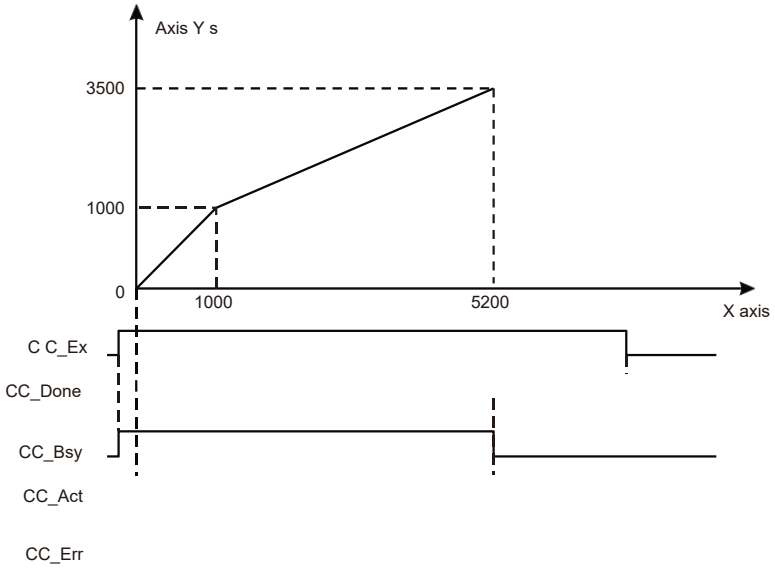
Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_En	BOOL	FALSE
Pwr_BM	MC_Buffer_Mode	0
Pwr_Sta	BOOL	
Pwr_Bsy	BOOL	
Pwr_Act	BOOL	
Pwr_Err	BOOL	
Pwr_ErrID	WORD	
Pwr1	MC_Power	
Axis2	USINT	2
Pwr1_BM	MC_Buffer_Mode	0
Pwr1_Sta	BOOL	
Pwr1_Bsy	BOOL	
Pwr1_Act	BOOL	
Pwr1_Err	BOOL	
Pwr1_ErrID	WORD	
ATG	DMC_AddAxisToGroup	
Axes1	USINT	1
ATG_Ex	BOOL	FALSE
ATG_Done	BOOL	
ATG_Bsy	BOOL	
ATG_Err	BOOL	
ATG_ErrID	WORD	
ATG1	DMC_AddAxisToGroup	
ATG1_Ex	BOOL	FALSE
ATG1_Done	BOOL	
ATG1_Bsy	BOOL	
ATG1_Err	BOOL	
ATG1_ErrID	WORD	
SG1	DMC_SetG1Para	
SG1_AG	USINT	1
SG1_Ex	BOOL	FALSE
SG1_Vel	LREAL	10000
SG1_Acc	LREAL	5000
SG1_грудень	LREAL	5000
SG1_Jer	LREAL	5000
SG1_Done	BOOL	

Ім'я змінної	Тип даних	Початкове значення
SG1_Bsy	BOOL	
SG1_Err	BOOL	
SG1_ErrID	WORD	
CC	DMC_CartesianCoordinate	
CC_Ex	BOOL	FALSE
CC_Pa	BOOL	FALSE
CC_St	BOOL	FALSE
CC_Done	BOOL	
CC_Bsy	BOOL	
CC_Act	BOOL	
CC_Abt	BOOL	
CC_Err	BOOL	
CC_ErrID	WORD	
CC_CL	UDINT	

код G:
N0 G1 X1000 Y1000
N1 G1 X5200 Y3500



2. Крива руху

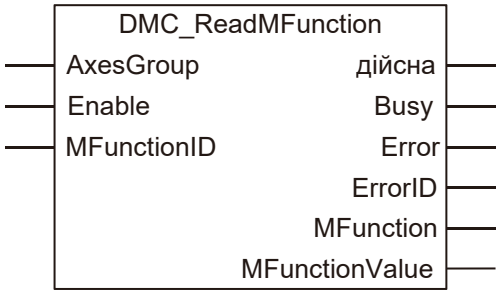


- Коли Pwr_En встановлено на TRUE, інструкція MC_Power виконується для ввімкнення двох осей. Потім для ATG_Ex і ATG1_Ex встановлюється значення TRUE, і виконується інструкція DMC_AddAxisToGroup для додавання Axis1 і Axis2 до групи осей Axes1. Після цього встановіть для SG1_Ex значення TRUE, щоб виконати інструкцію DMC_SetG1Para та встановити швидкість за замовчуванням G1/G2/G3. Нарешті, CC_Ex встановлюється на TRUE і виконується інструкція DMC_CartesianCoordinate для керування віссю 1 і віссю 2 для інтерполяції на основі шляху, запланованого за допомогою G-кодів.
- Коли CC_Ex змінюється з FALSE на TRUE, виконується інструкція DMC_CartesianCoordinate. У цьому ж циклі CC_Bsy змінюється з FALSE на TRUE. У другому циклі CC_Act змінюється з FALSE на TRUE, робот рухатиметься відповідно до шляху, запланованого за допомогою G-коду. Після завершення виконання коду G вихід CC_Done змінюється з FALSE на TRUE, а тим часом CC_Bsy і CC_Act змінюються з TRUE на FALSE.
- Коли CC_Ex змінюється з TRUE на FALSE, CC_Done змінюється з TRUE на FALSE.

11.6.5.2 DMC_ReadMFunction

FB/FC	Пояснення	Застосовна модель
FB	DMC_ReadMFunction використовується для **ResetMFunction**	AS516E-B AS532EST-B AS564EST-B

DMC_ReadMFunctio n_instance



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
MFunctionID	Номер коду M	USINT	0~99 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE

● **Вихідні параметри**

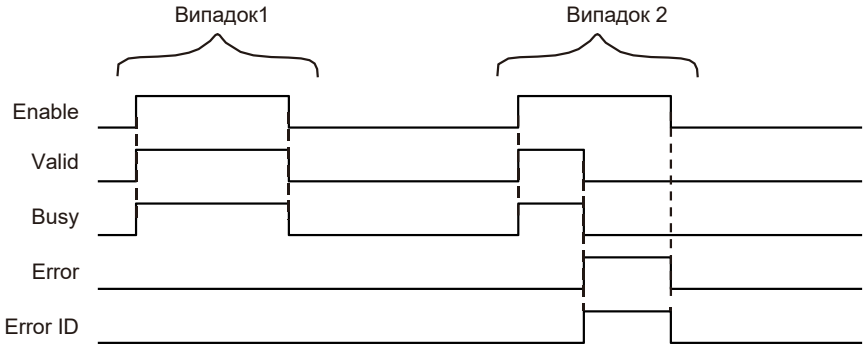
Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-
Мфункція	TRUE, коли код M досягається під час виконання коду G.	BOOL	TRUE / FALSE
MFunctionValue	Значення параметра коду M виводиться тут, коли вихід *MFunction* має значення TRUE.	LREAL	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли інструкція читає стан коду M.	◆ Коли *Enable* змінюється з TRUE на FALSE.
Busy	◆ Коли *Enable* змінюється на TRUE.	◆ Коли *Valid* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли вхідний параметр є недопустимим або виникає помилка	◆ Коли *Enable* змінюється з TRUE на FALSE

Ім'я	Час зміни на TRUE	Час зміни на FALSE
	під час виконання інструкції.	

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Valid* і *Busy* змінюються на TRUE одночасно. Коли *Enable* змінюється на FALSE, *Valid* і *Busy* змінюються на FALSE.

Випадок 2 : коли виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок. Тим часом *Busy* і *Valid* змінюються на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається.

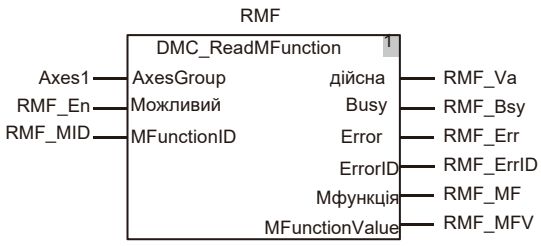
- **функція**

DMC_ReadMFunction використовується для читання стану коду M і даних з нього. Коли виконання G-коду досягає значення M-коду, встановленого інструкцією, *MFunction* змінюється на TRUE, а *MFunctionValue тим часом* виводить значення параметра після M-коду. Прошивка V1.01 і вище підтримує цю функцію.

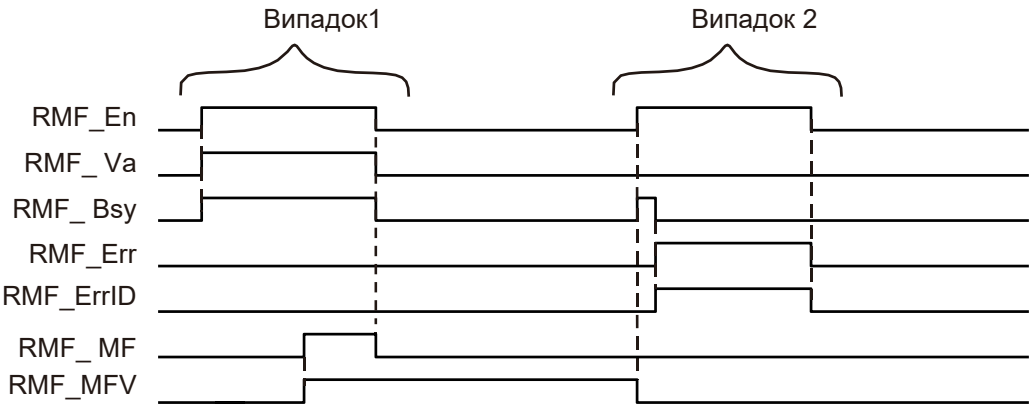
Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
RMF	DMC_ReadMFunction	
Axes1	USINT	1
RMF_En	BOOL	FALSE
RMF_MID	USINT	0
RMF_Va	BOOL	
RMF_Bsy	BOOL	
RMF_Err	BOOL	
RMF_ErrID	WORD	
RMF_MF	BOOL	
RMF_MFV	LREAL	



2. Графік часу



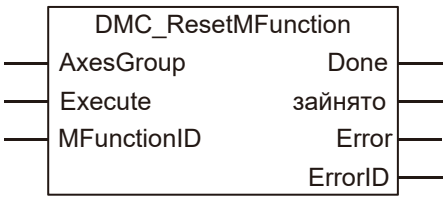
- Коли RMF_En змінюється з FALSE на TRUE і виконується інструкція DMC_ReadMFunction, RMF_Va і RMF_Bsy змінюються з FALSE на TRUE у першому циклі. Коли виконання G-коду досягає значення встановленого M-коду, RMF_MF змінюється з FALSE на TRUE, а тим часом RMF_MFV виводить значення параметра M-коду. Коли RMF_En змінюється з TRUE на FALSE, RMF_Va, RMF_Bsy і RMF_MF змінюються з TRUE на FALSE в тому самому циклі, а значення параметра коду M RMF_MFV не очищається.

- Коли RMF_En змінюється з FALSE на TRUE, виникає помилка у введенні параметра інструкції, RMF_Bsy змінюється з FALSE на TRUE у першому циклі, а значення параметра коду M, яке було виконано останнього разу, очищається. У другому циклі RMF_Err змінюється з FASLE на TRUE, тим часом як RMF_Bsy змінюється з TRUE на FALSE, а RMF_ErrID видає відповідні коди помилок. Коли RMF_En змінюється з TRUE на FALSE, RMF_Err змінюється з TRUE на FALSE, а значення RMF_ErrID тим часом очищається.

11.6.5.3 DMC_ResetMFunction

FB/FC	Пояснення	Застосовна модель
FB	Інструкція DMC_ResetMFunction скидає стан коду M.	AS516E-B AS532EST-B AS564EST-B

DMC_ResetMFunction_instance



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 ((Значення змінної має бути встановлено)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
MFunctionID	Номер коду M	USINT	0~99 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

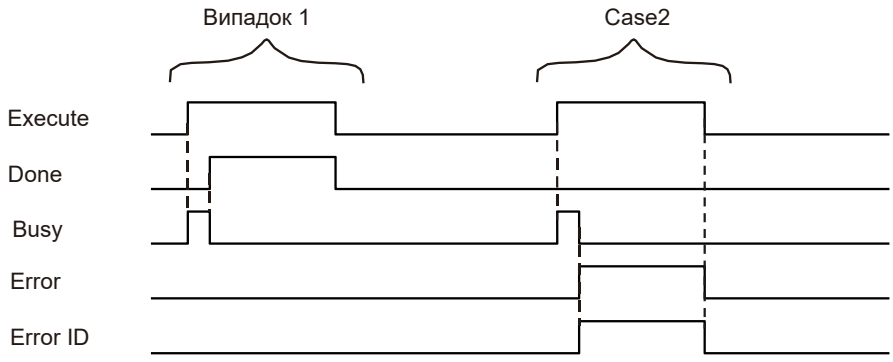
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

● **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Після завершення скидання M коду.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли вхідний параметр є недопустимим або під час виконання інструкції сталася помилка.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Зроблено* зміни на TRUE і *зайнятий* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

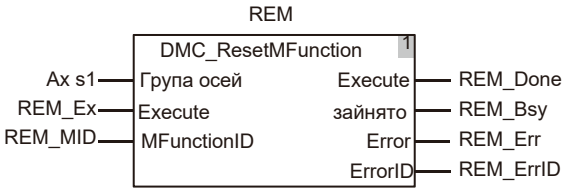
- **функція**

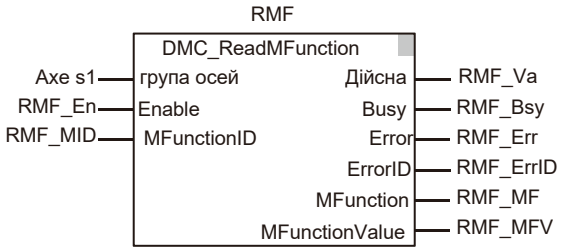
Інструкція DMC_ResetMFunction скидає стан коду M. Коли виконання G-коду досягає місця, де M-код встановлений інструкцією, вихідний стан M-коду є TRUE і виконується інструкція DMC_ResetMFunction. Вихідний стан M-коду скидається на FALSE, і виконання G-коду продовжиться. Прошивка V1.01 і вище підтримує цю функцію.

Приклад програмування

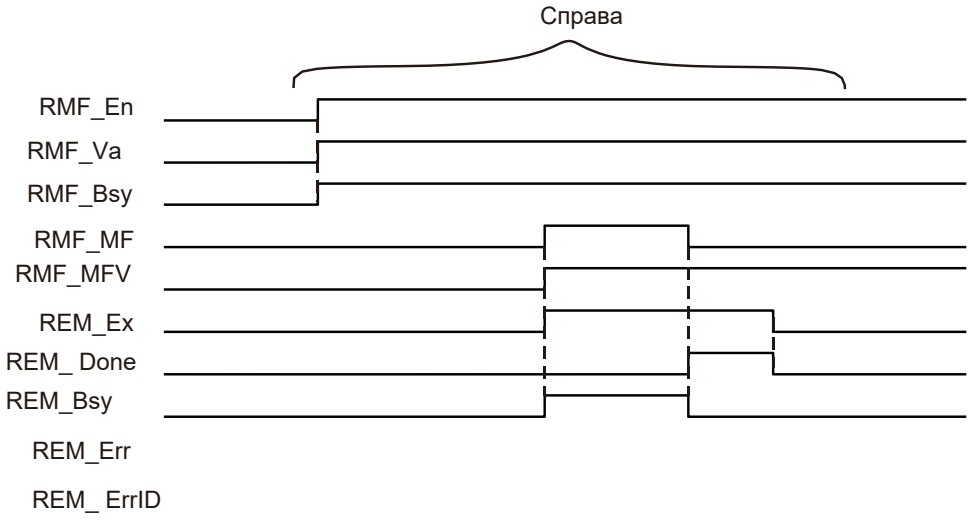
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
RMF	DMC_ReadMFunction	
Axes1	USINT	1
RMF_En	BOOL	FALSE
RMF_MID	USINT	0
RMF_Va	BOOL	
RMF_Bsy	BOOL	
RMF_Err	BOOL	
RMF_ErrID	WORD	
RMF_MF	BOOL	
RMF_MFV	LREAL	
REM	DMC_ResetMFunction	
REM_Ex	BOOL	FALSE
REM_MID	USINT	0
REM_Done	BOOL	
REM_Bsy	BOOL	
REM_Err	BOOL	
REM_ErrID	WORD	





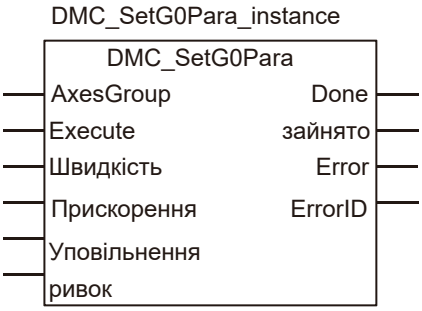
2. Графік часу



- Коли RMF_En змінюється з FALSE на TRUE, виконується інструкція DMC_ReadMFunction. У тому ж циклі RMF_Va і RMF_Bsy змінюються з FALSE на TRUE. Коли виконання G-коду досягає значення встановленого M-коду, RMF_MF змінюється з FALSE на TRUE, а тим часом RMF_MFV виводить значення параметра M-коду.
- Коли REM_Ex змінюється з FALSE на TRUE, виконується інструкція DMC_ResetMFunction. У цьому ж циклі REM_Bsy змінюється з FALSE на TRUE. У другому циклі REM_Done змінюється з FALSE на TRUE. Тим часом вихід RMF_MF DMC_ReadMFunction змінюється з TRUE на FALSE, а REM_Bsy змінюється з TRUE на FALSE. Коли REM_Ex змінюється з TRUE на FALSE, REM_Done змінюється з TRUE на FALSE у тому самому циклі.

11.6.5.4 DMC_SetG0Para

FB/FC	Пояснення	Застосовна модель
FB	DMC_SetG0Para використовується для встановлення швидкості, прискорення, уповільнення та ривка G0.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
швидкість	Укажіть цільову швидкість (одиниця: одиниця/секунда)	ARRAY [1..8] LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	ARRAY [1..8] LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	ARRAY [1..8] LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Вкажіть цільовий ривок. (Одиниця: одиниця/с ^{3})	ARRAY [1..8] LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

● **Вихідні параметри**

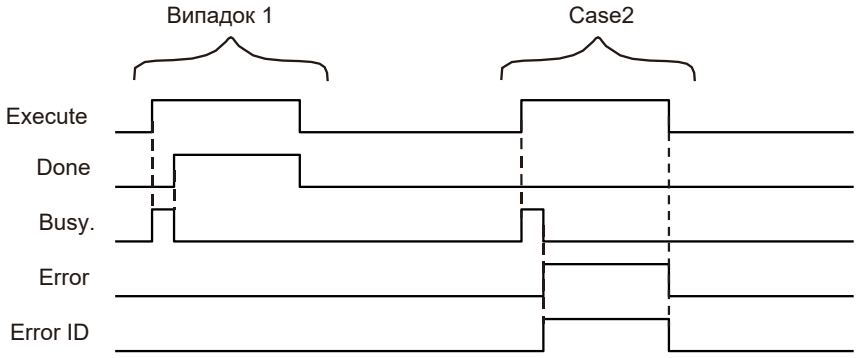
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли налаштування є	◆ Коли *Execute* змінюється з TRUE на FALSE після

Ім'я	Час зміни на TRUE	Час зміни на FALSE
	завершено.	виконання інструкції завершено.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли вхідний параметр є недопустимим або під час виконання інструкції сталася помилка.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Зроблено* змінюється на TRUE, а тим часом *Busy* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

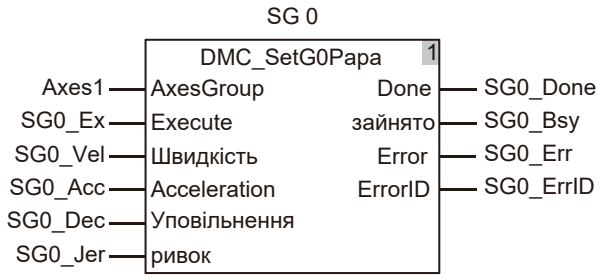
● **функція**

DMC_SetG0Para використовується для встановлення швидкості, прискорення, уповільнення та ривка G0. Коли G0 кодів G виконується, швидкість, прискорення, уповільнення та ривок виконуватимуться відповідно до параметрів, встановлених інструкцією DMC_SetG0Para. Прошивка V1.01 і вище підтримує цю функцію.

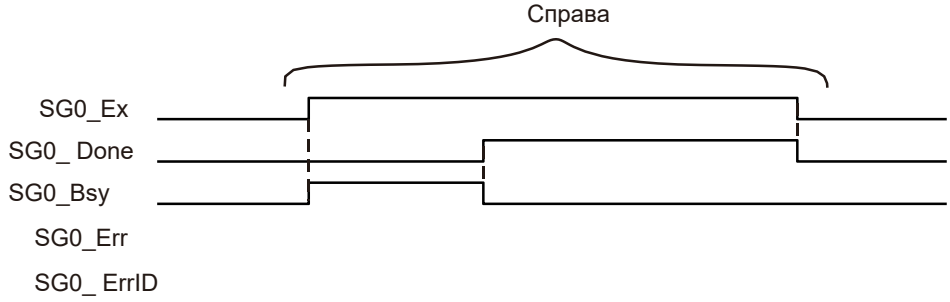
Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
SG0	DMC_SetG0Para	
Axes1	USINT	1
SG0_Ex	BOOL	FALSE
SG0_Vel	МАСИВ[1..8]LREAL	
SG0_Acc	МАСИВ[1..8]LREAL	
SG0_Dec	МАСИВ[1..8]LREAL	
SG0_Jer	МАСИВ[1..8]LREAL	
SG0_Done	BOOL	
SG0_Bsy	BOOL	
SG0_Err	BOOL	
SG0_ErrID	WORD	



2. Графік часу



- ❖ Коли SG0_Ex змінюється з FALSE на TRUE, виконується інструкція DMC_SetG0Para. У цьому ж циклі SG0_Bsy змінюється з FALSE на TRUE. У другому циклі SG0_Done змінюється з FALSE на TRUE, а тим часом SG0_Bsy змінюється з TRUE на FALSE. Таким чином, G0 серед G-кодів буде виконано відповідно до швидкості, прискорення, уповільнення та ривка, які встановлені інструкцією.
- ❖ Коли SG0_Ex змінюється з TRUE на FALSE, SG0_Done змінюється з TRUE на FALSE.

11.6.5.5 DMC_SetG1Para

FB/FC	Пояснення	Застосовна модель
FB	Інструкція DMC_SetG1Para встановлює швидкість, прискорення, уповільнення та ривок за замовчуванням для G1/G2/G3.	AS516E-B AS532EST-B AS564EST-B

DMC_SetG1Para_примірник



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
швидкість	Укажіть цільову швидкість (одиниця: одиниця/секунда)	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Прискорення	Вкажіть цільову швидкість прискорення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Вкажіть цільову швидкість уповільнення. (Одиниця вимірювання: одиниця/с ^{2})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Вкажіть цільовий ривок. (Одиниця: одиниця/с ^{3})	LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

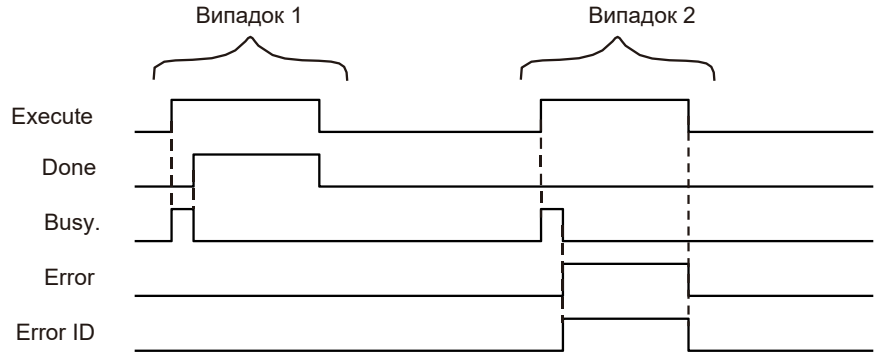
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Після завершення налаштування.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли вхідний параметр є недопустимим або під час виконання інструкції сталася помилка.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Зроблено* зміни на TRUE і *зайнятий* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

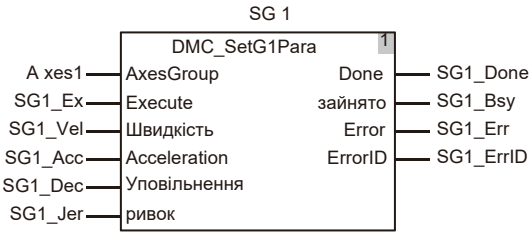
● **функція**

Інструкція DMC_SetG1Para встановлює швидкість, прискорення, уповільнення та ривок за замовчуванням для G1/G2/G3. Коли виконання кодів G досягає G1/G2/G3, G1/G2/G3 буде виконано відповідно до швидкості, прискорення та уповільнення, встановлених інструкцією, якщо значення E та F не вказані. В іншому випадку вони працюватимуть на основі заповнених значень E та F у кодах G. Прошивка V1.01 і вище підтримує цю функцію.

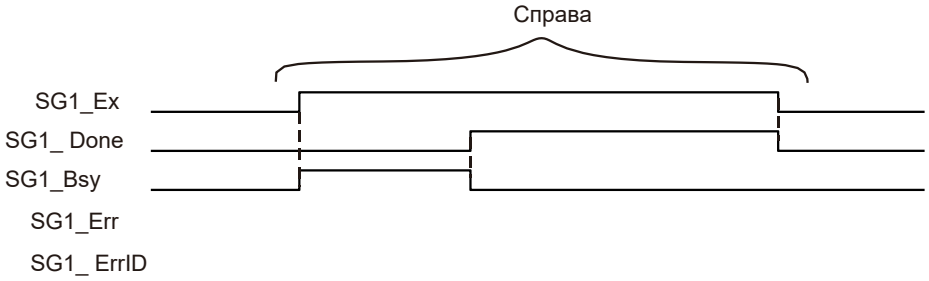
Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
SG1	DMC_SetG1Para	
Axes1	USINT	1
SG1_Ex	BOOL	FALSE
SG1_Vel	LREAL	5000
SG1_Acc	LREAL	1000
SG1_грудень	LREAL	1000
SG1_Jer	LREAL	1000
SG1_Done	BOOL	
SG1_Bsy	BOOL	
SG1_Err	BOOL	
SG1_ErrID	WORD	



2. Графік часу

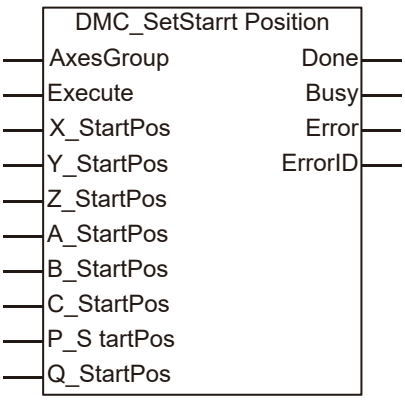


- ❖ Коли SG1_Ex змінюється з FALSE на TRUE, виконується інструкція DMC_SetG1Para. У цьому ж циклі SG1_Bsy змінюється з FALSE на TRUE. У другому циклі SG1_Done змінюється з FALSE на TRUE, а SG1_Bsy тим часом змінюється з TRUE на FALSE. Таким чином, G1 серед кодів G буде виконуватися відповідно до швидкості, прискорення, уповільнення та ривка, які встановлені інструкцією.
- ❖ Коли SG1_Ex змінюється з TRUE на FALSE, SG1_Done змінюється з TRUE на FALSE.

11.6.5.6 DMC_SetStartPosition

FB/FC	Пояснення	Застосовна модель
FB	Інструкція DMC_SetStartPosition задає початкові позиції осей G-коду.	AS516E-B AS532EST-B AS564EST-B

DMC_SetSta rt Position_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
X_StartPos	Укажіть початкове положення осі X.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Y_StartPos	Укажіть початкове положення осі Y.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Z_StartPos	Укажіть початкове положення осі Z.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
A_StartPos	Укажіть початкове положення осі A.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
B_StartPos	Укажіть початкове положення осі B.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
C_StartPos	Укажіть початкове положення осі C.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
P_StartPos	Вкажіть початкове положення осі P.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Q_StartPos	Укажіть початкове положення осі Q.	LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.

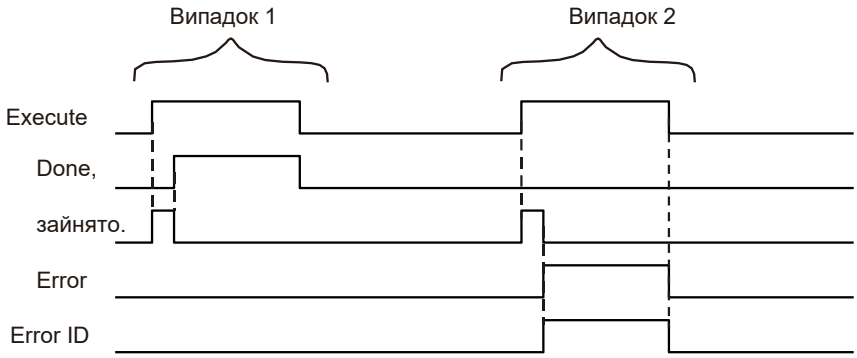
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	-

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Після завершення налаштування.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли вхідний параметр є недопустимим або під час виконання інструкції сталася помилка.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Зроблено* зміни на TRUE і *зайнятий* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

- **функція**

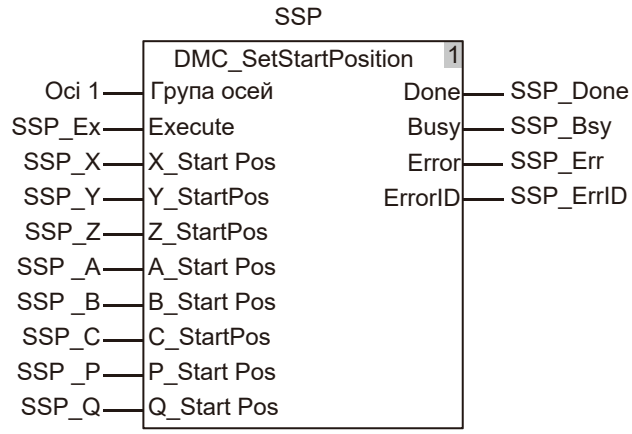
Інструкція DMC_SetStartPosition встановлює початкові позиції 8 осей G-коду. Після виконання інструкції рух починається з початкових позицій осей X, Y, Z, A, B, C, P і Q, визначених інструкцією. Наприклад, початкове положення осі X встановлено на 10000, а код G – G0 X1000. Отже, якщо Г

код для виконання, для осі X рух почнеться з позиції 10000 і досягне 1000. Прошивка V1.01 і вище підтримує цю функцію.
Початкові позиції не потрібно встановлювати за допомогою інструкції, якщо інструкція DMC_CartesianCoordinate використовується для виконання G-коду. Інструкція DMC_CartesianCoordinate автоматично встановить початкові позиції.

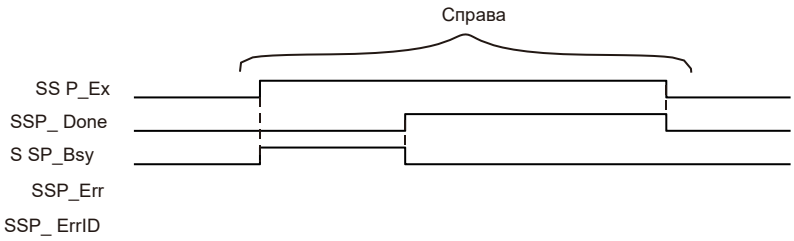
Приклад програмування

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
SSP	DMC_SetStartPosition	
Axes1	USINT	1
SSP_Ex	BOOL	FALSE
SSP_X	LREAL	1000
SSP_Y	LREAL	1000
SSP_Z	LREAL	1000
SSP_A	LREAL	1000
SSP_B	LREAL	1000
SSP_C	LREAL	1000
SSP_P	LREAL	1000
SSP_Q	LREAL	1000
SSP_Done	BOOL	
SSP_Bsy	BOOL	
SSP_Err	BOOL	
SSP_ErrID	WORD	



2. Графік часу



❖ Коли SSP_Ex змінюється з FALSE на TRUE, виконується інструкція DMC_SetStartPosition. У цьому ж циклі SSP_Bsy змінюється з FALSE на TRUE. У другому циклі SSP_Done змінюється з FALSE на TRUE, а тим часом SSP_Bsy змінюється з TRUE на FALSE. Отже

початкові позиції осей у G-кодах для руху — це початкові позиції, встановлені інструкцією.

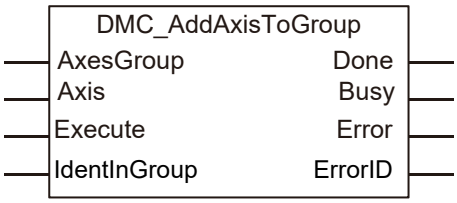
- ❖ Коли SSP_Ex змінюється з TRUE на FALSE, SSP_Done змінюється з TRUE на FALSE.

11.7 Інструкції групи Axes

11.7.1 DMC_AddAxisToGroup

FB/FC	Пояснення	Застосовна модель
FB	DMC_AddAxisToGroup використовується для додавання осі до групи осей.	AS516E-B AS532EST-B AS564EST-B

DMC_AddAxisTo Group_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Axis	Номер осі, яка додається до групи осей	USINT	Див. функції розділу 2.2. (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
IdentInGroup	Ідентифікаційний номер осі в групі осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

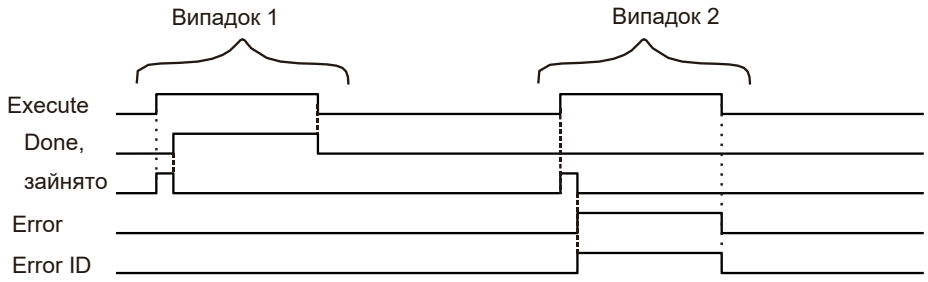
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли додавання осі до групи осей завершено.	◆ Коли *Execute* змінюється з TRUE на FALSE
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, а значення ErrorID очищається до 0, коли *Execute* змінюється з TRUE на FALSE.

- **функція**

DMC_AddAxisToGroup використовується для додавання осі до групи осей і встановлення номера осі в групі осей. Прошивка V1.01 і вище підтримує цю функцію.

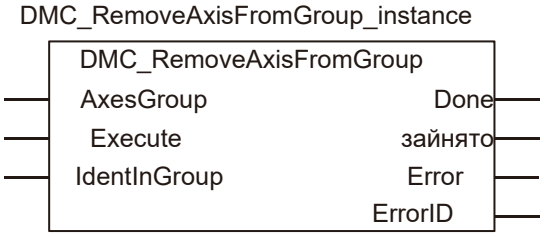
1. Коли *Done* для DMC_AddAxisToGroup змінюється на TRUE, це означає, що вісь успішно додано до групи осей. Зміна *Execute* на FALSE не може видалити вісь із групи осей. Щоб видалити вісь із групи осей, потрібна інструкція DMC_RemoveAxisFromGroup. Зверніться до розділу 11.7.2, щоб дізнатися більше про пояснення інструкції DMC_RemoveAxisFromGroup.

2. *IdentInGroup* — це ідентифікаційний номер осі в групі осей. Діапазон: 1~8. 1: вісь X, 2: вісь Y, 3: вісь Z, 4: вісь A, 5: вісь B, 6: вісь C, 7: вісь P і 8: вісь Q.

3. Інструкцію DMC_AddAxisToGroup можна виконати лише до ввімкнення групи осей. Якщо інструкція виконується після ввімкнення групи осей, в інструкції виникне помилка.

11.7.2 DMC_RemoveAxisFromGroup

FB/FC	Пояснення	Застосовна модель
FB	DMC_RemoveAxisFromGroup використовується для видалення осі з групи осей.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
IdentInGroup	Ідентифікаційний номер осі, яку потрібно видалити з групи осей	INT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.

- **Вихідні параметри**

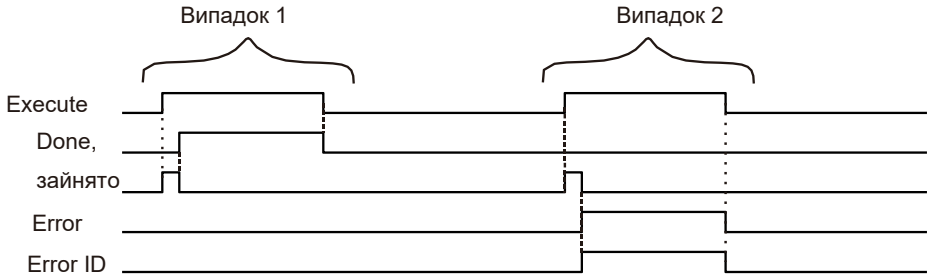
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли вісь видаляється з групи осей.	◆ Коли *Execute* змінюється з TRUE на FALSE
Busy	◆ Коли інструкція виконується.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли є помилка у вхідних параметрах інструкції або група осей не вимкнена, і *Execute* змінює значення FALSE на TRUE, *Busy* змінюється на TRUE і на один цикл пізніше *Error* змінюється на TRUE, а *ErrorID* показує коди помилок і тим часом *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0.

● **функція**

DMC_RemoveAxisFromGroup використовується для видалення осі з групи осей. Значення вхідного параметра *IdentInGroup* має бути в діапазоні від 1 до 8. Якщо діапазон перевищено, виникне помилка. Прошивка V1.01 і вище підтримує цю функцію.

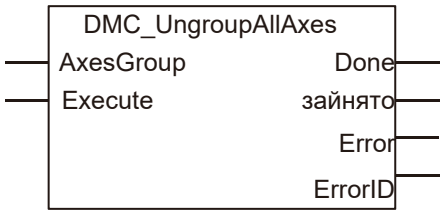
Коли інструкція виконується, помилка в інструкції виникне негайно, якщо групу осей увімкнено.

Інструкцію можна використовувати лише тоді, коли група осей не включена.

11.7.3 DMC_UngroupAllAxes

FB/FC	Пояснення	Застосовна модель
FB	DMC_UngroupAllAxes використовується для видалення всіх осей у групі осей.	AS516E-B AS532EST-B AS564EST-B

DMC_UngroupAllAxes_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	

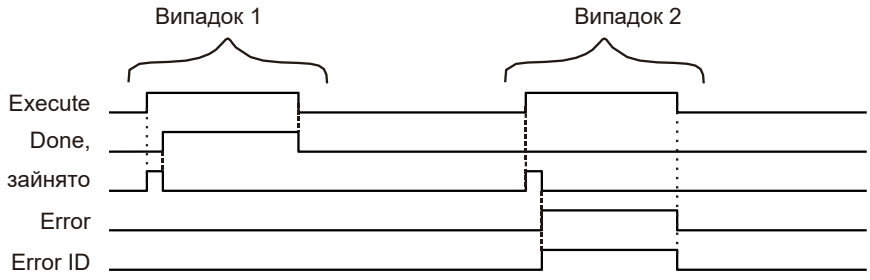
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли група осей розгрупована.	◆ Коли *Execute* змінюється з TRUE на FALSE
Busy	◆ Коли вказівка виконана.	◆ Коли *Execute* змінюється з TRUE на FALSE ◆ Коли *DONE* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE і через один цикл, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли є помилка у вхідних параметрах інструкції або група осей не вимкнена, і *Execute* змінює значення FALSE на TRUE, *Busy* змінюється на TRUE і на один цикл пізніше *Error* змінюється на TRUE, а *ErrorID* показує коди помилок і тим часом *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0.

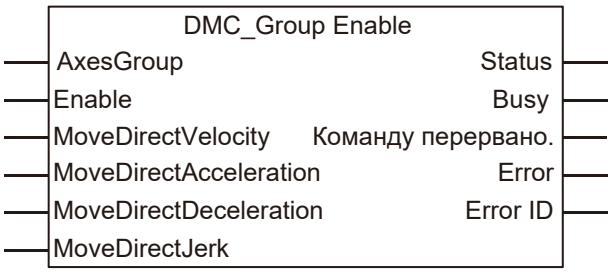
- **функція**

DMC_UngroupAllAxes використовується для видалення всіх осей у групі осей. Якщо групу осей увімкнено, помилка виникне одразу після використання інструкції. Прошивка V1.01 і вище підтримує цю функцію.

11.7.4 DMC_GroupEnable

FB/FC	Пояснення	Застосовна модель
FB	DMC_GroupEnable використовується для ввімкнення групи осей.	AS516E-B AS532EST-B AS564EST-B

DMC_Group Enable_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон (за умовчанням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Швидкість MoveDirectV	Швидкості осей для швидкого позиціонування	ARRAY [1..8] LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
Прискорення MoveDirectA	Прискорення осей для швидкого позиціонування	ARRAY [1..8] LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
Прискорення MoveDirectD	Уповільнення осей для швидкого позиціонування	ARRAY [1..8] LREAL	Позитивне число (потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
Функція MoveDirectJ	Ривки всіх осей для швидкого позиціонування	ARRAY [1..8] LREAL	Додатне число (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.

Примітка. Групою осей не можна керувати для виконання відповідних дій, якщо її не ввімкнено. Швидке позиціонування, лінійна інтерполяція та кругова інтерполяція не можуть бути виконані, якщо групу осей не ввімкнено.

- **Вихідний параметр**

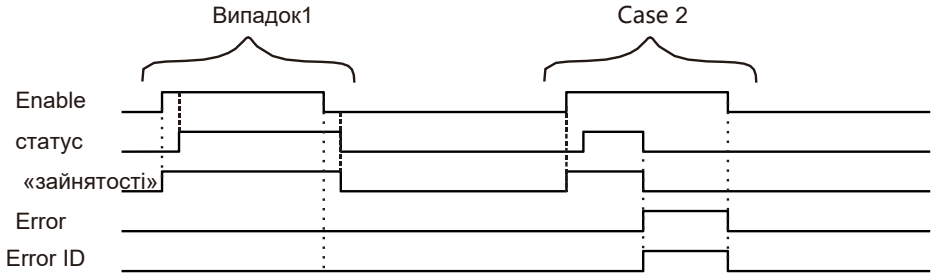
Назва параметра	функція	Тип даних	Допустимий діапазон
Status	TRUE, коли ввімкнено групу осей.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE

Назва параметра	функція	Тип даних	Допустимий діапазон
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE/FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Status	◆ Коли вказану групу осей увімкнено.	◆ Коли вказана група осей вимкнена ◆ Коли *помилка* змінюється на TRUE
Busy	◆ Коли інструкція виконується.	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється на TRUE
CommandAborted	◆ Коли виконання інструкції переривається.	◆ Коли *Enable* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Після успішного ввімкнення групи осей *Status* змінюється на TRUE. Після того, як *Enable* змінюється з TRUE на FALSE, а групу осей вимкнено, *Busy і Status* змінюються з TRUE на FALSE.

Випадок 2 : коли інструкцію DMC_GroupEnable переривається під час виконання, *CommandAborted* змінюється на TRUE, тим часом *Status* і *Busy* змінюються на FALSE, а група осей вимкнена. Коли *Enable* змінюється на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, *ErrorID* показує відповідні коди помилок, а *cmamyc* і *Busy* змінюються на FALSE. Коли *Enable* змінюється на FALSE, *Error* змінюється на FALSE, а значення в *ErrorID* скидається на 0.

- **функція**

DMC_GroupEnable використовується для ввімкнення або вимкнення групи осей. Прошивка V1.01 і вище підтримує цю функцію.

1. Перед виконанням інструкції DMC_GroupEnable усі осі в групі осей повинні перебувати в стані StandStill, щоб інструкція DMC_GroupEnable могла виконуватися нормально. Щоб увімкнути осі та перевести їх у стан StandStill, потрібно використовувати інструкцію MC_Power.

2. Коли осі перебувають у стані StandStill, *Status* змінюється на TRUE після того, як *Enable* встановлено на TRUE. Будь ласка, переконайтеся, що *статус* змінився на TRUE, перш ніж керувати рухом групи осей.
   Після зміни *статусу* на TURE осі переходять у стан дискретного руху. Щоб виконувати інші вказівки щодо руху, які можна виконати, лише коли осі перебувають у стані StandStill, для *Enable потрібно встановити значення FALSE.*

3. Після того, як *Enable* змінюється з TRUE на FALSE, групу осей вимикається, а осі перебувають у стані StandStill. *Я*кщо *Enable* змінюється з TRUE на FALSE, коли виконується інструкція групи осей, наприклад DMC_MoveDirectAbsolute, інструкція групи осей повідомить про помилку, усі осі в групі осей припиняться, а осі перебуватимуть у стані StandStill.

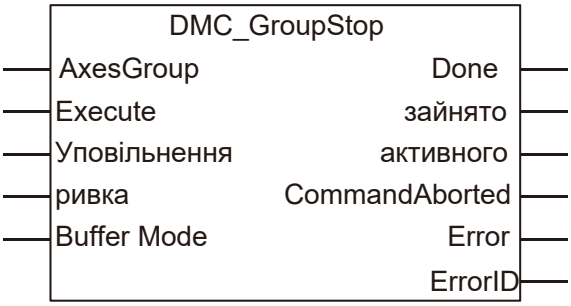
4. *Я*кщо одноосьова інструкція виконується під час виконання інструкції групи осей і значення *BufferMode* одноосьової інструкції встановлено на 0 (mcAbroting), одноосьова інструкція перериває інструкцію групи осей і групу осей буде вимкнено. *Я*кщо для значення *BufferMode* одноосьової інструкції встановлено ненульове число (1~5), одноосьова інструкція не виконуватиметься.

5. Перед виконанням інструкції DMC_GroupEnable усі осі в групі осей повинні перебувати в стані StandStill, щоб інструкція DMC_GroupEnable могла виконуватися нормально.

11.7.5 DMC_ GroupStop

FB/FC	Пояснення	Застосовна модель
FB	DMC_GroupStop використовується для зупинки руху поточної групи осей.	AS516E-B AS532EST-B AS564EST-B

DMC_GroupStop_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Вкажіть кількість груп осей, якими потрібно керувати	USINT	1~8 (Значення змінної має бути встановлено)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Уповільнення	Зарезервований	-	-	-
ривок	Зарезервований	-	-	-
BufferMode	Вкажіть поведінку під час виконання двох інструкцій. 0: перервати	MC_Buffer_Mode_	0: mcAborting	Коли *Execute* змінюється з FALSE на TRUE

Примітка:

1. *Уповільнення* та *ривок* зарезервовані, а їхні значення налаштувань недійсні.
2. *BufferMode* не підтримує жодного режиму, крім режиму 0 (mcAborting). Якщо вибрано будь-який інший режим, під час виконання інструкції виникне помилка.

- **Вихідні параметри**

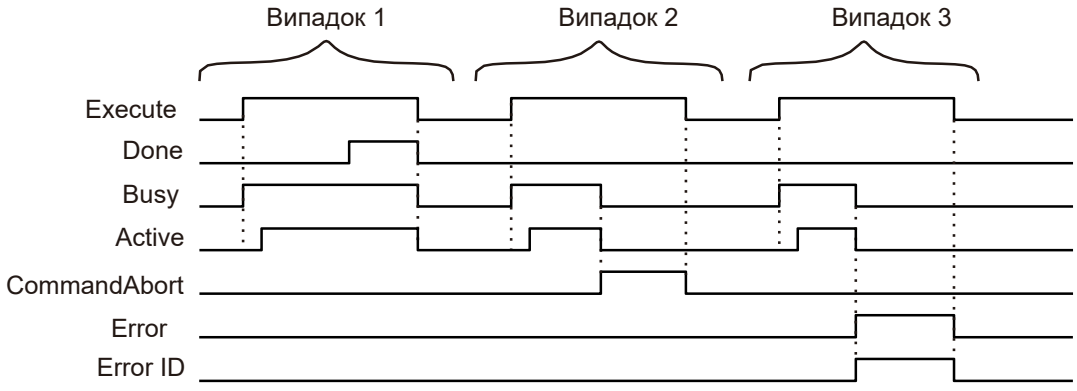
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли група осей керується.	BOOL	TRUE/FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE/FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE/FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного	WORD	

Назва параметра	функція	Тип даних	Допустимий діапазон
	код помилки.		

■ **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ При цьому рух групи осей припиняється.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли *помилка* змінюється з TRUE на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли *CommandAborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
Active	◆ Коли інструкція керує групою осей.	◆ Коли *Execute* змінюється з TRUE на FALSE. ◆ Коли *CommandAborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
CommandAborted	◆ Коли виконання інструкції переривається.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

■ **Діаграма часу оновлення вихідних даних**



Випадок 1: коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE, а через період *активний* змінюється на TRUE. Коли рух групи осей успішно зупинено, *Done* змінюється на TRUE, а *Busy* та *Active* залишаються TRUE. Коли *Execute* змінюється з TRUE на FALSE, *Done* , *Busy* і *Active* змінюються на FALSE.

Випадок 2: Коли інструкція DMC_GroupStop переривається під час виконання, *CommandAborted* змінюється на TRUE, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3: Коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на FALSE. *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0, коли *Execute* змінюється на FALSE.

- **функція**

  DMC_GroupStop використовується для зупинки руху поточної групи осей.

  1. Вхідні параметри *Deceleration* і *Jerk* інструкції DMC_GroupStop зарезервовані, а вхідні значення для них недійсні. Коли виконується інструкція DMC_GroupStop, усі осі в групі осей негайно припиняються.
  2. Інструкція DMC_GroupStop може бути виконана лише тоді, коли виконуються інструкції групи осей, такі як DMC_MoveDirectAbsolute, DMC_MoveDirectRelative, DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute і DMC_MoveCircularRelative. Після завершення виконання інструкції всі осі в групі осей припиняться, а статус осі все ще буде станом дискретного руху. Перед виконанням інших інструкцій руху, які можуть бути виконані лише коли осі перебувають у стані StandStill, *Enable* інструкції DMC_ GroupEnable має бути встановлено на FALSE.
  3. Коли *Execute* інструкції DMC_GroupStop змінюється з FALSE на TRUE, інструкція виконується. Після завершення виконання інструкцій інструкції групи осей DMC_MoveDirectAbsolute, DMC_MoveDirectRelative, DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute і DMC_MoveCircularRelative усі не можуть бути виконані. Ці інструкції можна виконати лише після того, як *Execute* інструкції DMC_GroupStop зміниться на FALSE.

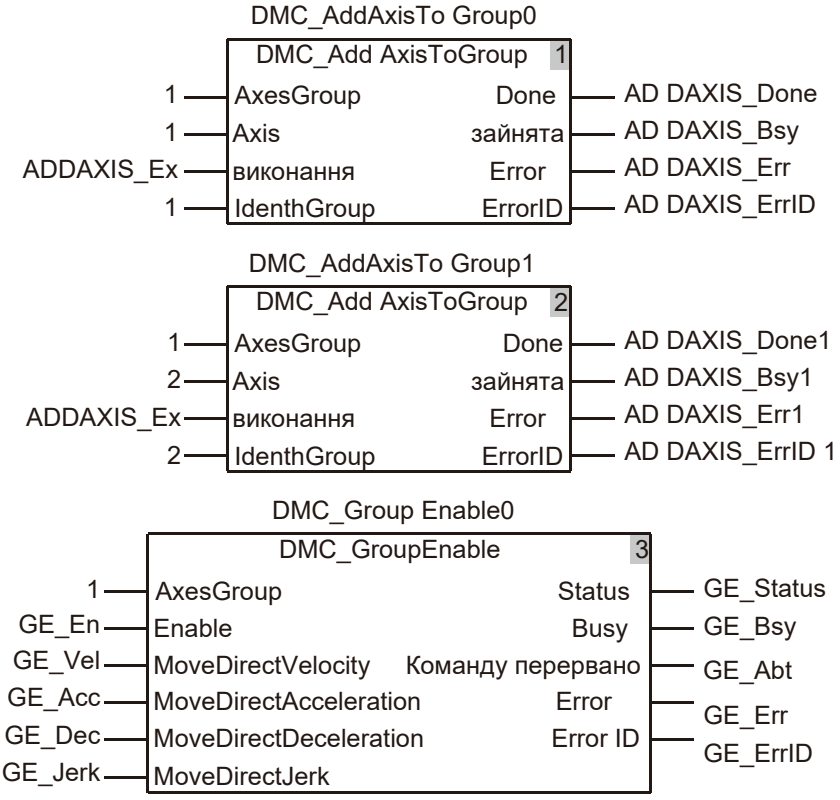
Приклад програмування

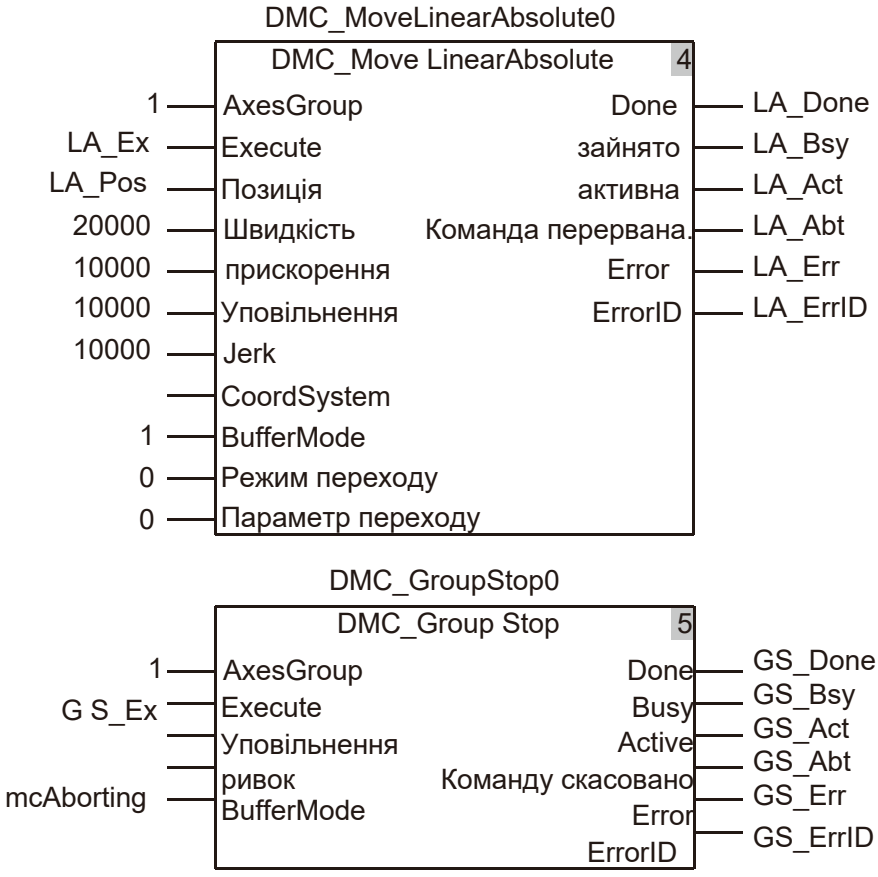
Приклад використання інструкції DMC_GroupStop описано наступним чином.

1. Таблиця змінних і програма

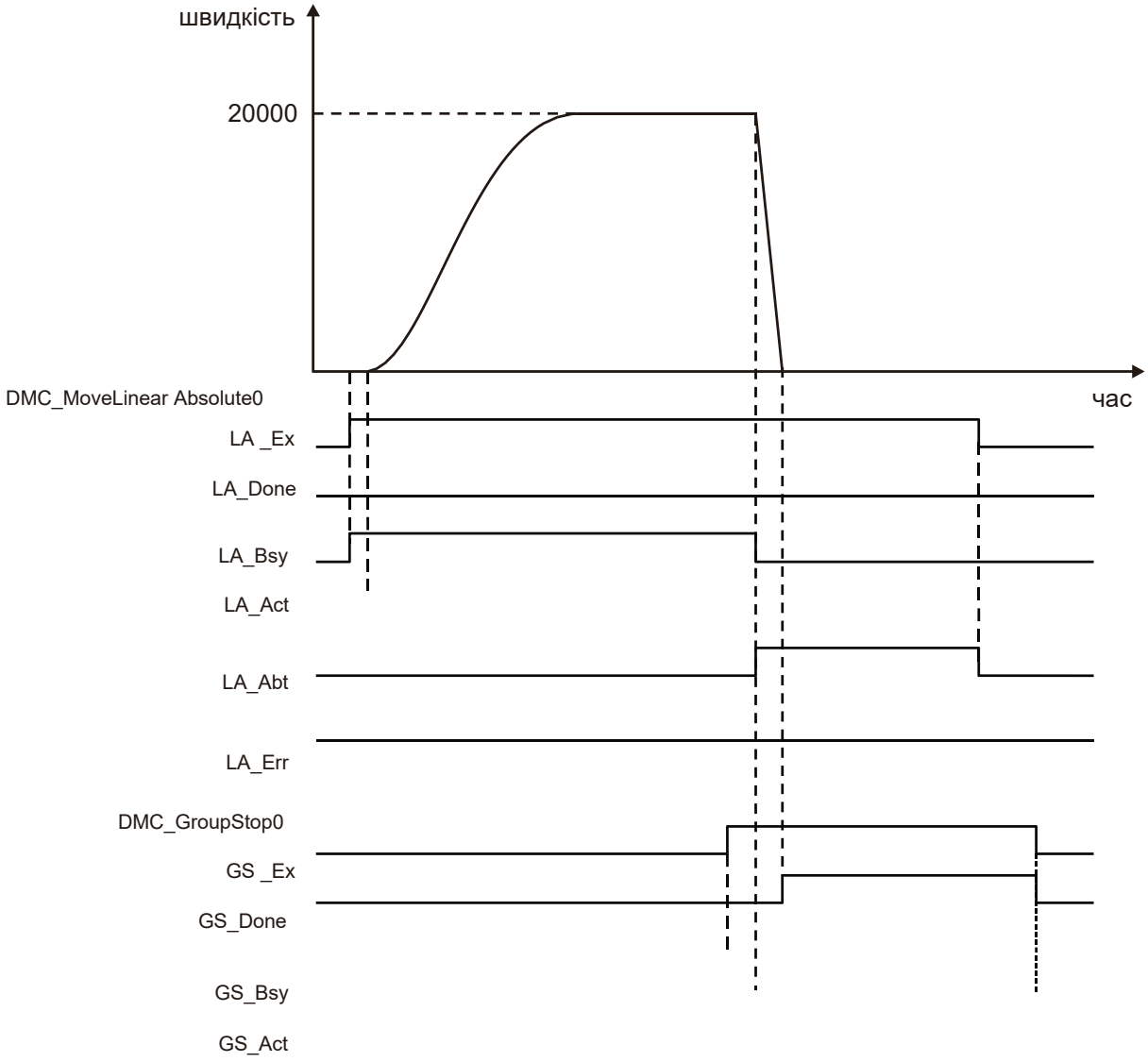
Ім'я змінної	Тип даних	Початкове значення
M1	BOOL	TRUE
DMC_AddAxisToGroup0	DMC_AddAxisToGroup	
ADDAXIS_Ex	BOOL	
ADDAXIS_Done	BOOL	
ADDAXIS_Bsy	BOOL	
ADDAXIS_Err	BOOL	
ADDAXIS_Eid	WORD	
DMC_AddAxisToGroup1	DMC_AddAxisToGroup	
ADDAXIS_Done1	BOOL	
ADDAXIS_Bsy1	BOOL	
ADDAXIS_Err1	BOOL	
ADDAXIS_Eid1	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_EN	BOOL	
GE_VEL	ARRAY [1..8] LREAL	[10000,10000]
GE_ACC	ARRAY [1..8] LREAL	[10000,10000]
GE_DEC	ARRAY [1..8] LREAL	[10000,10000]
GE_JERK	ARRAY [1..8] LREAL	[10000,10000]
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_Eid	WORD	
DMC_MoveLinearAbsolute0	DMC_MoveLinearAbsolute	

Ім'я змінної	Тип даних	Початкове значення
LA_Ex	BOOL	
LA_Pos	ARRAY [1..8] LREAL	[200000,200000]
LA_Done	BOOL	
LA_Bsy	BOOL	
LA_Act	BOOL	
LA_Abt	BOOL	
LA_Err	BOOL	
LA_Eid	WORD	
DMC_GroupStop0	DMC_GroupStop	
GS_Ex	BOOL	
GS_Done	BOOL	
GE_Bsy	BOOL	
GE_Act	BOOL	
GE_Abt	BOOL	
GS_Err	BOOL	
GS_Eid	WORD	




DMC_MoveLinearAbsolute0
DMC_Move LinearAbsolute
4
1
AxesGroup
LA_Ex
Execute
LA_Pos
Позиція
20000
Швидкість
10000
прискорення
10000
Уповільнення
10000
Jerk
CoordSystem
1
BufferMode
0
Режим переходу
0
Параметр переходу
Done
LA_Done
зайнято
LA_Bsy
активна
LA_Act
Команда перервана.
LA_Abt
Error
LA_Err
ErrorID
LA_ErrID
DMC_GroupStop0
DMC_Group Stop
5
1
AxesGroup
G S_Ex
Execute
Уповільнення
ривок
mcAborting
BufferMode
Done
GS_Done
Busy
GS_Bsy
Active
GS_Act
Команду скасовано
GS_Abt
Error
GS_Err
ErrorID
GS_ErrID

3. Крива руху та часовий графік:

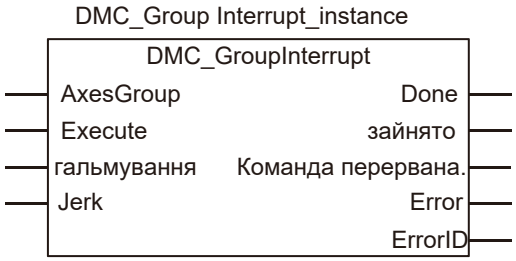


- Виконується DMC_AddAxisToGroup, а потім виконується DMC_GroupEnable, щоб увімкнути групу осей. Після ввімкнення групи осей виконується DMC_MoveLinearAbsolute.

- DMC_GroupStop виконується під час виконання DMC_MoveLinearAbsolute. На даний момент швидкість групи осей швидко стає 0, і DMC_MoveLinearAbsolute переривається. Після припинення роботи групи осей *Done* of DMC_GroupStop змінюється на TRUE.

11.7.6 DMC_GroupInterrupt

FB/FC	Пояснення	Застосовна модель
FB	DMC_GroupInterrupt використовується для призупинення руху поточної групи осей на певний період часу.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Вкажіть кількість груп осей, які потрібно включити	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Уповільнення	Зарезервований	-	-	-
ривок	Зарезервований	-	-	-

Примітка:

Уповільнення та *ривок* зарезервовані, а їхні значення налаштувань недійсні.

- **Вихідний параметр**

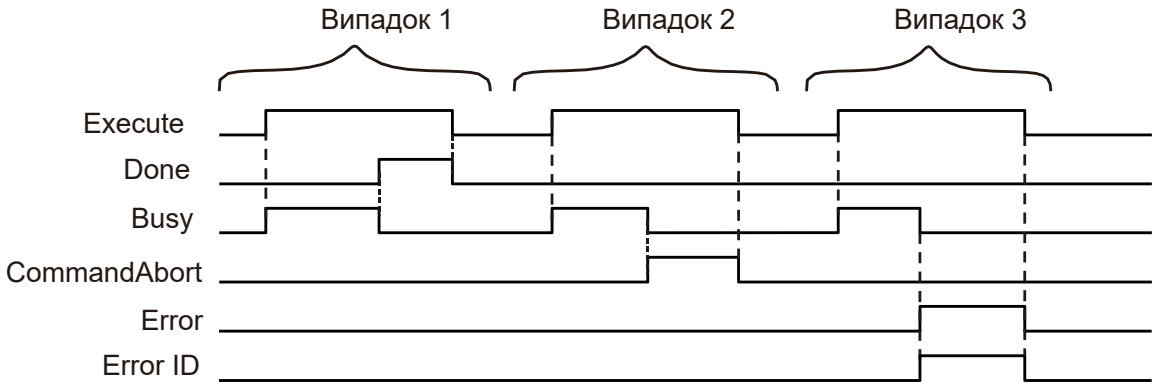
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли групу осей успішно призупинено.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Done* змінюється з FALSE на

Ім'я	Час зміни на TRUE	Час зміни на FALSE
		TRUE. ◆ Коли *CommandAborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
CommandAborted	◆ Коли виконання інструкції переривається.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Коли групу осей успішно призупинено, *Done* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли виконання інструкції перервано, *CommandAborted* змінюється на TRUE, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *CommandAborted* змінюється на FALSE.

Випадок 3 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, а *ErrorID* показує відповідний код помилки. Тим часом *Busy* змінюється на FALSE. *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0, коли *Execute* змінюється на FALSE.

- **функція**

DMC_GroupInterrupt використовується для призупинення руху поточної групи осей на певний період часу.

1. Вхідні параметри *Deceleration* і *Jerk* інструкції DMC_GroupInterrupt зарезервовані, а вхідні значення для них недійсні. Коли виконується інструкція DMC_GroupInterrupt, ПЛК буде сповільнюватися зі швидкістю уповільнення інструкції групи осей, керуючи рухом групи осей, доки група осей не припинить роботу.
2. Інструкція DMC_GroupInterrupt може бути виконана лише тоді, коли групові інструкції осей DMC_MoveDirectAbsolute, DMC_MoveDirectRelative, DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute і DMC_MoveCircularRelative виконуються нормально.

3. DMC_GroupContinue можна використовувати для відновлення групи осей до стану перед призупиненням після виконання DMC_GroupInterrupt.
4. Результати виконання є узгодженими, коли декілька інструкцій DMC_GroupInterrupt виконуються одночасно.

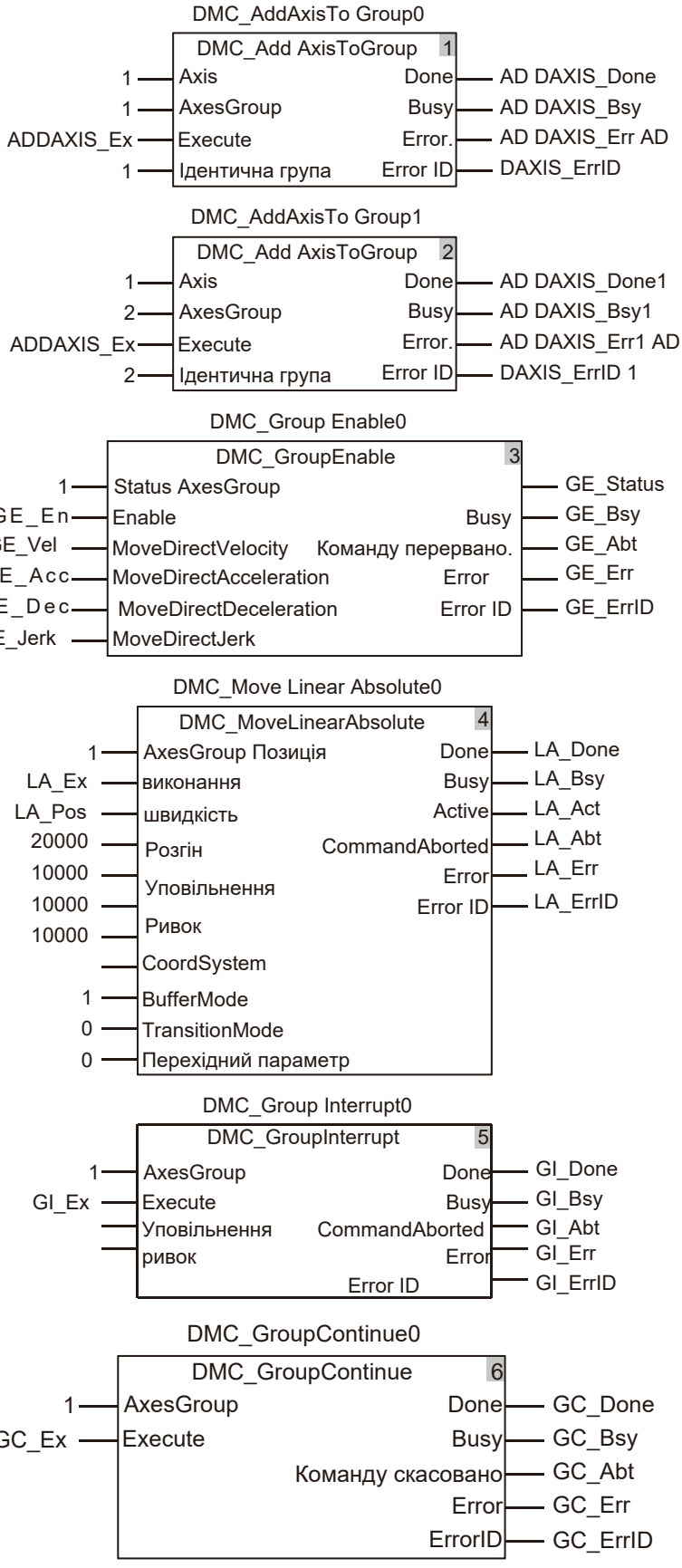
Приклад програмування

Приклад спільного використання DMC_GroupInterrupt з DMC_GroupContinue описаний нижче.

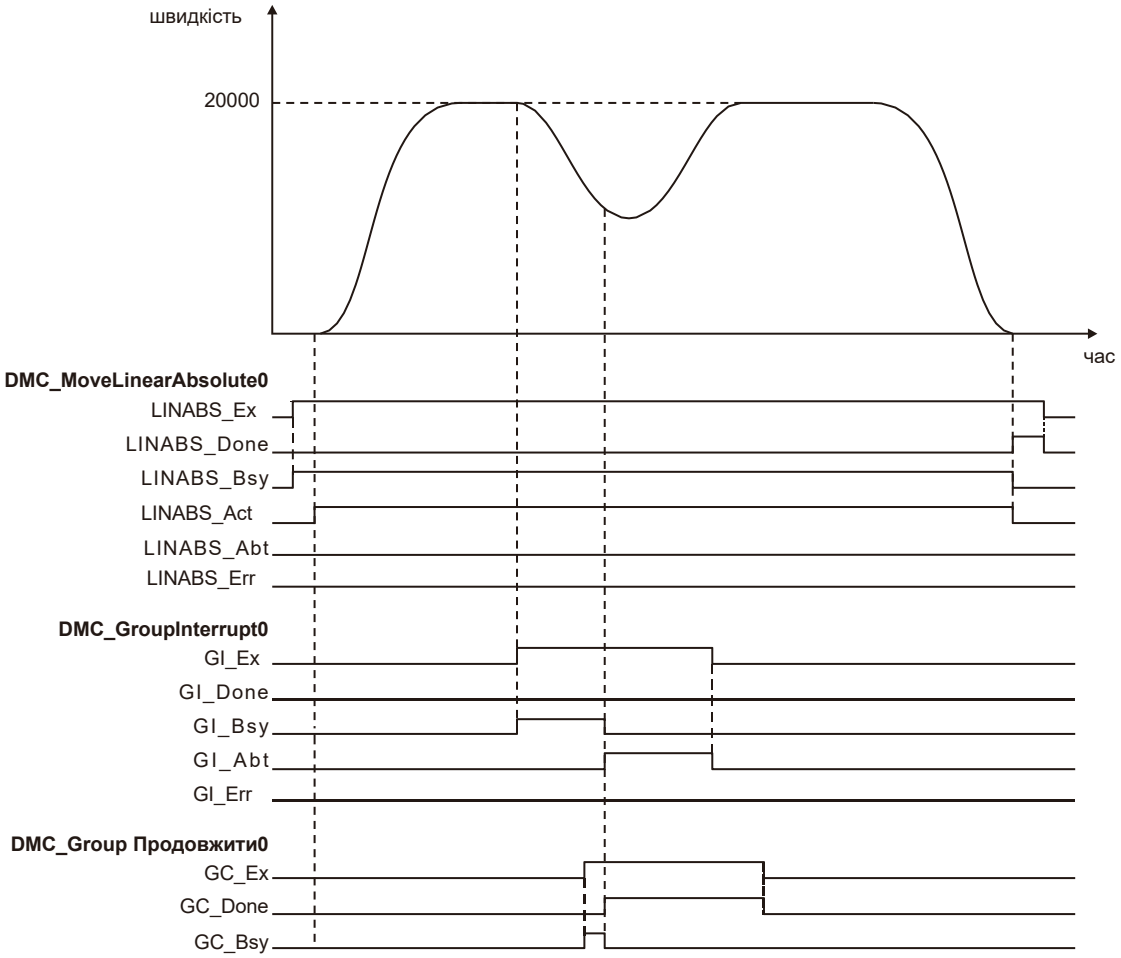
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
M1	BOOL	TRUE
DMC_AddAxisToGroup0	DMC_AddAxisToGroup	
ADDAXIS_Ex	BOOL	
ADDAXIS_Done	BOOL	
ADDAXIS_Bsy	BOOL	
ADDAXIS_Err	BOOL	
ADDAXIS_ErrID	WORD	
DMC_AddAxisToGroup1	DMC_AddAxisToGroup	
ADDAXIS_Done1	BOOL	
ADDAXIS_Bsy1	BOOL	
ADDAXIS_Err1	BOOL	
ADDAXIS_ErrID1	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	[10000,10000]
GE_Acc	ARRAY [1..8] LREAL	[10000,10000]
GE_Dec	ARRAY [1..8] LREAL	[10000,10000]
GE_Jerk	ARRAY [1..8] LREAL	[10000,10000]
GE_Status	BOOL	
GE_Buy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DMC_MoveLinearAbsolute0	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	МАСИВ [1 .. 8] LREAL	[200000,200000]
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_ErrID	WORD	
DMC_GroupInterrupt0	DMC_GroupInterrupt	
GI_Ex	BOOL	
GI_Done	BOOL	
GI_Bsy	BOOL	

Ім'я змінної	Тип даних	Початкове значення
GI_Abt	BOOL	
GI_Err	BOOL	
GI_ErrID	WORD	
DMC_GroupContinue0	DMC_GroupContinue	
GC_Ex	BOOL	
GC_Done	BOOL	
GC_Bsy	BOOL	
GC_Abt	BOOL	
GC_Err	BOOL	
GC_ErrID	WORD	



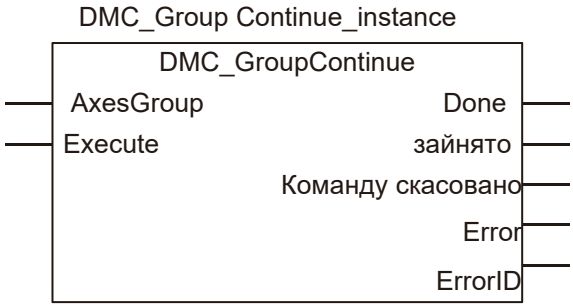
2. Крива руху та часовий графік:



- Виконується DMC_AddAxisToGroup, а потім виконується DMC_GroupEnable, щоб увімкнути групу осей. Після ввімкнення групи осей виконується DMC_MoveLinearAbsolute.
- DMC_GroupInterrupt виконується під час виконання DMC_MoveLinearAbsolute. Після виконання DMC_GroupInterrupt група осей починає сповільнюватися зі швидкістю уповільнення інструкції DMC_MoveLinearAbsolute. Але інструкція DMC_MoveLinearAbsolute не переривається.
- DMC_GroupContinue виконується в процесі уповільнення групи осей. DMC_GroupContinue негайно зупиняє виконання DMC_GroupInterrupt. Тим часом виконання DMC_MovelinearAbsolute продовжується.

11.7.7 DMC_GroupContinue

FB/FC	Пояснення	Застосовна модель
FB	DMC_GroupContinue використовується, щоб призупинена група осей продовжувала працювати.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Вкажіть кількість груп осей, які потрібно включити	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	

- **Вихідний параметр**

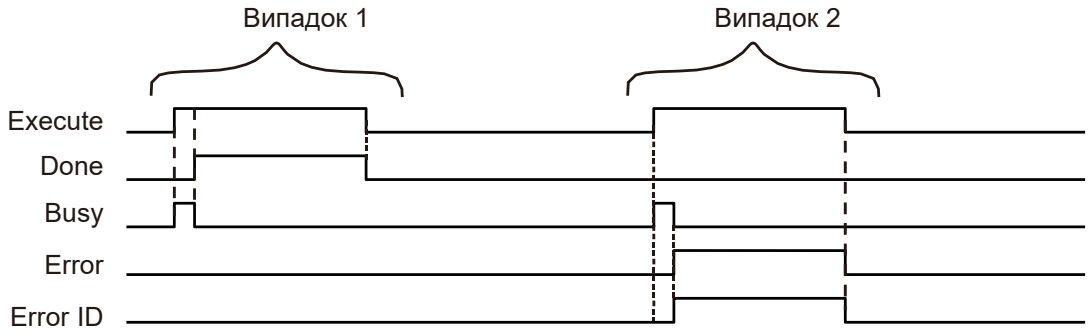
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли інструкцію перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли є помилка.	BOOL	TRUE / FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідного коду помилки.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли група осей продовжує працювати.	◆ Коли *Execute* змінюється з TRUE на FALSE.

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *Done* змінюється з FALSE на TRUE. ◆ Коли *CommandAborted* змінюється з FALSE на TRUE. ◆ Коли *помилка* змінюється з FALSE на TRUE.
CommandAborted	◆ Коли виконання інструкції переривається.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

● **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один період Done змінюється на TRUE, Busy змінюється на FALSE, а рух групи осей не призупиняється. Коли *Execute* змінюється з TRUE на FALSE, *Done* змінюється на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка та *параметр Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один період *помилка* змінюється на TRUE, *ErrorID* відображає відповідні коди помилок, а *Busy* змінюється на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0.

● **функція**

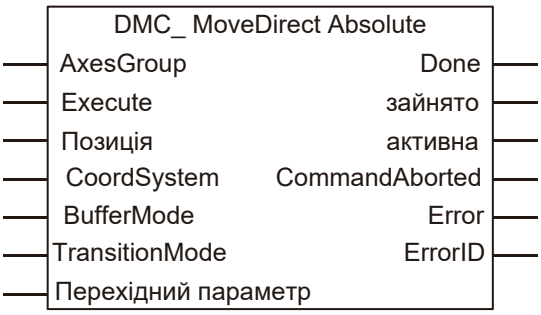
DMC_GroupContinue використовується, щоб призупинена група осей продовжувала працювати.

1. DMC_GroupContinue можна виконати лише тоді, коли група осей інструкції DMC_MoveDirectAbsolute, DMC_MoveDirectRelative, DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute і DMC_MoveCircularRelative перериваються DMC_GroupInterrupt.
2. Після виконання DMC_GroupContinue група осей не буде призупинена та продовжуватиме виконувати дію до призупинення.

11.7.8 DMC_MoveDirectAbsolute

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveDirectAbsolute використовується для керування всіма осями в групі осей для переміщення від поточної позиції до кінцевої позиції з указаною швидкістю.	AS516E-B AS532EST-B AS564EST-B

DMC_ MoveDirect Absolute_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Значення змінної має бути встановлено)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Позиція	Кінцеві положення осей	ARRAY [1..8] LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей 1 : Буферизовано	MC_Buffer_Mode	1: mcBuffered	Коли *Execute* змінюється з FALSE на TRUE.
TransitionMode	Вкажіть режим переходу між двома вказівками групи осей 0 : без вставки кривої переходу	MC_Transition_Mode	0: mcTMНемає	Коли *Execute* змінюється з FALSE на TRUE.
Параметр TransitionP	Встановіть параметр переходу для певного режиму переходу	LREAL	Позитивно, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Виконання інструкції DMC_MoveDirectAbsolute починається, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.

2. Значення вхідного параметра Position[1] ~ Position[8] означає кінцеві позиції для швидкого позиціонування.
3. Вхідний *режим BufferMode* DMC_MoveDirectAbsolute підтримує лише режим mcBuffered. Якщо вибрано інший режим, під час виконання інструкції виникне помилка. Докладніше про BufferMode див. у розділі 10.3.
4. Вхід *TransitionMode* DMC_MoveDirectAbsolute підтримує лише режим mcTMNone. Якщо вибрано інший режим, під час виконання інструкції виникне помилка.
5. Значення вхідного *TransitionParameter* інструкції DMC_MoveDirectAbsolute недійсне.

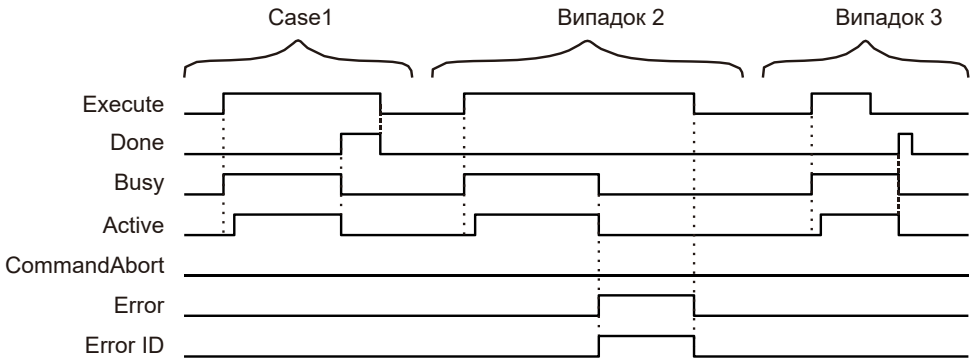
● **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує осями.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли досягнуто кінцеве положення	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли осі починають керуватись інструкцією	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху	◆ Коли *Execute* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через два цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy ma Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Active* і *Busy* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 3 : після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

- **функція**

DMC_MoveDirectAbsolute використовується для групи осей для швидкого позиціонування, і можна керувати однією або декількома осями в групі осей. Прошивка V1.01 і вище підтримує цю функцію.
Осі під час руху відносно незалежні одна від одної. Швидкості, прискорення, уповільнення та ривки осей залежать від вхідних значень DMC_GroupEnable: MoveDirectVelocity, MoveDirectAcceleration, MoveDirectDeceleration і MoveDirectJerk.

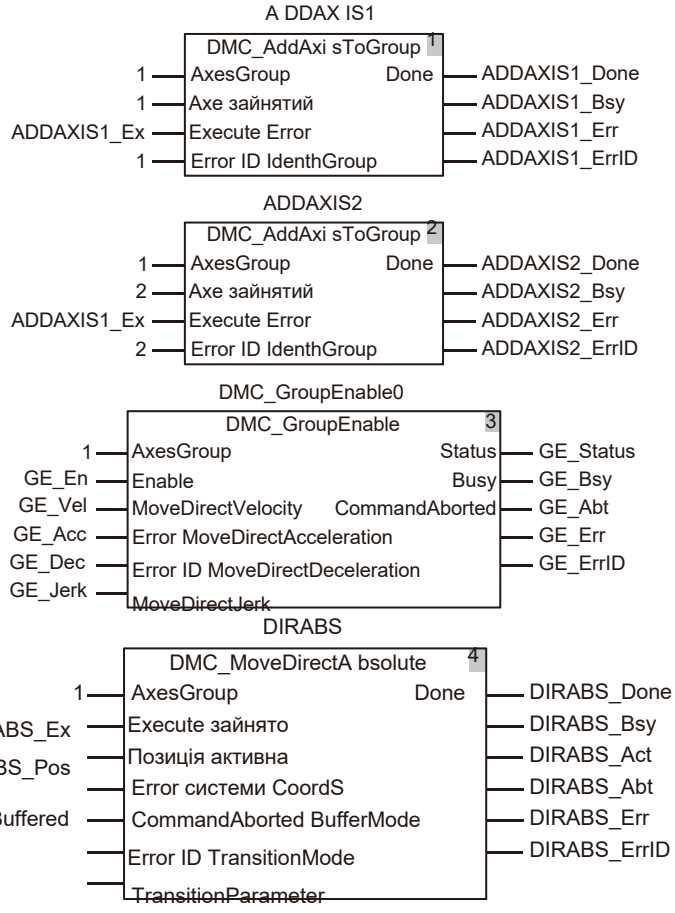
Приклад програмування

Приклад, у якому виконується одна інструкція DMC_MoveDirectAbsolute, виглядає наступним чином.

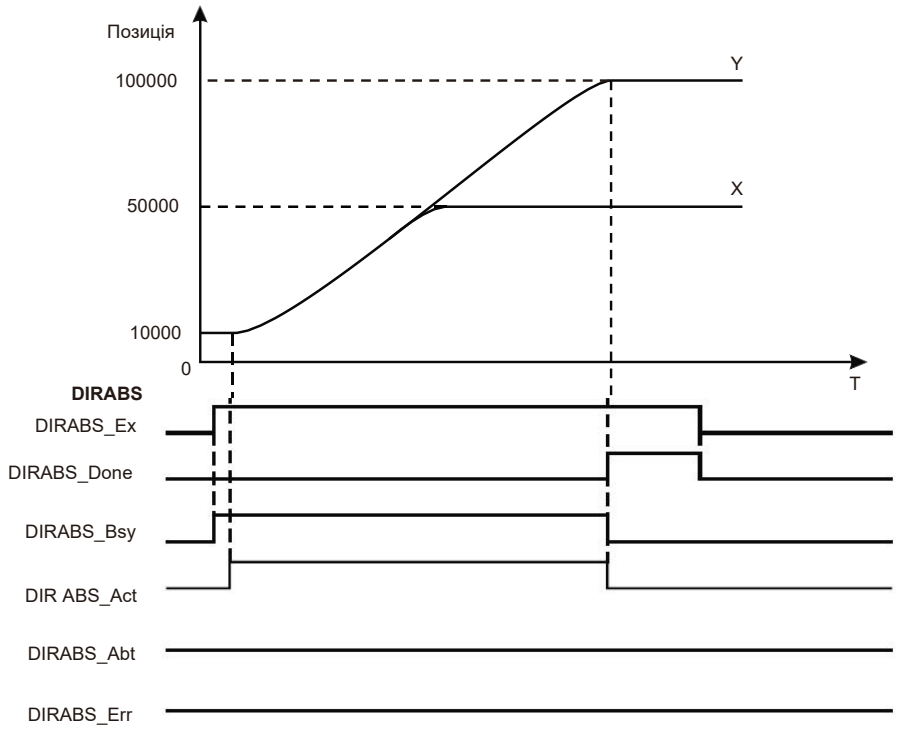
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	

Ім'я змінної	Тип даних	Початкове значення
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DIRABS	DMC_MoveDirectAbsolute	
DIRABS_Ex	BOOL	
DIRABS_Pos	ARRAY [1..8] LREAL	
DIRABS_Done	BOOL	
DIRABS_Bsy	BOOL	
DIRABS_Act	BOOL	
DIRABS_Abt	BOOL	
DIRABS_Err	BOOL	
DIRABS_ErrID	WORD	



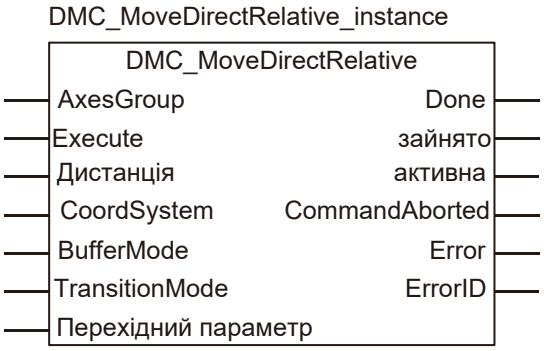
2. Крива руху вісь X-Y і часовий графік



- Початкові позиції осей X і Y дорівнюють 10000. Вхідні дані DIRABS_Pos[1] і DIRABS_Pos[2] DMC_MoveDirectAbsolute встановлені на 50 000 і 100 000 відповідно.
- Коли DIRABS_Ex змінюється на TRUE, DIRABS_Bsy змінюється на TRUE. Через два цикли DIRABS_Act змінюється на TRUE, і група осей починає працювати.
- Вісь X і Y працюють відповідно до встановлених швидкостей, прискорень і уповільнень DMC_GroupEnable. Коли вісь X переміщується до 50 000, вісь X припиняє працювати. На даний момент вісь Y продовжується. Коли вісь Y досягає 100 000, виконання інструкції завершено.

11.7.9 DMC_MoveDirectRelative

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveDirectRelative використовується для керування осями в групі осей для переміщення відповідних відстаней від поточних позицій із заданими швидкостями.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон	Назва параметра
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Відстань	Встановіть відстані, на які переміщуються осі	ARRAY [1..8] LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей 1 : Буферизовано	MC_Buffer_Mode	1: mcBuffered	Коли *Execute* змінюється з FALSE на TRUE.
TransitionMode	Вкажіть режим переходу між двома вказівками групи осей 0 : без вставки кривої переходу	MC_Transition _Mode	0: mcTMНемає	Коли *Execute* змінюється з FALSE на TRUE.
TransitionParameter	Встановіть параметр переходу для певного режиму переходу	LREAL	Позитивне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Виконання інструкції DMC_MoveDirectRelative починається, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.
2. Значення вхідного параметра Distance[1] ~ Distance[8] означає відстані руху від початкових позицій до кінцевих позицій для осей.
3. Вхідний *режим BufferMode* DMC_MoveDirectRelative підтримує лише режим mcBuffered. Якщо вибрано інший режим, під час виконання інструкції виникне помилка. Докладніше про BufferMode див. у розділі 10.3.
4. Вхід *TransitionMode* DMC_ MoveDirectRelative підтримує лише режим mcTMNone. Якщо вибрано інший режим, під час виконання інструкції виникне помилка.
5. Значення вхідного *TransitionParameter* інструкції DMC_ MoveDirectRelative недійсне.

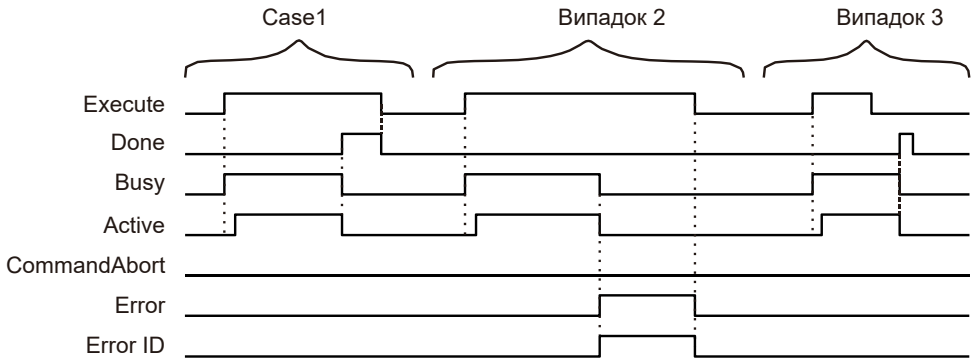
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує осями.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли кінцеві позиції досягнуті.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли інструкція починає керувати осями.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через два цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy ma Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 3 : після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

- **функція**

DMC_MoveDirectRelative використовується для групи осей для швидкого позиціонування, і можна керувати однією або декількома осями в групі осей. Прошивка V1.01 і вище підтримує цю функцію.
Осі під час руху відносно незалежні одна від одної. Швидкості, прискорення, уповільнення та ривки осей залежать від вхідних значень DMC_GroupEnable: MoveDirectVelocity, MoveDirectAcceleration, MoveDirectDeceleration і MoveDirectJerk.

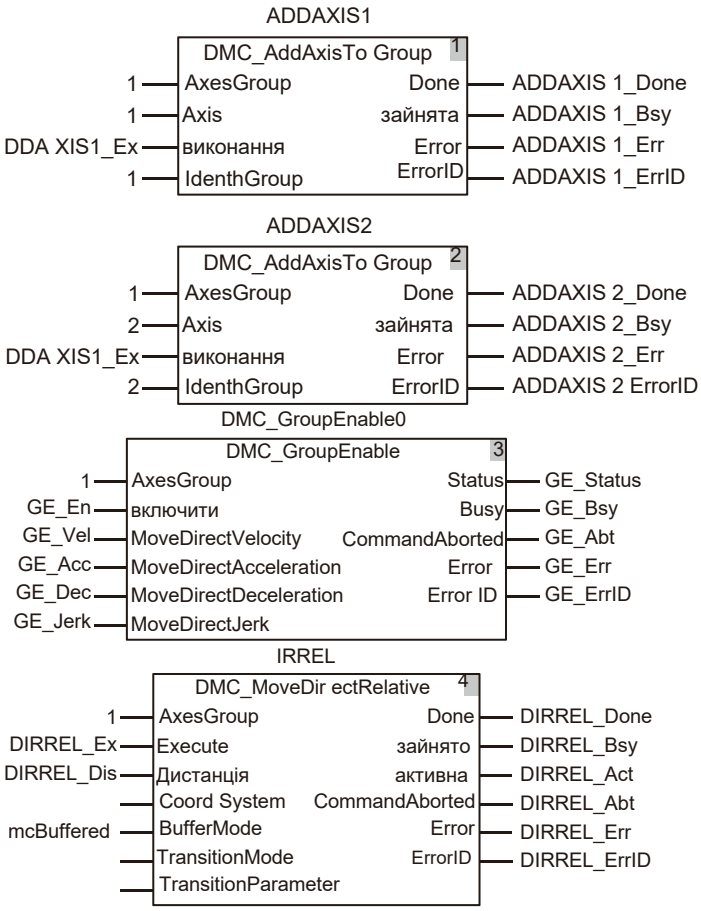
Приклад програмування

Приклад, у якому виконується одна інструкція DMC_MoveDirectRelative, виглядає наступним чином.

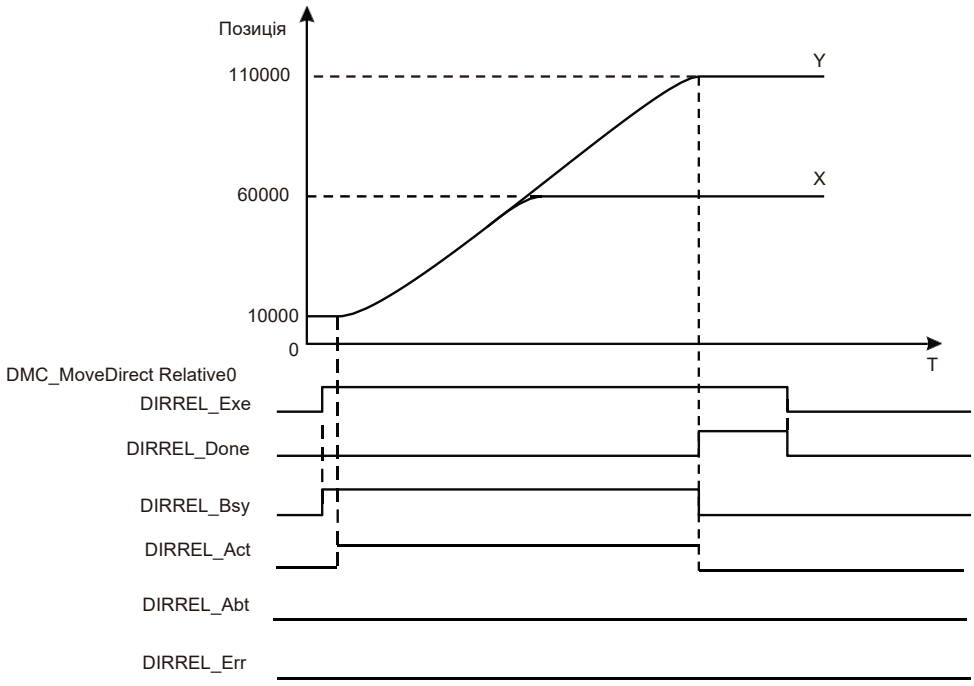
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	

Ім'я змінної	Тип даних	Початкове значення
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
DIRREL	DMC_MoveDirectRelative	
DIRREL_Ex	BOOL	
DIRREL_Dis	ARRAY [1..8] LREAL	
DIRREL_Done	BOOL	
DIRREL_Bsy	BOOL	
DIRREL_Act	BOOL	
DIRREL_Abt	BOOL	
DIRREL_Err	BOOL	
DIRREL_ErrID	WORD	



2. Крива руху вісь X-Y і часовий графік

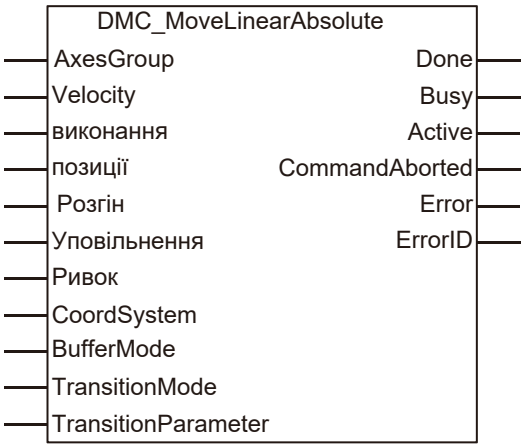


- Початкові позиції осей X і Y дорівнюють 10000. Вхідні дані DIRREL_Dis[1] і DIRREL_Dis[2] DMC_MoveDirectRelative встановлені на 50 000 і 100 000 відповідно.
- Коли DIRREL_Ex змінюється на TRUE, DIRREL_Bsy змінюється на TRUE. Через два цикли DIRREL_Act змінюється на TRUE, і група осей починає працювати.
- Вісь X і Y працюють відповідно до заданих швидкостей, прискорень і уповільнень DMC_GroupEnable. Коли вісь X переміщується до 60 000, вісь зупиняється. На даний момент вісь Y продовжується. Коли вісь Y досягає 110000, виконання інструкції завершено.

11.7.10 DMC_MoveLinearAbsolute

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveLinearAbsolute керує осями для виконання руху лінійної інтерполяції. Кінцеві положення осей є абсолютними.	AS516E-B AS532EST-B AS564EST-B

DMC_MoveLinearAbsolute_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Позиція	Вкажіть кінцеві позиції для лінійної інтерполяції	ARRAY [1..8] LREAL	Від'ємне число, Додатне число, 0, (0)	Коли *Execute* змінюється з FALSE на TRUE.
швидкість	Встановіть максимальну результуючу швидкість. (Одиниця: одиниця/секунда)	LREAL	Додатне число (0)	Коли Execute змінюється з FALSE на TRUE.
Прискорення	Встановіть максимальне результуюче прискорення (Одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Встановіть максимальне результуюче уповільнення (Одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Встановіть максимальний результуючий ривок (Одиниця вимірювання: одиниця/секунда ^{3})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей 1: Буферизовано 3: Змішано зі швидкістю попередньої інструкції	MC_Buffer _Mode	1: mcBuffered 3: mcBlendingPrevious	Коли *Execute* змінюється з FALSE на TRUE.
TransitionMode	Вкажіть режим переходу між двома груповими інструкціями осей 0: криву переходу не вставлено 2: Зробіть перехід із встановленою постійною швидкістю 3: Зробіть перехід на основі вказаної кутової відстані	MC_Transition_ Mode	0: mcTMНемає 2: mcTMConstantVelocity 3: mcTMCornerDistance	Коли *Execute* змінюється з FALSE на TRUE.
Перехідний параметр	Встановіть параметр переходу для певного режиму переходу	LREAL	Додатне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Інструкція DMC_MoveLinearAbsolute починає виконуватися, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.
2. Значення вхідного параметра Position[1] ~ Position[8] означає кінцеві позиції для лінійної інтерполяції.
3. Зверніться до розділу 10.2 щодо зв'язку між швидкістю, прискоренням, уповільненням і ривком.
4. Докладніше про BufferMode див. у розділі 10.3.
5. Значення вхідного *TransitionParameter* інструкції DMC_ MoveLinearAbsolute недійсне, якщо mcTMCornerDistance не вибрано як TransitionMode.

● **Вихідні параметри**

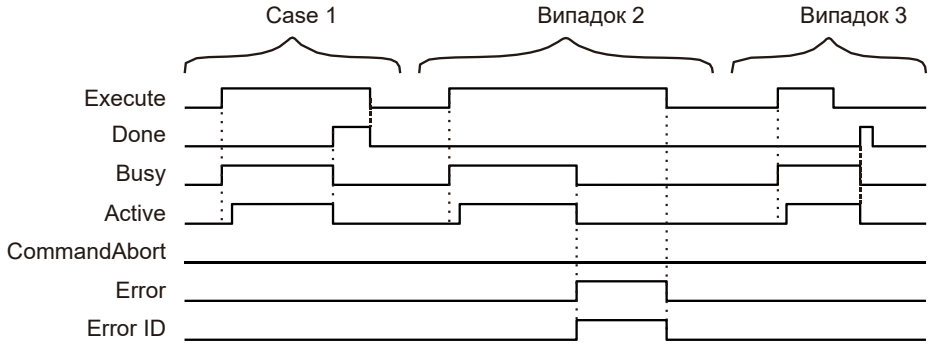
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує осями.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

11

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли кінцеві позиції досягнуті.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли осі починають керуватись інструкцією.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через два цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy ma Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 3 : після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

- **функція**

1. DMC_ MoveLinearAbsolute використовується для групи осей для проведення лінійної інтерполяції, і можна керувати однією або декількома осями в групі осей. Прошивка V1.01 і вище підтримує цю функцію.

Вхідна *швидкість* DMC_MoveLinearAbsolute є цільовою швидкістю термінального приводу. Швидкість кінцевого приводу і швидкості осей мають наступне співвідношення.

Квадрат швидкості кінцевого приводу = сума квадратів швидкостей осей

Вхідні дані *Acceleration* і *Deceleration* DMC_ MoveLinearAbsolute є цільовим прискоренням і цільовим уповільненням термінального приводу. Прискорення та уповільнення кінцевого приводу та прискорення та уповільнення осей мають таке співвідношення.

（ уповільнення ） кінцевого приводу = сума квадратів прискорення （ уповільнення ） осей

2. *Інструкція Jerk* DMC_MoveLinearAbsolute зарезервована.
3. Нижче наведено зв'язок між *BufferMode* , *TransitionMode* і *TransitionParameter* . Якщо mcBuffered вибрано як *BufferMode* , *TransitionMode* підтримує лише mcTMNone. Якщо mcBlendingPrevious вибрано як *BufferMode* , *TransitionMode* підтримує два режими: mcTMConstantVelocity та mcTMCornerDistance.

Значення BufferMode	Значення TransitionMode	опис
mcBuffered(1)	mcTMЖодного(0)	Зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію.
mcBlending-Previous(3)	mcTMConstant-Velocity(2)	Плавний перехід: зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію. Швидкість переходу є результуючою швидкістю попередньої інструкції. Після перемикання команд кінцевий привод виконує плавний перехід, а потім здійснює лінійний рух, як показано в наступному прикладі 2.
	mcTMCorner-Distance(3)	Перехід кутової відстані: зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію. Коли відстань між положенням термінального приводу та кінцевим положенням попередньої інструкції інтерполяції дорівнює значенню *TransitionParameter* під час виконання попередньої інструкції інтерполяції, виконання попередньої інструкції інтерполяції завершується негайно, а потім починається виконання поточної інструкції. Під час виконання поточної інструкції термінальний актуатор рухається по коловому шляху, а потім виконує лінійну інтерполяцію. Відстань між кінцевою точкою дуги кола та кінцевою точкою попередньої інструкції інтерполяції все ще дорівнює значенню *TransitionParameter* як показано в наступному прикладі 3.

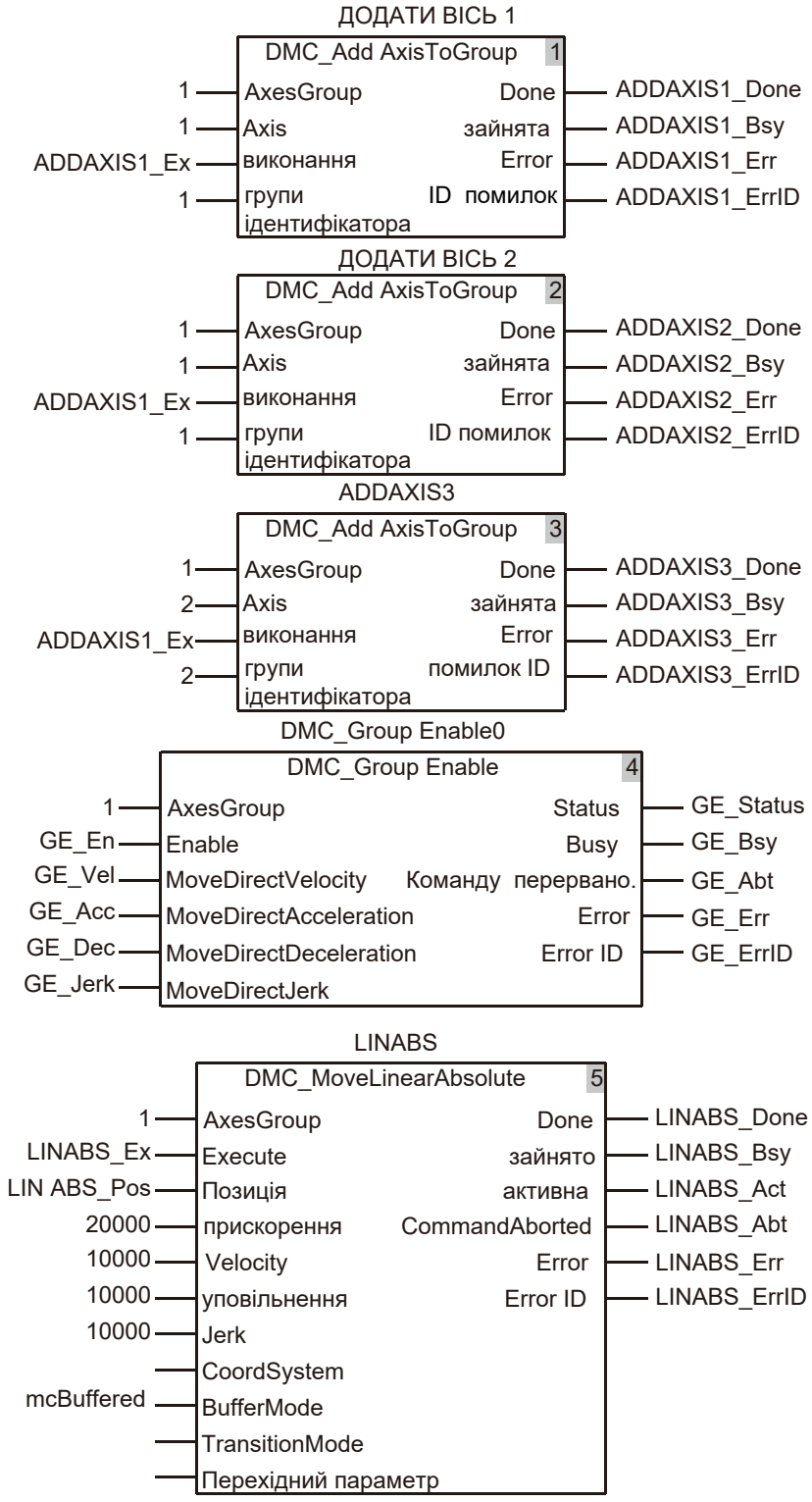
Приклад програмування 1

Приклад, у якому виконується одна інструкція DMC_MoveLinearAbsolute, виглядає наступним чином.

1. Таблиця змінних і програма

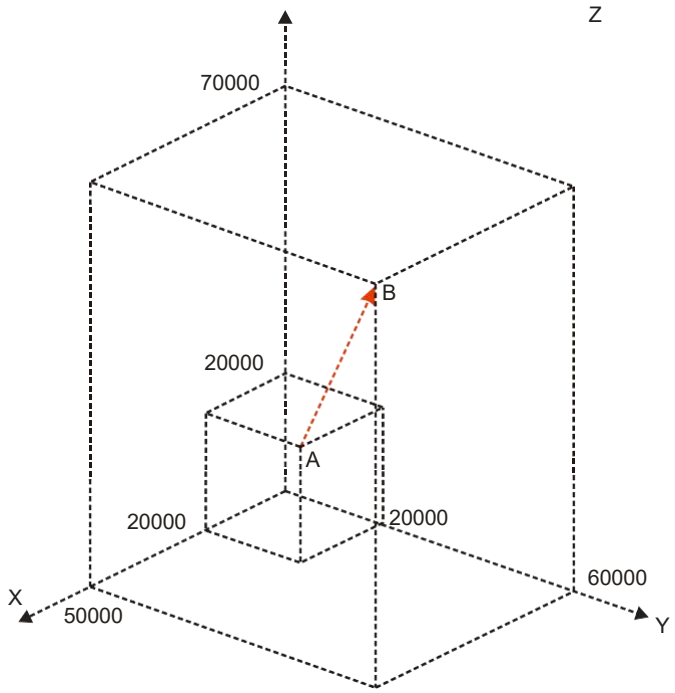
Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS 2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
ADDAXIS3	DMC_AddAxisToGroup	
ADDAXIS3_Done	BOOL	
ADDAXIS3_Bsy	BOOL	
ADDAXIS3_Err	BOOL	
ADDAXIS3_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	ARRAY [1..8] LREAL	
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_ErrID	WORD	

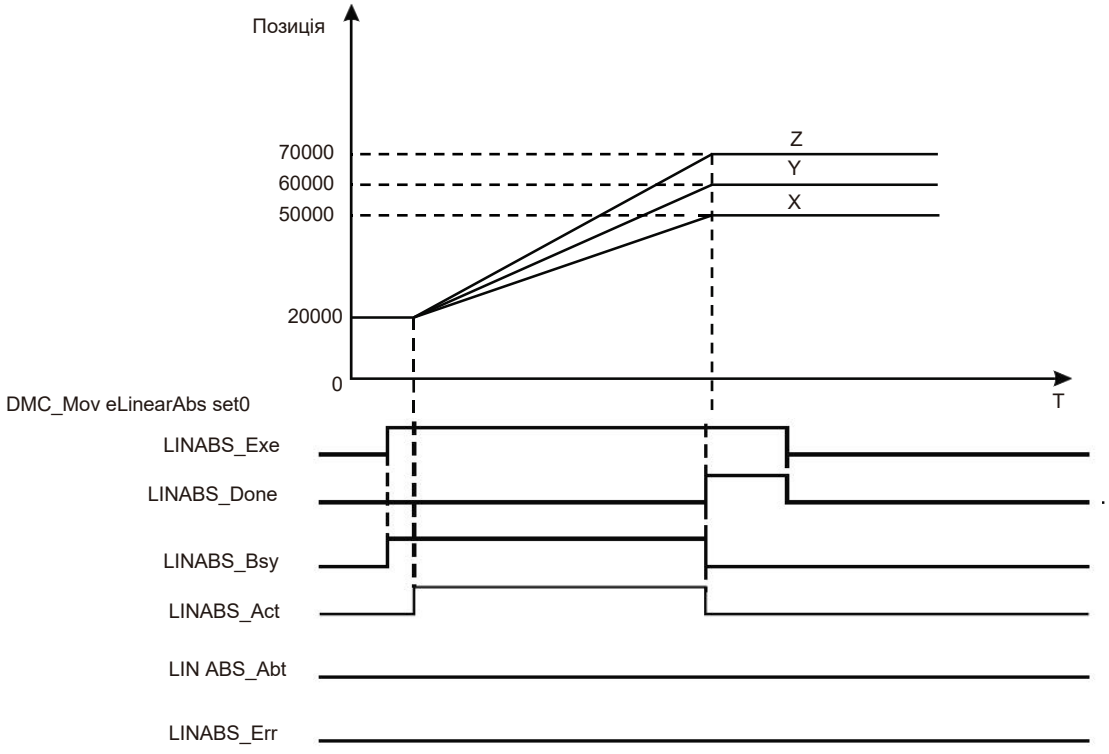

ДОДАТИ ВІСЬ 1
DMC_Add AxisToGroup
1
1
AxesGroup
1
Axis
ADDAXIS1_Ex
виконання
1
групи ідентифікатора
Done
зайнята
Error
ID помилок
ADDAXIS1_Done
ADDAXIS1_Bsy
ADDAXIS1_Err
ADDAXIS1_ErrID
ДОДАТИ ВІСЬ 2
DMC_Add AxisToGroup
2
1
AxesGroup
1
Axis
ADDAXIS1_Ex
виконання
1
групи ідентифікатора
Done
зайнята
Error
ID помилок
ADDAXIS2_Done
ADDAXIS2_Bsy
ADDAXIS2_Err
ADDAXIS2_ErrID
ADDAXIS3
DMC_Add AxisToGroup
3
1
AxesGroup
2
Axis
ADDAXIS1_Ex
виконання
2
групи ідентифікатора
Done
зайнята
Error
помилок ID
ADDAXIS3_Done
ADDAXIS3_Bsy
ADDAXIS3_Err
ADDAXIS3_ErrID
DMC_Group Enable0
DMC_Group Enable
4
1
AxesGroup
GE_En
Enable
GE_Vel
MoveDirectVelocity
GE_Acc
MoveDirectAcceleration
GE_Dec
MoveDirectDeceleration
GE_Jerk
MoveDirectJerk
Status
Busy
Команду перервано.
Error
Error ID
GE_Status
GE_Bsy
GE_Abt
GE_Err
GE_ErrID
LINABS
DMC_MoveLinearAbsolute
5
1
AxesGroup
LINABS_Ex
Execute
LIN ABS_Pos
Позиція
20000
прискорення
10000
Velocity
10000
уповільнення
10000
Jerk
CoordSystem
mcBuffered
BufferMode
TransitionMode
Перехідний параметр
Done
зайнято
активна
CommandAborted
Error
Error ID
LINABS_Done
LINABS_Bsy
LINABS_Act
LINABS_Abt
LINABS_Err
LINABS_ErrID

11

2. Перегляньте весь процес руху після виконання інструкції.



3. Крива руху вісь X, вісь Y, вісь Z і часова діаграма



❖ Початкові позиції осей X, Y і Z дорівнюють 20000. LINABS_Pos[1], LINABS_Pos[2] і LINABS_Pos[3] DMC_MoveLinearAbsolute мають значення 50 000, 60 000 і 70 000 відповідно.

- Коли LINABS_Ex змінюється на TRUE, LINABS_Bsy змінюється на TRUE. Через два цикли LINABS_Act змінюється на TRUE, і група осей починає працювати.
- Вісь X, вісь Y і вісь Z досягають відповідних цільових положень одночасно, LINABS_Done змінюється на TRUE, а LINABS_Bsy і LINABS_Act змінюються на FALSE, коли виконання інструкції DMC_MoveLinearAbsolute завершено.

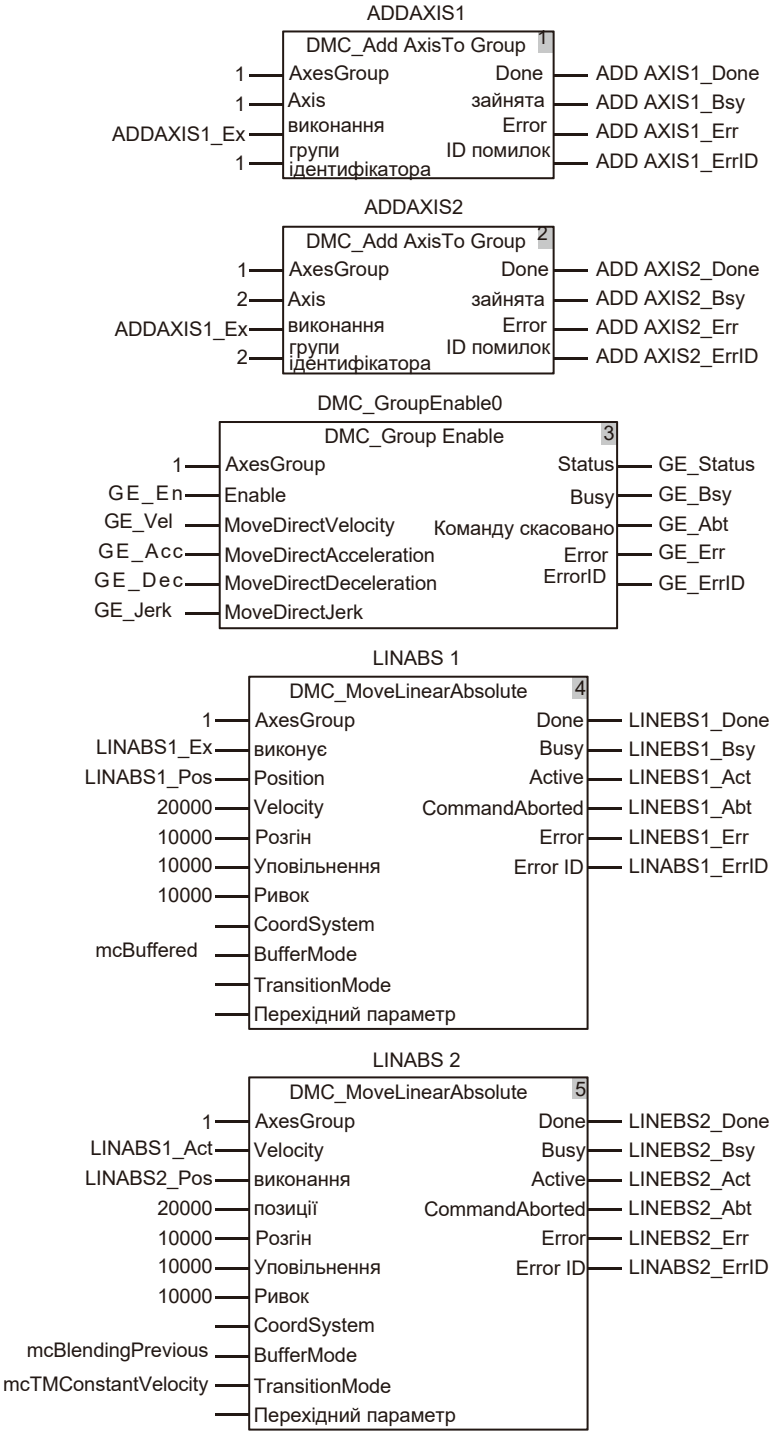
Приклад програмування 2

Приклад, у якому є дві інструкції DMC_MoveLinearAbsolute і режим переходу між ними mcTMConstantVelocity, виглядає наступним чином.

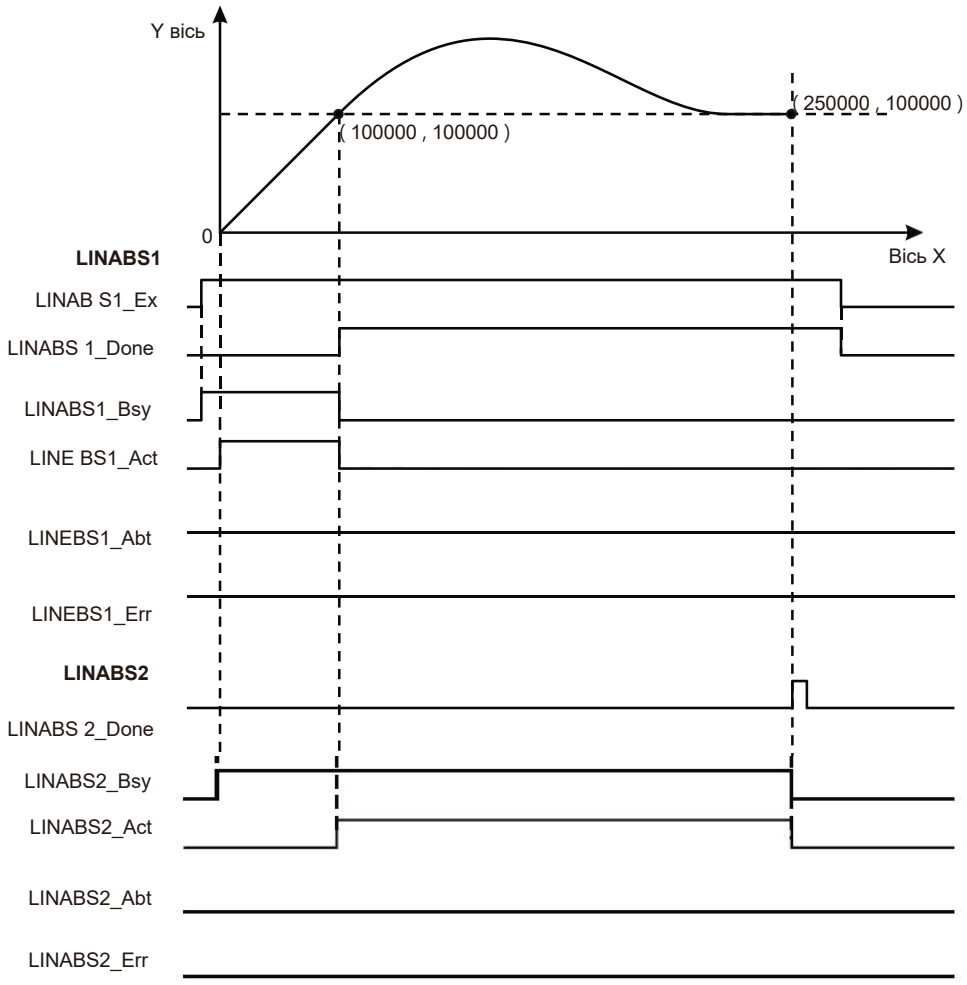
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS1	DMC_MoveLinearAbsolute	
LINABS1_Ex	BOOL	
LINABS1_Pos	МАСИВ [1 .. 8] LREAL	
LINABS1_Done	BOOL	
LINABS1_Bsy	BOOL	
LINABS1_Act	BOOL	
LINABS1_Abt	BOOL	
LINABS1_Err	BOOL	

Ім'я змінної	Тип даних	Початкове значення
LINABS1_ErrID	WORD	
LINABS2	DMC_MoveLinearAbsolute	
LINABS2_Pos	МАСИВ [1 .. 8] LREAL	
LINABS2_Done	BOOL	
LINEBS2_Bsy	BOOL	
LINABS2_Act	BOOL	
LINABS2_Abt	BOOL	
LINABS2_Err	BOOL	
LINABS2_ErrID	WORD	



2. Перегляньте криву руху та часову діаграму термінального приводу.



- Встановіть BufferMode на mcBlendingPrevious і TransitionMode на mcTMConstantVelocity для LINABS2.
- Значення LINABS1_Pos[1] і LINABS1_Pos[2] LINABS1 обидва встановлені на 100,0 Значення LINABS2_Pos[1] і LINABS2_Pos[2] LINABS2 встановлені на 250 000 і 100 000 відповідно.
- Спочатку виконуються ADDAXIS1 і ADDAXIS2, а потім виконується DMC_GroupEnable0. Коли групу осей увімкнено, LINABS1 виконується, а потім LINABS2 виконується негайно.
- Коли привод терміналу досягає координат (100 000, 100 000), LINABS1_Done змінюється на TRUE. Тим часом LINABS2_Act змінюється на TRUE і LINABS2 починає виконуватися. На даний момент швидкість термінального приводу є цільовою швидкістю 20000 попередньої інструкції.
- Термінальний привід виконує плавний перехід, а потім здійснює лінійний рух після виконання LINABS2. Виконання інструкції завершується, коли виконавчий механізм терміналу досягає координат (250 000, 100 000).

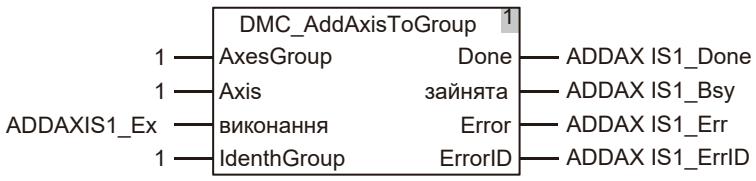
Приклад програмування 3

Приклад, у якому є дві інструкції DMC_MoveLinearAbsolute і режим переходу між ними mcTMCornerDistance, виглядає наступним чином.

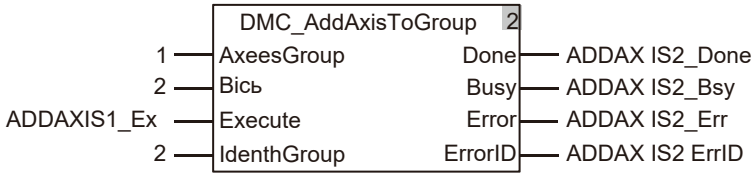
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINABS1	DMC_MoveLinearAbsolute	
LINABS1_Ex	BOOL	
LINABS1_Pos	МАСИВ [1 .. 8] LREAL	
LINABS1_Done	BOOL	
LINABS1_Bsy	BOOL	
LINABS1_Act	BOOL	
LINABS1_Abt	BOOL	
LINABS1_Err	BOOL	
LINABS1_ErrID	WORD	
LINABS2	DMC_MoveLinearAbsolute	
LINABS2_Pos	МАСИВ [1 .. 8] LREAL	
LINABS2_Done	BOOL	
LINEBS2_Bsy	BOOL	
LINABS2_Act	BOOL	
LINABS2_Abt	BOOL	
LINABS2_Err	BOOL	
LINABS2_ErrID	WORD	

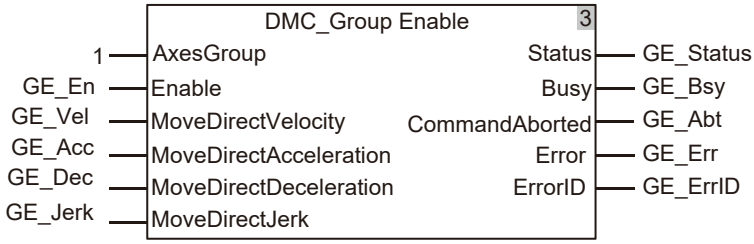
ADDAXIS1



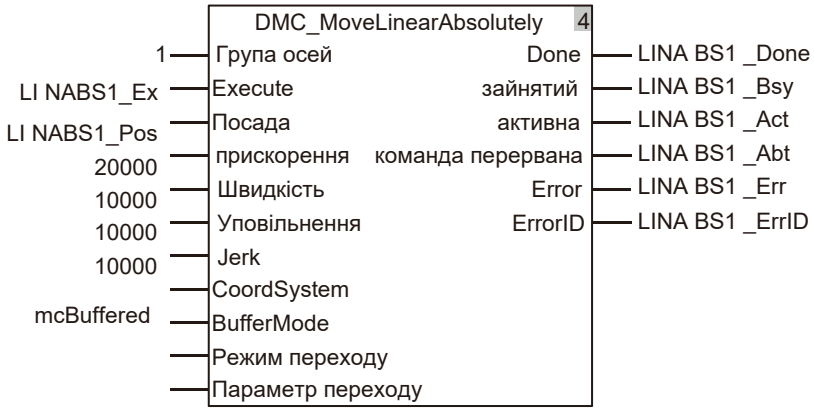
ADDAXIS2



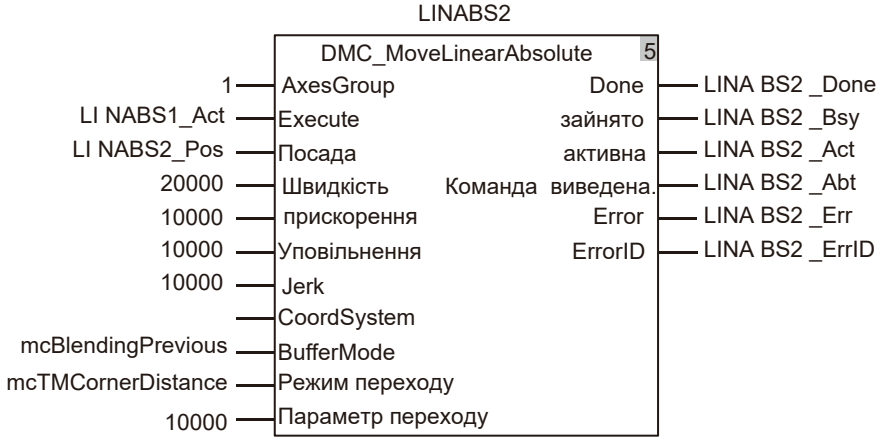
DMC_GroupEnable 0



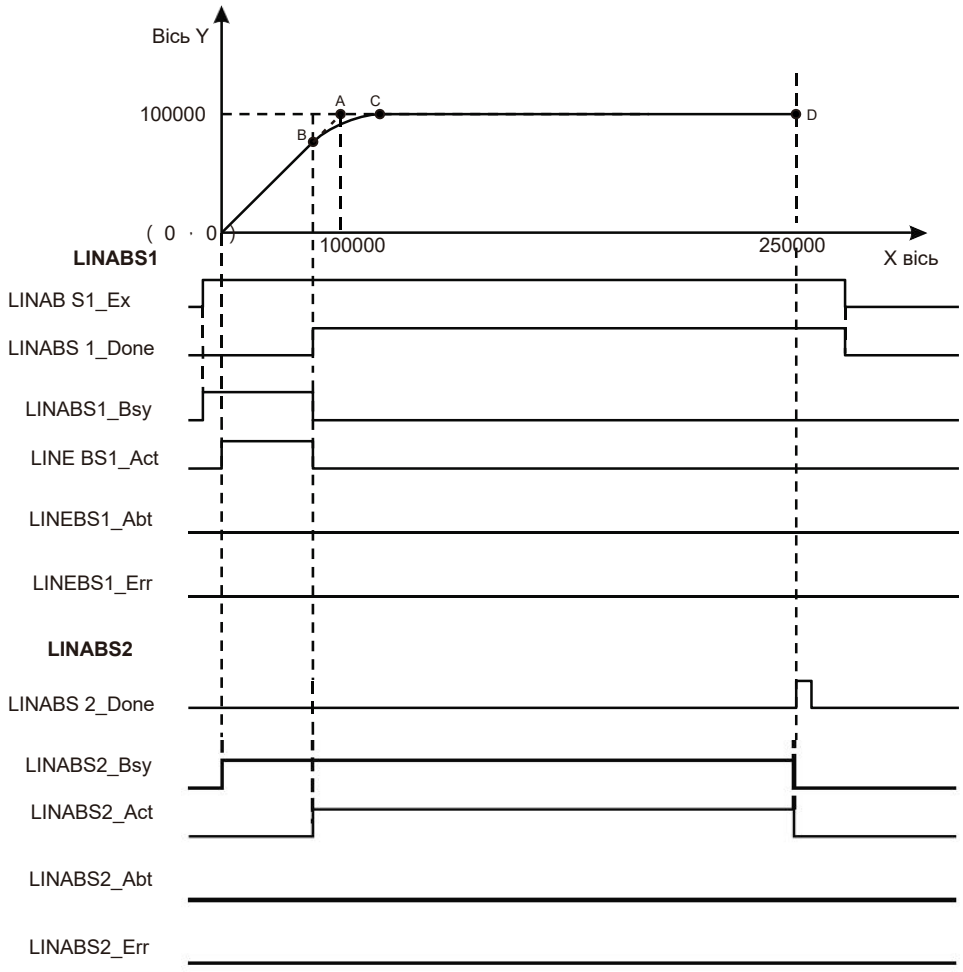
LINABS1



LINABS2



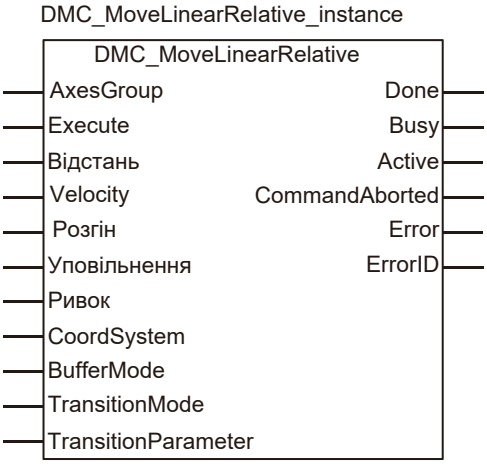
2. Перегляньте криву руху та часову діаграму термінального приводу.



- Встановіть BufferMode на mcBlendingPrevious, TransitionMode на mcTMCornerDistance і TransitionParameter на 10000 для LINABS2.
- Значення LINABS1_Pos [1] і LINABS1_Pos [2] LINABS1 мають значення 100 000. А значення LINABS2_Pos [1] і LINABS2_Pos [2] для LINABS2 встановлені на 250 000 і 100 000 відповідно.
- Спочатку виконуються ADDAXIS1 і ADDAXIS2, а потім виконується DMC_GroupEnable0. Коли групу осей увімкнено, LINABS1 виконується, а потім LINABS2 виконується негайно.
- Коли виконавчий механізм терміналу потрапляє в точку B у системі координат, LINABS1_Done змінюється на TRUE. Тим часом LINABS2_Act змінюється на TRUE і LINABS2 починає виконуватися.
- Після виконання LINABS2 виконавчий механізм термінала рухається по дузі, поки не досягне точки C. Після цього він продовжує виконувати лінійну інтерполяцію.
- Відстань від точки B до точки A дорівнює відстані від точки C до точки A. Вона також дорівнює 10000, значенню вхідного *TransitionParameter* LINABS2.

11.7.11 DMC_MoveLinearRelative

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveLinearRelative керує осями для виконання руху лінійної інтерполяції. Кінцеві положення осей є взаємними.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
Відстань	Встановіть відстані, на які переміщуються осі	ARRAY [1..8] LREAL	Від'ємне число, Додатне число, 0, (0)	Коли *Execute* змінюється з FALSE на TRUE.
швидкість	Встановіть максимальну результуючу швидкість. (Одиниця: одиниця/секунда)	LREAL	Позитивний (0)	Коли *Execute* змінюється з FALSE на TRUE.
Прискорення	Встановіть максимальне результуюче прискорення (Одиниця: одиниця/секунда ^{2})	LREAL	Позитивний (0)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Встановіть максимальне результуюче уповільнення (Одиниця: одиниця/секунда ^{2})	LREAL	Позитивний (0)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Встановіть максимальний результуючий ривок (Одиниця: одиниця/секунду ^{3})	LREAL	Позитивний (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей 1: Буферизовано 3: Змішано зі швидкістю попередньої інструкції	MC_Buffer_Mode	1: mcBuffered 3: mcBlending-Previous	Коли *Execute* змінюється з FALSE на TRUE.
TransitionMode	Вкажіть режим переходу між двома груповими інструкціями осей 0: криву переходу не вставлено 2: Зробіть перехід із встановленою постійною швидкістю 3: Зробіть перехід на основі вказаної кутової відстані	MC_Transition_Mode	0: mcTMNone 2 : mcTMConstantVelocity 3: mcTMCorner-Distance	Коли *Execute* змінюється з FALSE на TRUE.
TransitionParameter	Встановіть параметр переходу для певного режиму переходу	LREAL	Позитивно, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Інструкція DMC_MoveLinearRelative починає виконуватися, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.
2. Значення вхідного параметра Distance[1] ~ Distance[8] означає кінцеві позиції для лінійної інтерполяції.
3. Зверніться до розділу 10.2 щодо зв'язку між швидкістю, прискоренням, уповільненням і ривком.
4. Докладніше про BufferMode див. у розділі 10.3.
5. Значення вхідного параметра *TransitionParameter* інструкції DMC_MoveLinearRelative недійсне, якщо mcTMCornerDistance не вибрано як TransitionMode.

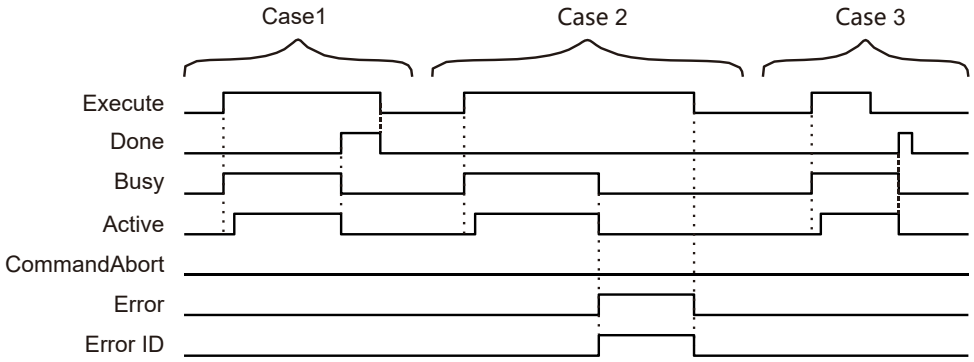
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує осями.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли кінцеві позиції досягнуті.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли осі починають керуватись інструкцією.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через два цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy та Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

Випадок 3 : після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

- **функція**

DMC_MoveLinearRelative використовується для групи осей для проведення лінійної інтерполяції, і можна керувати однією або декількома осями в групі осей. Прошивка V1.01 і вище підтримує цю функцію.

1. Вхідна *швидкість* інструкції DMC_MoveLinearRelative є цільовою швидкістю термінального приводу. Швидкість кінцевого приводу і швидкості осей мають наступне співвідношення. **Квадрат швидкості кінцевого приводу = сума квадратів швидкостей осей**
   Вхідні дані *Acceleration* і *Deceleration* DMC_MoveLinearRelative є цільовим прискоренням і цільовим уповільненням термінального приводу. Прискорення та уповільнення кінцевого приводу та прискорення та уповільнення осей мають таке співвідношення.
   Прискорення кінцевого приводу（уповільнення）= сума квадратів прискорення（уповільнення）осей
2. *Інструкція Jerk* DMC_MoveLinearRelative зарезервована.
3. Нижче наведено зв'язок між *BufferMode* , *TransitionMode* і *TransitionParameter* . Якщо mcBuffered вибрано як *BufferMode* , *TransitionMode* підтримує лише mcTMNone. Якщо mcBlendingPrevious вибрано як *BufferMode* , *TransitionMode* підтримує два режими: mcTMConstantVelocity та mcTMCornerDistance.

Значення BufferMode	Значення TransitionMode	опис
mcBuffered(1)	mcTMЖодного(0)	Зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію.
mcBlending-Previous(3)	mcTMConstant-Velocity(2)	Плавний перехід: зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію. Швидкість переходу є результуючою швидкістю попередньої інструкції. Після перемикання команд кінцевий привод виконує плавний перехід, а потім здійснює лінійний рух, як показано в наступному прикладі 2.
	mcTMCorner-Distance(3)	Перехід кутової відстані: зачекайте, доки завершиться виконання попередньої інструкції інтерполяції, а потім негайно виконайте поточну інструкцію. Коли відстань між положенням термінального приводу та кінцевим положенням попередньої інструкції інтерполяції дорівнює значенню *TransitionParameter* під час виконання попередньої інструкції інтерполяції, виконання попередньої інструкції інтерполяції завершується негайно, а потім починається виконання поточної інструкції. Під час виконання поточної інструкції термінальний актуатор переміщається по дузі кола, а потім виконує лінійну інтерполяцію. Відстань між кінцевою точкою дуги кола та кінцевою точкою попередньої інструкції інтерполяції дорівнює значенню *TransitionParameter* як показано в наступному прикладі 3.

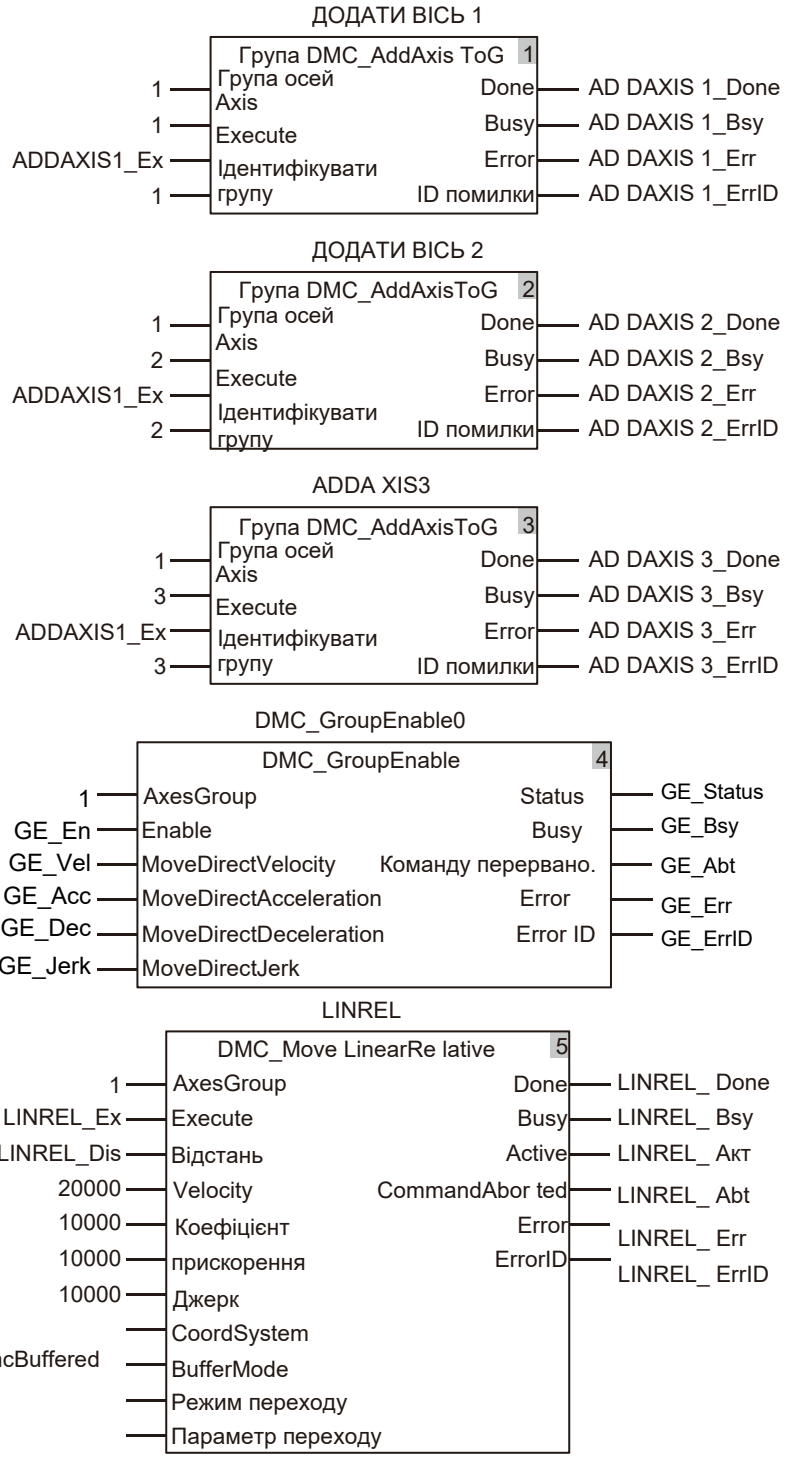
Приклад програмування 1

Приклад, у якому виконується одна інструкція DMC_MoveLinearRelative, виглядає наступним чином.

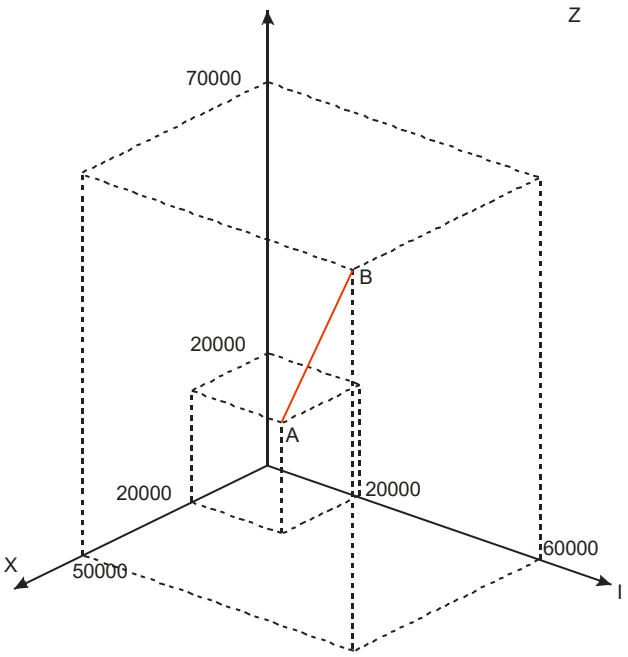
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	

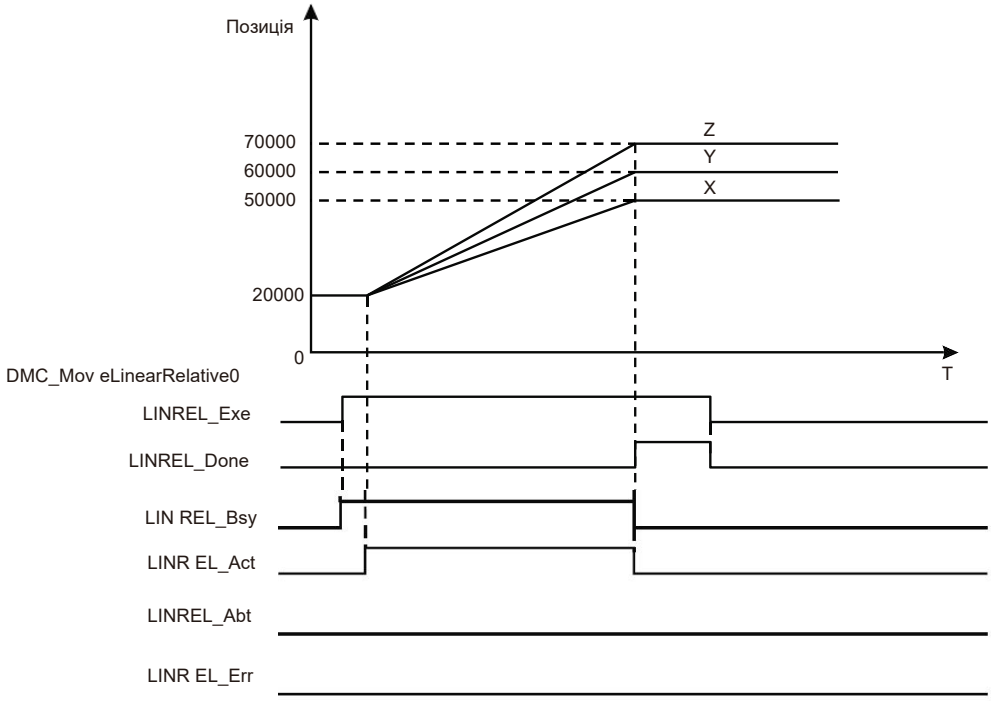
Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
ADDAXIS3	DMC_AddAxisToGroup	
ADDAXIS3_Done	BOOL	
ADDAXIS3_Bsy	BOOL	
ADDAXIS3_Err	BOOL	
ADDAXIS3_ErrID	WORD	
DMC_GroupEnable	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL	DMC_MoveLinearRelative	
LINREL_Ex	BOOL	
LINREL_Dis	ARRAY [1..8] LREAL	
LINREL_Done	BOOL	
LINREL_Bsy	BOOL	
LINREL_Act	BOOL	
LINREL_Abt	BOOL	
LINREL_Err	BOOL	
LINREL_ErrID	WORD	


ДОДАТИ ВІСЬ 1
Група DMC_AddAxis ToG
1
Група осей
Axis
Execute
Ідентифікувати
групу
Done
Busy
Error
ID помилки
1
1
ADDAXIS1_Ex
1
AD DAXIS 1_Done
AD DAXIS 1_Bsy
AD DAXIS 1_Err
AD DAXIS 1_ErrID
ДОДАТИ ВІСЬ 2
Група DMC_AddAxisToG
2
Група осей
Axis
Execute
Ідентифікувати
групу
Done
Busy
Error
ID помилки
1
2
ADDAXIS1_Ex
2
AD DAXIS 2_Done
AD DAXIS 2_Bsy
AD DAXIS 2_Err
AD DAXIS 2_ErrID
ADDA XIS3
Група DMC_AddAxisToG
3
Група осей
Axis
Execute
Ідентифікувати
групу
Done
Busy
Error
ID помилки
1
3
ADDAXIS1_Ex
3
AD DAXIS 3_Done
AD DAXIS 3_Bsy
AD DAXIS 3_Err
AD DAXIS 3_ErrID
DMC_GroupEnable0
DMC_GroupEnable
4
AxesGroup
Enable
MoveDirectVelocity
MoveDirectAcceleration
MoveDirectDeceleration
MoveDirectJerk
Status
Busy
Команду перервано.
Error
Error ID
1
GE_En
GE_Vel
GE_Acc
GE_Dec
GE_Jerk
GE_Status
GE_Bsy
GE_Abt
GE_Err
GE_ErrID
LINREL
DMC_Move LinearRe lative
5
AxesGroup
Execute
Відстань
Velocity
Коефіцієнт
прискорення
Джерк
CoordSystem
BufferMode
Режим переходу
Параметр переходу
Done
Busy
Active
CommandAbor ted
Error
ErrorID
1
LINREL_Ex
LINREL_Dis
20000
10000
10000
10000
mcBuffered
LINREL_ Done
LINREL_ Bsy
LINREL_ Акт
LINREL_ Abt
LINREL_ Err
LINREL_ ErrID

2. Перегляньте весь процес руху після виконання інструкції.



3. Крива руху вісь X, вісь Y, вісь Z і часова діаграма



- Початкові позиції осей X, Y і Z дорівнюють 20000. Для LINREL_Dis[1], LINREL_Dis[2] і LINREL_Dis[3] DMC_MoveLinearRelative встановлено значення 30 000, 40 000 і 50 000 відповідно.
- Коли LINREL_Ex змінюється на TRUE, LINREL_Bsy змінюється на TRUE. Через два цикли LINREL_Act змінюється на TRUE, і група осей починає працювати.
- Вісь X, вісь Y і вісь Z досягають відповідних цільових положень одночасно, LINREL_Done змінюється на TRUE, а LINREL_Bsy і LINREL_Act змінюються на FALSE, коли виконання інструкції DMC_MoveLinearRelative завершено.

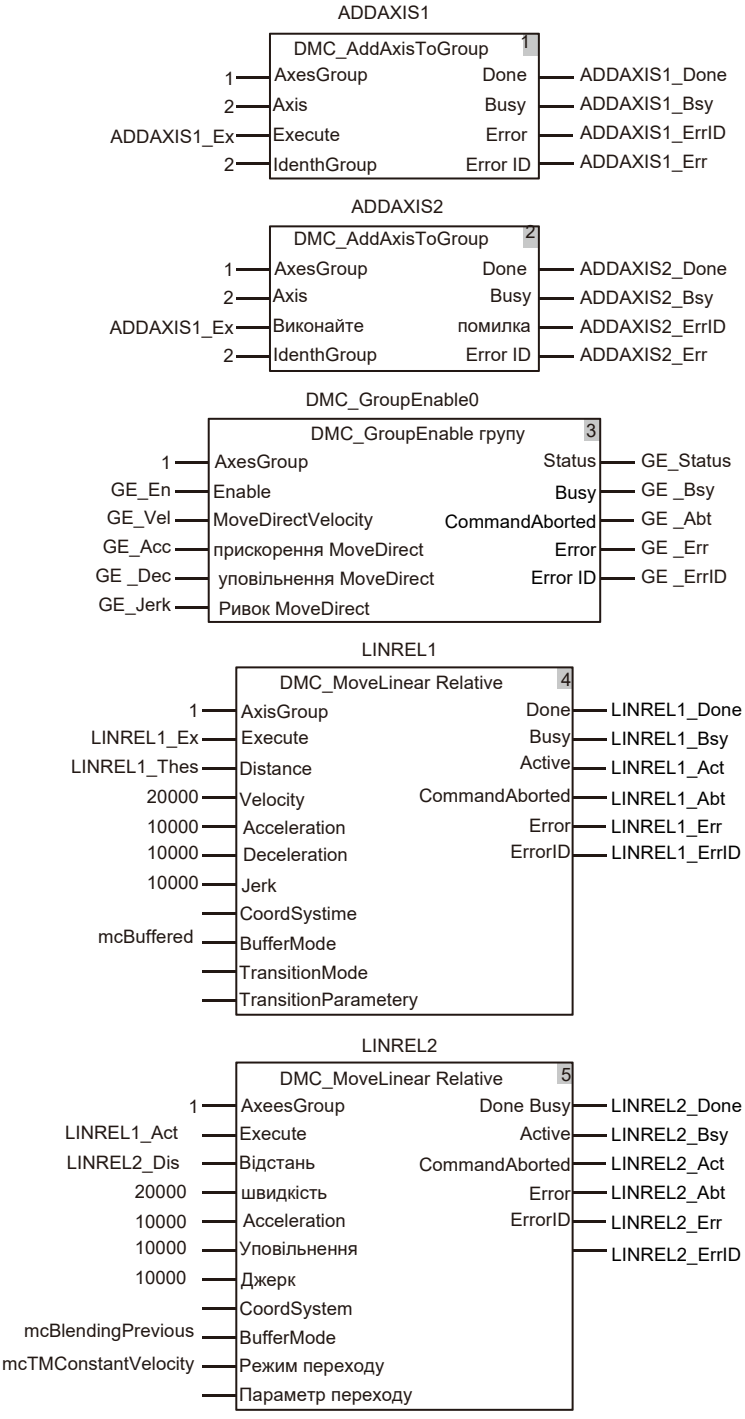
Приклад програмування 2

Подивіться наступний приклад, у якому є дві інструкції DMC_MoveLinearRelative і режим переходу між ними mcTMConstantVelocity.

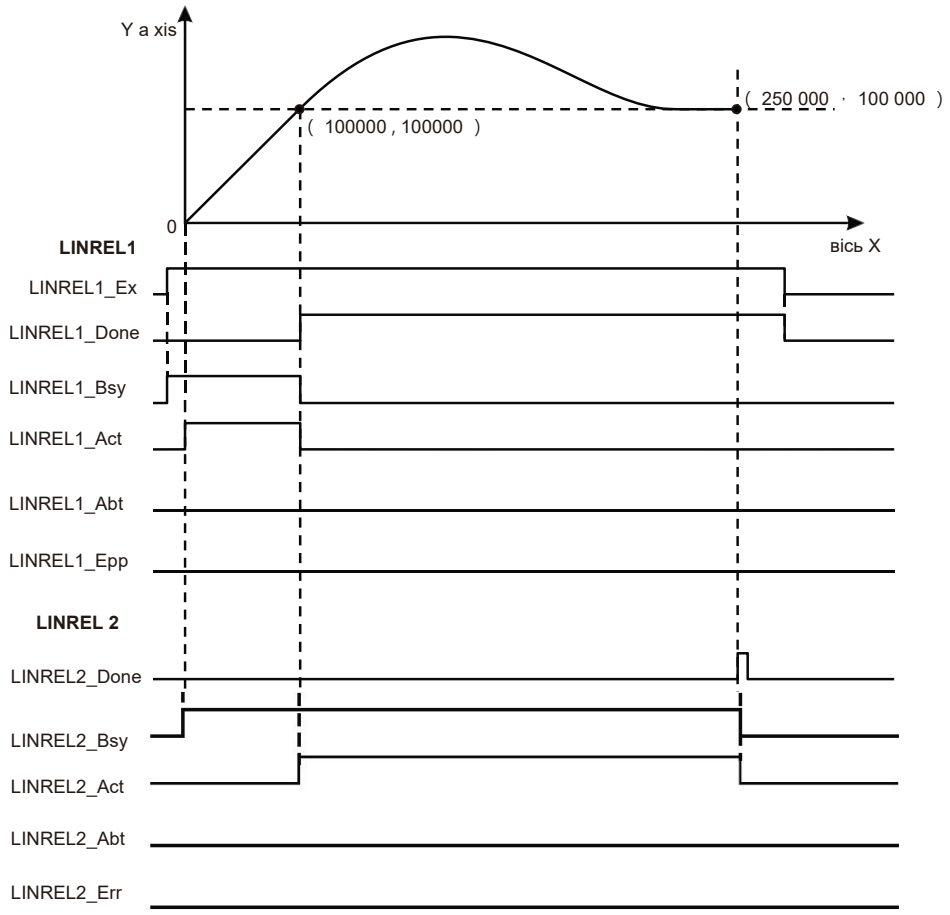
1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL1	DMC_MoveLinearRelative	
LINREL1_Ex	BOOL	
LINREL1_Dis	ARRAY [1..8] LREAL	
LINREL1_Done	BOOL	
LINREL1_Bsy	BOOL	
LINREL1_Act	BOOL	
LINREL1_Abt	BOOL	
LINREL1_Err	BOOL	
LINREL1_ErrID	WORD	
LINREL2	DMC_MoveLinearRelative	
LINREL2_Dis	ARRAY [1..8] LREAL	
LINREL2_Done	BOOL	

Ім'я змінної	Тип даних	Початкове значення
LINREL2_Bsy	BOOL	
LINREL2_Act	BOOL	
LINREL2_Abt	BOOL	
LINREL2_Err	BOOL	
LINREL2_ErrID	WORD	



2. Крива руху та часовий графік термінального приводу



- Встановіть BufferMode на mcBlendingPrevious і TransitionMode на mcTMConstantVelocityi для LINREL2.
- Для LINREL1_Dis [1] і LINREL1_Dis [2] для LINREL1 встановлено значення 100 000. LINREL2_Dis [1] і LINREL2_Dis [2] LINREL2 встановлені на 150 000 і 0.
- Спочатку виконуються ADDAXIS1 і ADDAXIS2, а потім виконується DMC_GroupEnable0. Коли групу осей увімкнено, виконується LINREL1, а потім негайно виконується LINREL2.
- Коли виконавчий механізм термінала досягає (100 000, 100 000) у системі координат, LINREL1_Done змінюється на TRUE. Тим часом LINREL2_Act змінюється на TRUE і LINREL2 починає виконуватися. На даний момент швидкість термінального приводу є цільовою швидкістю попередньої інструкції, 20000.
- Після виконання LINREL2 виконавчий механізм виконує плавний перехід, а потім здійснює лінійний рух. Виконання команди завершується, коли виконавчий механізм терміналу досягає координат (250 000 100 000).

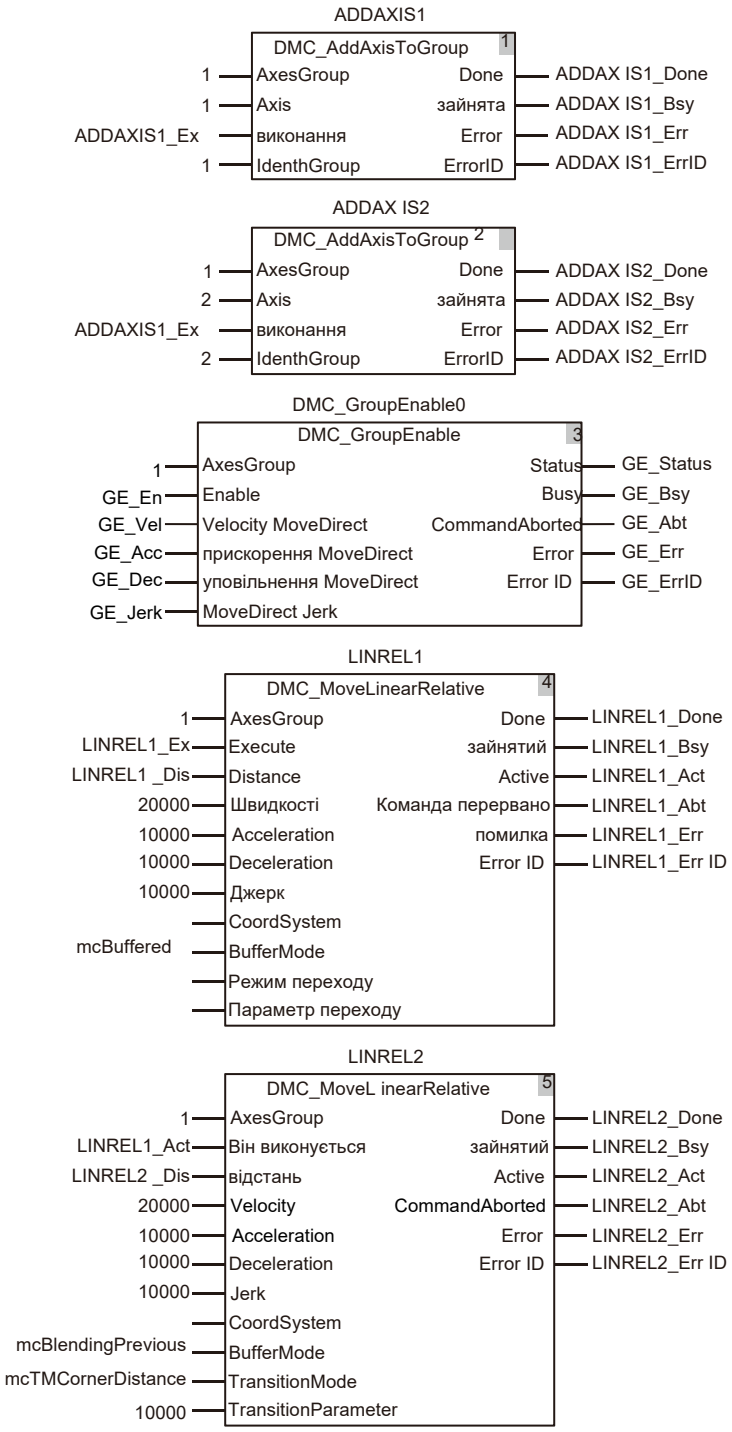
Приклад програмування 3

Приклад, у якому є дві інструкції DMC_MoveLinearRelative і режим переходу між ними mcTMCornerDistance, виглядає наступним чином.

1. Таблиця змінних і програма

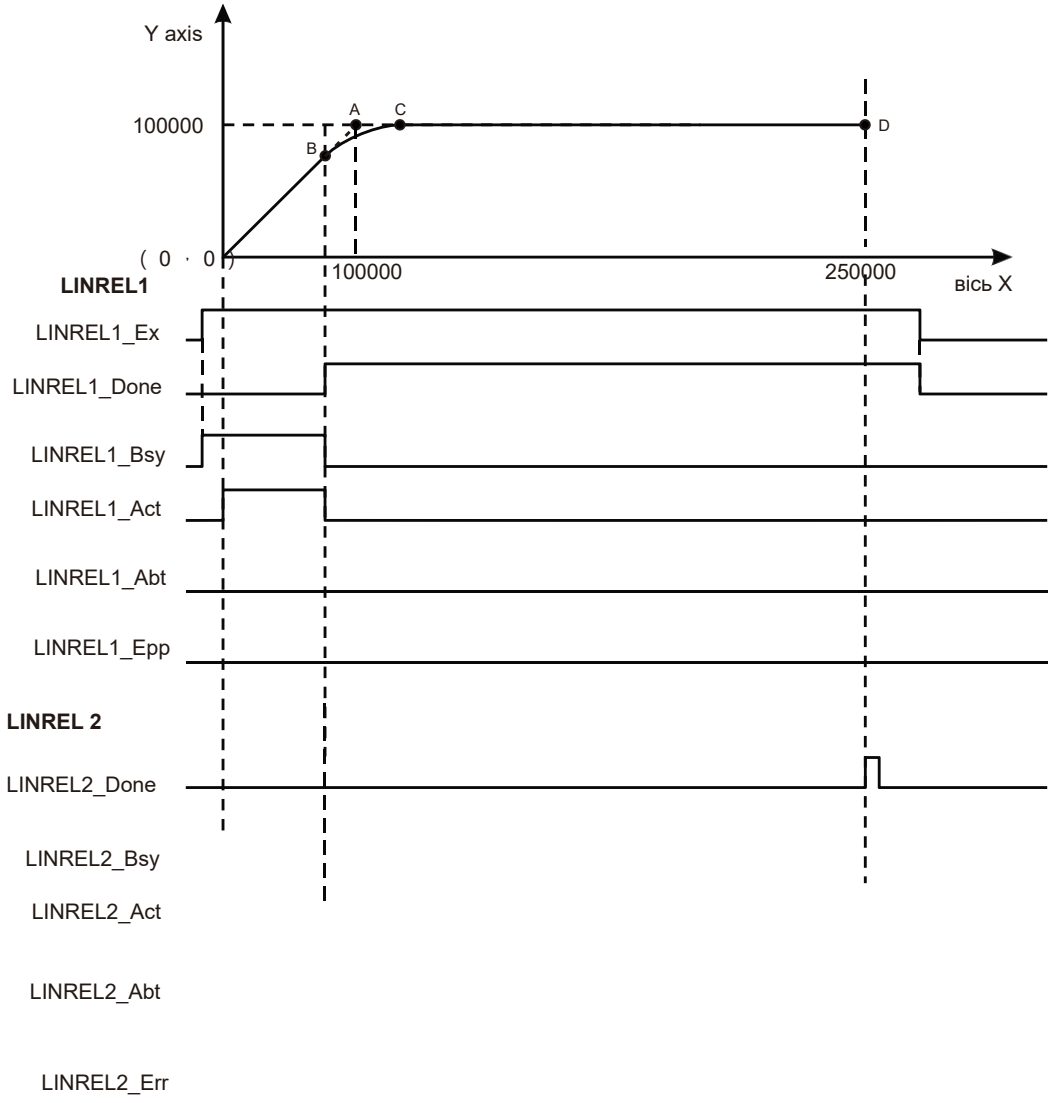
Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
LINREL1	DMC_MoveLinearRelative	
LINREL1_Ex	BOOL	
LINREL1_Dis	ARRAY [1..8] LREAL	
LINREL1_Done	BOOL	
LINREL1_Bsy	BOOL	
LINREL1_Act	BOOL	
LINREL1_Abt	BOOL	
LINREL1_Err	BOOL	
LINREL1_ErrID	WORD	
LINREL2	DMC_MoveLinearRelative	
LINREL2_Dis	ARRAY [1..8] LREAL	
LINREL2_Done	BOOL	
LINREL2_Bsy	BOOL	
LINREL2_Act	BOOL	
LINREL2_Abt	BOOL	
LINREL2_Err	BOOL	
LINREL2_ErrID	WORD	


ADDAXIS1
DMC_AddAxisToGroup
1
1
AxesGroup
1
Axis
ADDAXIS1_Ex
виконання
1
IdenthGroup
Done
зайнята
Error
ErrorID
ADDAX IS1_Done
ADDAX IS1_Bsy
ADDAX IS1_Err
ADDAX IS1_ErrID
ADDAX IS2
DMC_AddAxisToGroup
2
1
AxesGroup
2
Axis
ADDAXIS1_Ex
виконання
2
IdenthGroup
Done
зайнята
Error
ErrorID
ADDAX IS2_Done
ADDAX IS2_Bsy
ADDAX IS2_Err
ADDAX IS2_ErrID
DMC_GroupEnable0
DMC_GroupEnable
3
1
AxesGroup
GE_En
Enable
GE_Vel
Velocity MoveDirect
GE_Acc
прискорення MoveDirect
GE_Dec
уповільнення MoveDirect
GE_Jerk
MoveDirect Jerk
Status
Busy
CommandAborted
Error
Error ID
GE_Status
GE_Bsy
GE_Abt
GE_Err
GE_ErrID
LINREL1
DMC_MoveLinearRelative
4
1
AxesGroup
LINREL1_Ex
Execute
LINREL1 _Dis
Distance
20000
Швидкості
10000
Acceleration
10000
Deceleration
10000
Джерк
CoordSystem
mcBuffered
BufferMode
Режим переходу
Параметр переходу
Done
зайнятий
Active
Команда перервано
помилка
Error ID
LINREL1_Done
LINREL1_Bsy
LINREL1_Act
LINREL1_Abt
LINREL1_Err
LINREL1_Err ID
LINREL2
DMC_MoveL inearRelative
5
1
AxesGroup
LINREL1_Act
Він виконується
LINREL2 _Dis
відстань
20000
Velocity
10000
Acceleration
10000
Deceleration
10000
Jerk
CoordSystem
mcBlendingPrevious
BufferMode
mcTMCornerDistance
TransitionMode
10000
TransitionParameter
Done
зайнятий
Active
CommandAborted
Error
Error ID
LINREL2_Done
LINREL2_Bsy
LINREL2_Act
LINREL2_Abt
LINREL2_Err
LINREL2_Err ID

11

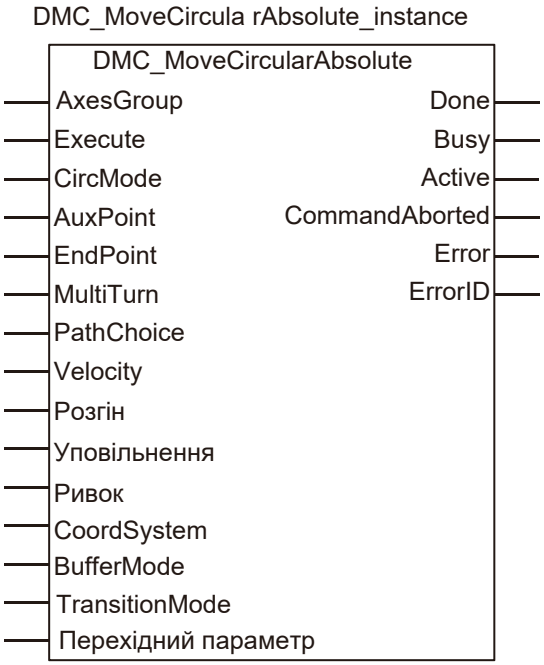
2. Крива руху та часовий графік термінального приводу



- Встановіть BufferMode на mcBlendingPrevious, TransitionMode на mcTMCornerDistance і TransitionParameter на 10000 для LINREL2.
- LINREL1_Dis [1] і LINREL1_Dis [2] LINREL1 обидва встановлені на 100 000. LINREL2_Dis [1] і LINREL2_Dis [2] LINREL2 встановлені на 150 000 і 0.
- Спочатку виконуються ADDAXIS1 і ADDAXIS2, а потім виконується DMC_GroupEnable0. Коли групу осей увімкнено, виконується LINREL1, а потім негайно виконується LINREL2.
- Коли виконавчий механізм терміналу потрапляє в точку B у системі координат, LINREL1_Done змінюється на TRUE. Тим часом LINREL2_Act змінюється на TRUE і LINREL2 починає виконуватися.
- Після виконання LINREL2 виконавчий механізм терміналу рухається по дузі, а потім продовжує виконувати лінійну інтерполяцію після проходження точки C.
- Відстань від точки B до точки A дорівнює відстані від точки C до точки A. Вона також дорівнює 10000, значенню *TransitionParameter* для LINREL2.

11.7.12 DMC_MoveCircularAbsolute

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveCircularAbsolute керує осями для виконання кругової інтерполяції. Кінцеві положення осей є абсолютними.	AS516E-B AS532EST-B AS564EST-B



● **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
CircMode	Встановити режим кругової інтерполяції 0: Намалюйте дугу через центр на площині XY. 1: Накресліть дугу через центр на площині ZX. 2: Накресліть дугу через центр на площині YZ. 3: Накресліть дугу через радіус на площині XY. 4: Накресліть дугу через радіус на площині ZX. 5: Накресліть дугу через радіус на площині YZ.	INT	0~5 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Допоміжна точка	Це координати центру, коли дуга проходить через центр. Auxpoint[1] означає радіус, а AuxPoint[2] не має сенсу, коли дуга малюється через радіус.	ARRAY [1..2] LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Кінцева точка	Положення кінцевої точки дуги на координатних осях	ARRAY [1..8] LREAL	Додатне число, 0, від'ємне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
MultiTurn	Встановіть кількість витків для спіральної інтерполяції	UINT	Позитивне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.
PathChoice	Напрямок для кругової інтерполяції 0: За годинниковою стрілкою 1: Проти годинникової стрілки	INT	0 , 1 (0)	Коли *Execute* змінюється з FALSE на TRUE.
швидкість	Встановіть максимальну швидкість (одиниця: одиниця/секунда)	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Прискорення	Встановити максимальне прискорення (Одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Встановіть максимальне уповільнення (Одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Встановіть максимальний ривок (Одиниця вимірювання: одиниця/секунда ^{3})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей. 1: Буферизовано 3: Змішування зі швидкістю попередньої інструкції	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	Коли *Execute* змінюється з FALSE на TRUE.
ПерехідМ ода	Вкажіть режим переходу між двома груповими інструкціями осей 0: криву переходу не вставлено 3: Зробіть перехід на основі вказаної кутової відстані	MC_Transition_Mode	0: mcTMНемає 3: mcTMCornerDistance	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
TransitionParameter	Встановіть параметр переходу для певного режиму переходу	LREAL	Позитивне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Інструкція DMC_MoveCircularAbsolute починає виконуватися, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.
2. Під час малювання дуги за допомогою методу центру значення AuxPoint[1] і AuxPoint[2] є різницею координат між центром і початковою точкою дуги кола. Під час малювання дуги за допомогою методу радіуса значення вхідного AuxPoint[1] є радіусом. Значення AuxPoint[2] не має сенсу.
3. Значення вхідного параметра EndPoint[1]~EndPoint[8] означає координати кінцевої точки дуги на осях.
4. Зверніться до розділу 10.2 щодо зв'язку між швидкістю, прискоренням, уповільненням і ривком.
5. Докладніше про BufferMode див. у розділі 10.3.

● **Вихідні параметри**

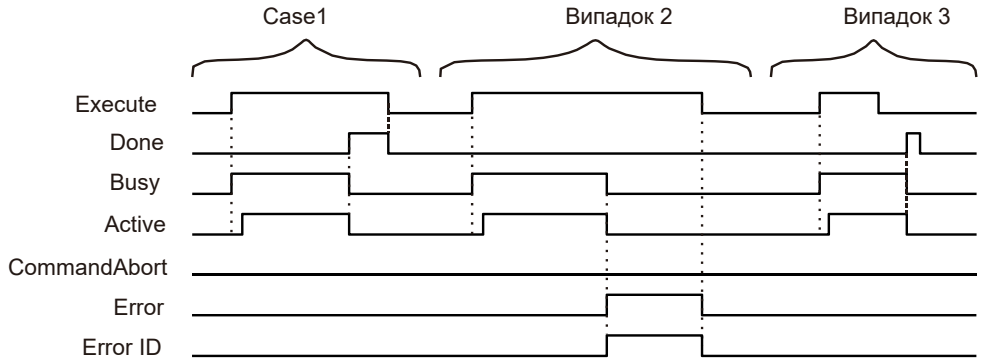
Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує осями.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

● **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли кінцеві позиції досягнуті.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли осі починають керуватись інструкцією.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через три цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy* та *Active* змінюються на FALSE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Busy* і *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

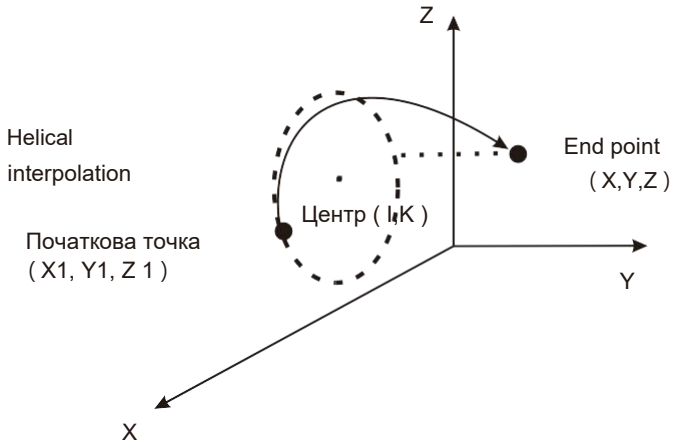
Випадок 3 : після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

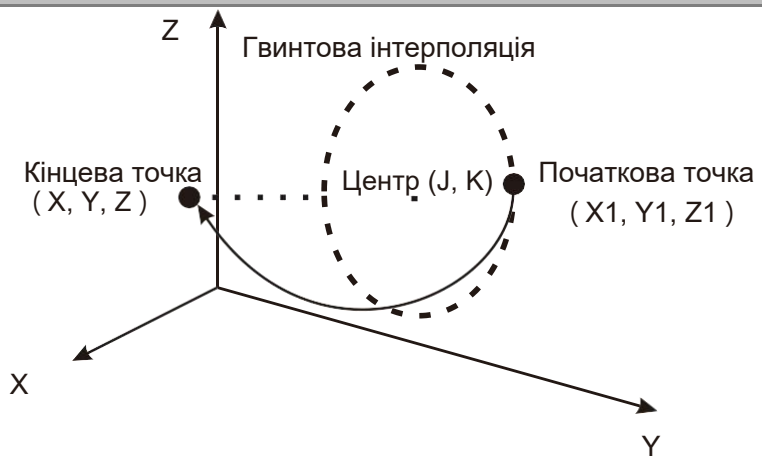
- **функція**

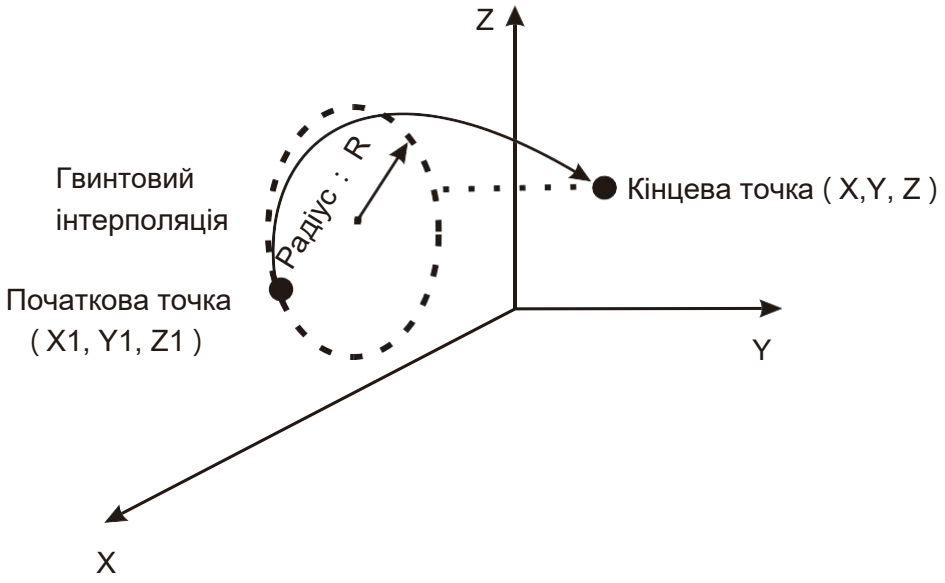
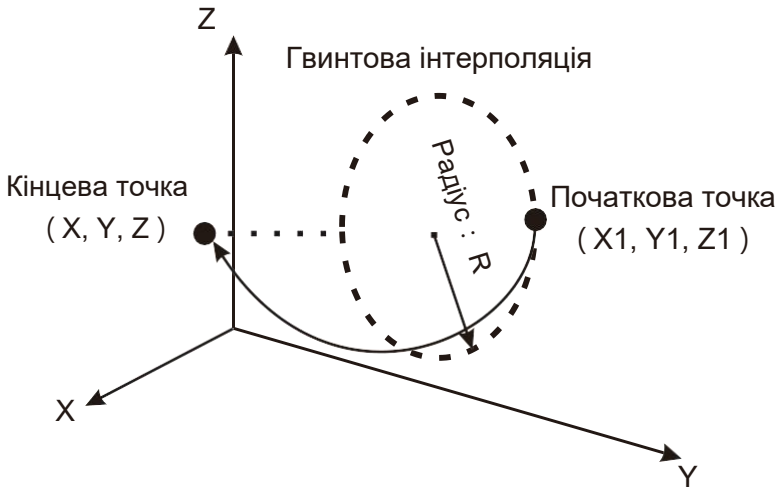
DMC_MoveCircularAbsolute використовується для осей для виконання кругової інтерполяції.

Прошивка V1.01 і вище підтримує цю функцію.

■CirMode (режим кругової інтерполяції)
Нижче наведено шість режимів CirMode.

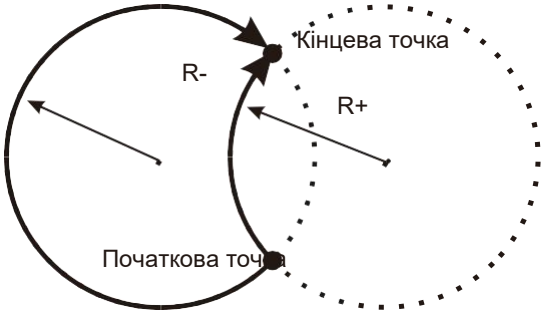
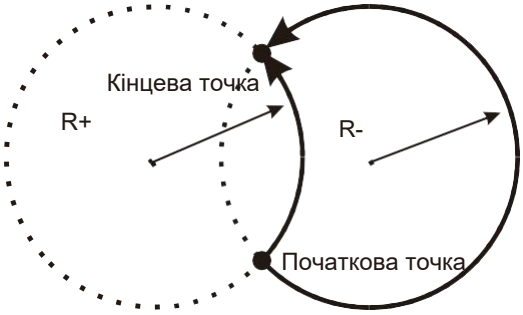
Значення CirMode	опис
0	Накресліть дугу через центр площини XY. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях X. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Y.  На малюнку вище координати центру: I=X1+AuxPoint[1],J=Y1+AuxPoint[2], координати кінцевої точки: X=EndPoint[1], Y=EndPoint[2], Z= Кінцева точка[3]
1	Накресліть дугу через центр на площині ZX. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях X. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Z.  На малюнку вище координати центру: I=X1+AuxPoint[1], K=Z1+AuxPoint[2], Координати кінцевої точки: X=Кінцева точка[1], Y=Кінцева точка[2], Z=Кінцева точка[3]
2	Накресліть дугу через центр площини YZ. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях Y. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Z.

Значення CirMode	опис
	 На малюнку вище координати центру: J=Y1+AuxPoint[1], K=Z1+AuxPoint[2] Координати кінцевої точки: X=Кінцева точка[1], Y=Кінцева точка[2], Z=Кінцева точка[3]
3	Накресліть дугу через радіус на площині XY. Значення AuxPoint[1] є радіусом кола на площині XY, а значення AuxPoint[2] не має сенсу. Якщо значення AuxPoint[1] більше 0, це мала дуга. Якщо значення AuxPoint[1] менше 0, це велика дуга .  На малюнку вище радіус: R=AuxPoint[1], Координати кінцевої точки: X=Кінцева точка[1], Y=Кінцева точка[2], Z=Кінцева точка[3]
4	Накресліть дугу через радіус на площині ZX. Значення AuxPoint[1] є радіусом кола на площині ZX, а значення AuxPoint[2] не має сенсу.

Значення CirMode	опис
	 На малюнку вище радіус: R=AuxPoint[1], Координати кінцевої точки: X=Кінцева точка[1]， Y=Кінцева точка[2]， Z=Кінцева точка[3]
5	Накресліть кола через радіус на площині ZY. Значення AuxPoint[1] є радіусом кола на площині ZY, а значення AuxPoint[2] не має сенсу.  На малюнку вище радіус: R=AuxPoint[1] Координати кінцевої точки: X=Кінцева точка[1], Y=Кінцева точка[2], Z=Кінцева точка[3].

■ PathChoice

Параметр визначає напрямок для кругової інтерполяції. Подробиці див. нижче.

PathChoice Значення	опис
0	Група осей проводить кругову інтерполяцію за годинниковою стрілкою на дугах заданої площини. Візьмемо, наприклад, метод радіуса, шлях для кругового руху показано нижче.  Рух за годинниковою стрілкою
1	Група осей проводить кругову інтерполяцію проти годинникової стрілки на дугах заданої площини. Візьмемо, наприклад, радіусний метод, шлях для кругового руху показаний нижче.  Рух проти годинникової стрілки

■ BufferMode

Вкажіть режим буфера між поточною інструкцією та попередньою інструкцією інтерполяції. Подробиці див. нижче.

Значення BufferMode	опис
mcBuffered （ 1 ）	Поточна інструкція чекатиме і не буде виконана, доки не завершиться виконання попередньої інструкції інтерполяції.
mcBlendingПопередній （ 3 ）	Перехід між поточною інструкцією та попередньою інструкцією інтерполяції здійснюється в режимі, визначеному *TransitionMode* . BufferMode DMC_MoveCircularRelative може вибрати цей режим лише тоді , коли поточна інструкція є інструкцією лінійної інтерполяції.

Приклад програмування

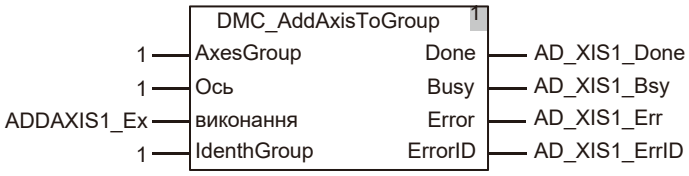
Приклад виконання однієї інструкції DMC_MoveCircularAbsolute показано наступним чином.

1. Таблиця змінних і програма

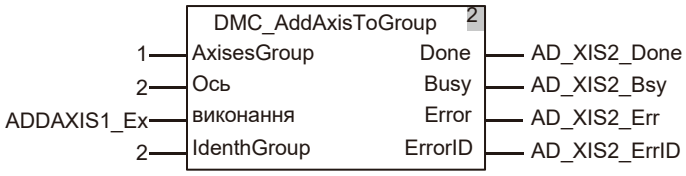
Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	
ADDAXIS1_ ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ ErrID	WORD	
CIRABS	DMC_MoveCircularAbsolute	
CIRABS_Ex	BOOL	
CIRABS_Mode	INT	0
CIRABS_Auxp	ARRAY [1..2] LREAL	
CIRABS_Endp	ARRAY [1..8] LREAL	
CIRABS_T	UINT	0
CIRABS_PC	INT	0
CIRABS_Done	BOOL	
CIRABS_Bsy	BOOL	
CIRABS_Act	BOOL	
CIRABS_Abt	BOOL	
CIRABS_Err	BOOL	
CIRABS_ErrID	WORD	

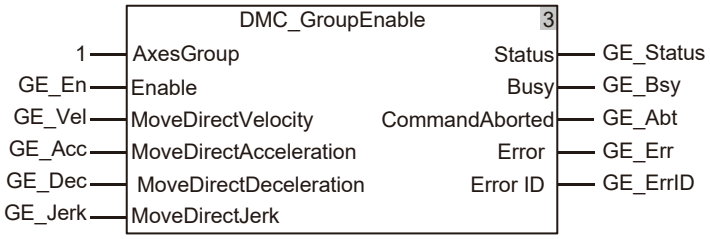
A DD A XIS 1



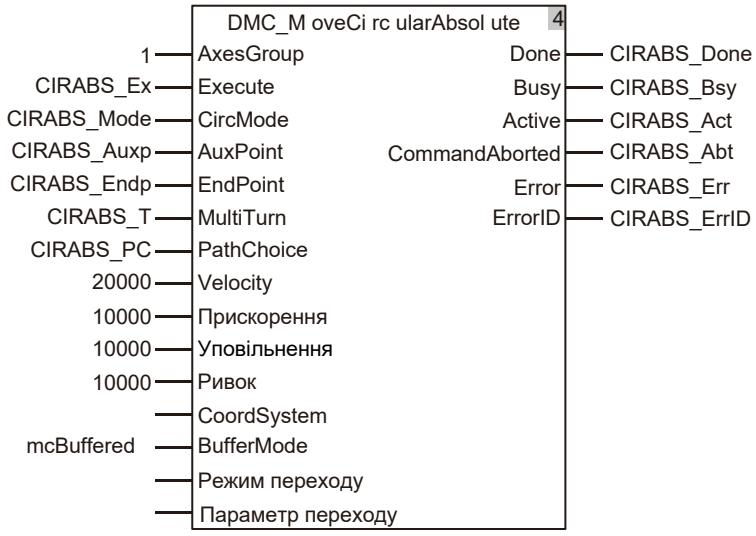
A DD ОСЬ 2



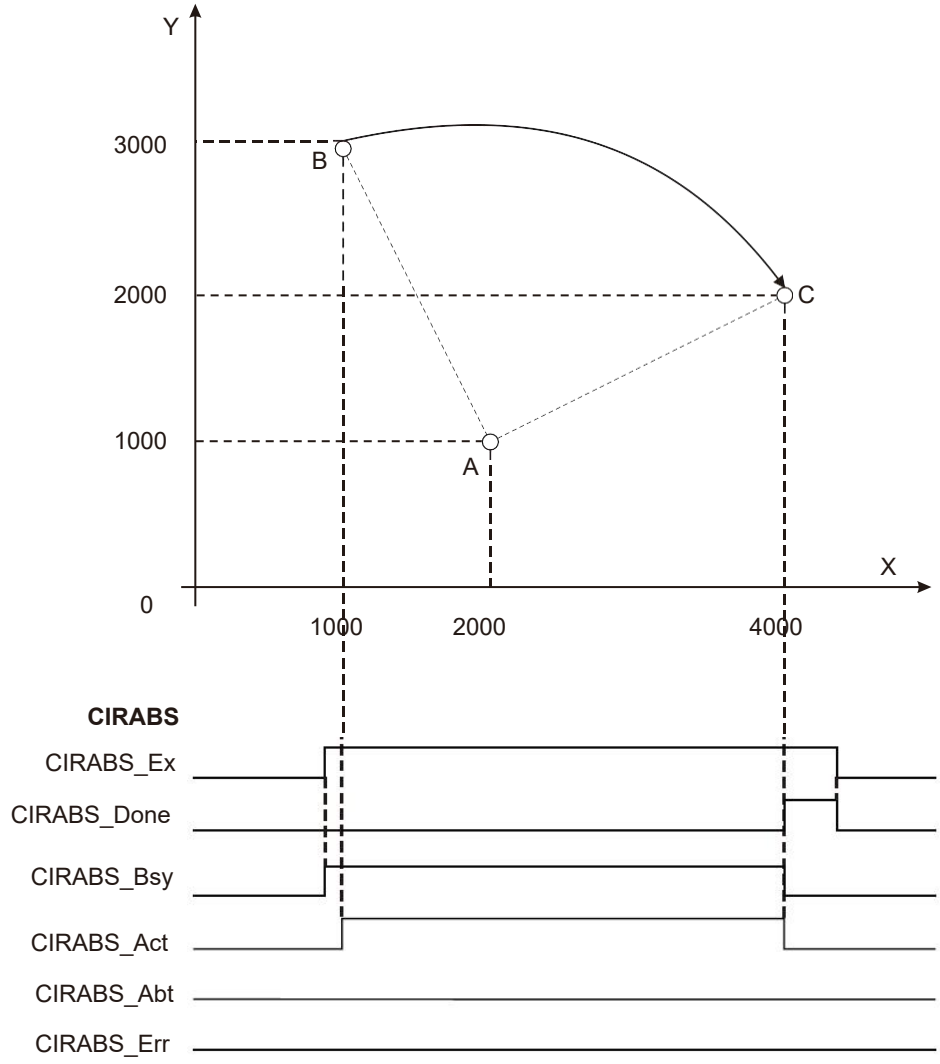
DMC_GroupEnable0



CIRABS



2. Крива руху в системі координат XY і часова діаграма

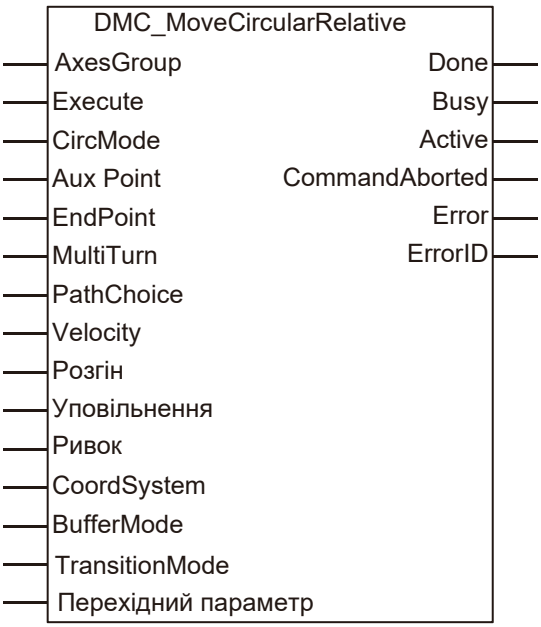


- Значення CIRABS_ Auxp[1] встановлено на 1000, значення CIRABS_Auxp[2] встановлено на -2000, значення CIRABS_Endp[1] встановлено на 4000, значення CIRABS_Endp[2] встановлено на 2000 і значення CIRABS_Mode встановлено на 0.
- Точка B є початковою точкою дуги з координатами (1000, 3000). Точка A є центром кола, де знаходиться дуга з координатами (1000+CIRABS_Auxp[1]=2000, 3000+CIRABS_Auxp[2]=1000). Точка C є кінцевою точкою дуги з координатами (CIRABS_Endp[1]=4000, CIRABS_Endp[2]=2000).
- Після виконання інструкції DMC_MoveCircularAbsolute виконується кругова інтерполяція за годинниковою стрілкою, починаючи з точки B і вважаючи точку A центром. Виконання інструкції закінчено, коли досягнуто точки C.

11.7.13 DMC_MoveCircularRelative

FB/FC	Пояснення	Застосовна модель
FB	DMC_MoveCircularRelative керує осями для виконання кругової інтерполяції. Кінцеві положення осей є взаємними.	AS516E-B AS532EST-B AS564EST-B

DMC_MoveCircularRelative_instance



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE.
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
CircMode	Встановити режим кругової інтерполяції. 0: Намалюйте дугу через а центр на площині XY. 1: Накресліть дугу через центр на площині ZX. 2: Накресліть дугу через центр на площині YZ. 3: Накресліть дугу через радіус на площині XY. 4: Накресліть дугу через радіус на площині ZX. 5: Накресліть дугу через радіус на площині YZ.	INT	0~5 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AuxPoint	Це координати центру, коли дуга проходить через центр. Auxpoint[1] означає радіус, а AuxPoint[2] не має сенсу, коли дуга малюється через радіус.	ARRAY [1..2] LREAL	Додатне число, 0, від'ємне число (0)	Коли Execute змінюється з FALSE на TRUE.
Кінцева точка	Різниці координат на осях між кінцевою та початковою точками дуги	ARRAY [1..8] LREAL	Додатне число, 0, від'ємне число (0)	Коли Execute змінюється з FALSE на TRUE.
MultiTurn	Встановіть кількість витків для спіральної інтерполяції	UINT	Позитивне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.
PathChoice	Напрямок для кругової інтерполяції 0: за годинниковою стрілкою 1: Проти годинникової стрілки	INT	0, 1 (0)	Коли *Execute* змінюється з FALSE на TRUE.
швидкість	Встановіть максимальну швидкість (Одиниця: одиниця/секунда)	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Прискорення	Встановіть максимальне прискорення (Одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
Уповільнення	Встановіть максимальне уповільнення (одиниця: одиниця/секунда ^{2})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
ривок	Встановіть максимальний ривок (Одиниця вимірювання: одиниця/секунду ^{3})	LREAL	Додатне число (0)	Коли *Execute* змінюється з FALSE на TRUE.
CoordSystem	Зарезервований			
BufferMode	Вкажіть режим буфера між двома інструкціями групи осей. 1: Буферизовано 3: Змішування зі швидкістю попередньої інструкції	MC_Buffer_Mode	1: mcBuffered 3: mcBlendingPrevious	Коли *Execute* змінюється з FALSE на TRUE.

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
TransitionMode	Вкажіть режим переходу між двома груповими інструкціями осей 0: криву переходу не вставлено 3: Зробіть перехід на основі вказаної кутової відстані	MC_Transition_Mode	0: mcTMНемає 3: mcTMCornerDistance	Коли *Execute* змінюється з FALSE на TRUE.
Параметр переходу	Встановіть параметр переходу для певного режиму переходу	LREAL	Позитивне число, 0 (0)	Коли *Execute* змінюється з FALSE на TRUE.

Примітка:

1. Інструкція DMC_MoveCircularRelative починає виконуватися, коли *Execute* змінюється з FALSE на TRUE. Зміна *Execute* з TRUE на FALSE під час виконання інструкції не матиме впливу на виконання інструкції.
2. Під час малювання дуги за допомогою методу центру значення AuxPoint[1] і AuxPoint[2] є різницею координат між центром і початковою точкою дуги кола. Під час малювання дуги за допомогою методу радіуса значення вхідного AuxPoint[1] є радіусом. Значення AuxPoint[2] не має сенсу.
3. Значення вхідного параметра EndPoint[1]~EndPoint[8] означає різницю координат між кінцевою та початковою точками на осях.
4. Зверніться до розділу 10.2 щодо зв'язку між швидкістю, прискоренням, уповільненням і ривком.
5. Докладніше про BufferMode див. у розділі 10.3.

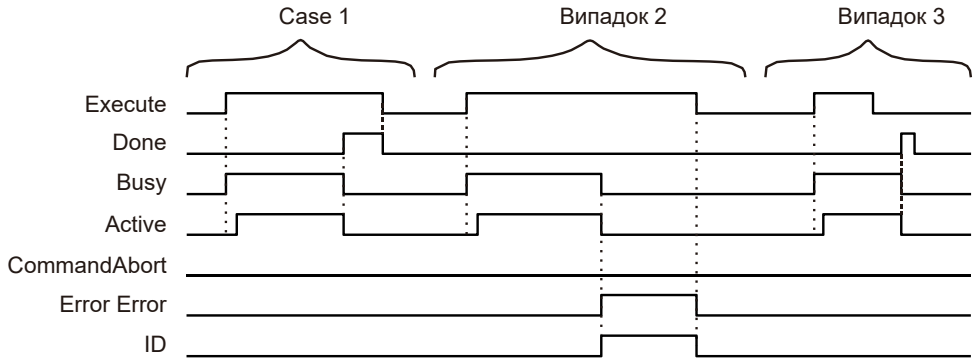
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли виконання інструкції завершено.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Active	TRUE, коли інструкція керує групою осей.	BOOL	TRUE / FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли кінцеві позиції досягнуті.	◆ Коли *Execute* змінюється з TRUE на FALSE після завершення виконання інструкції ◆ *Done* змінюється на TRUE, коли виконання інструкції завершено після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції. Через один цикл *Done* змінюється на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
Active	◆ Коли осі починають керуватись інструкцією.	◆ Коли *«Готово»* змінюється на TRUE ◆ Коли *помилка* змінюється на TRUE ◆ Коли *CommandAbort* змінюється на TRUE
CommandAbort	◆ Коли виконання інструкції переривається іншою інструкцією руху.	◆ Коли *Execute* змінюється з TRUE на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : Коли *Execute* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через три цикли *Active* змінюється на TRUE. Коли група осей досягає кінцевої позиції, *Done* змінюється на TRUE, *Busy* та *Active* змінюються на FALSE.

Випадок 2 : Коли *Execute* змінюється з FALSE на TRUE і виникає помилка (наприклад, помилка кінцевого автомата групи осей), *Error* змінюється на TRUE, а *ErrorID* показує відповідні коди помилок, а тим часом *Busy* та *Active* змінюються на FALSE. Коли *Execute* змінюється з TRUE на FALSE, *Error* змінюється на FALSE.

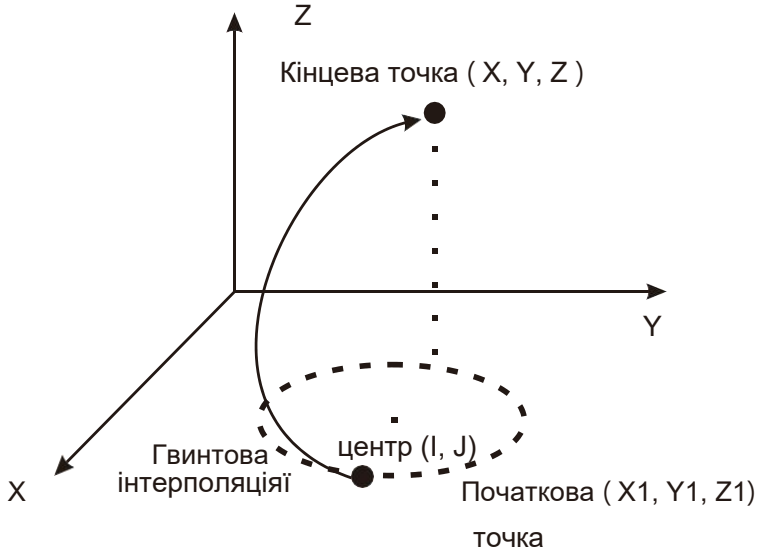
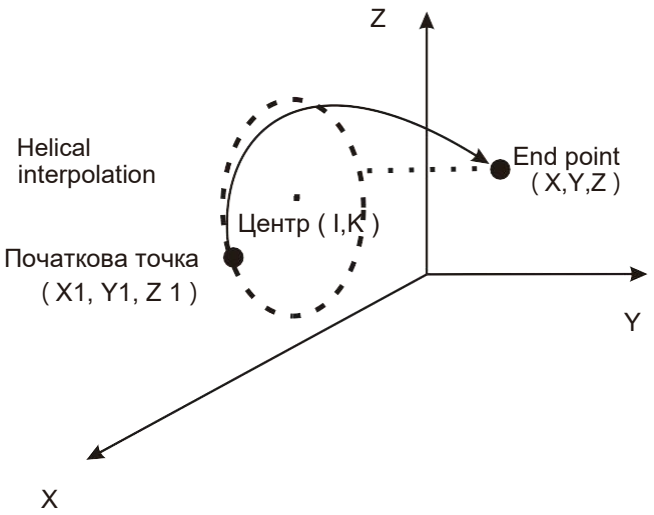
Випадок 3 : Після того, як *Execute* змінюється з TRUE на FALSE під час виконання інструкції, *Done* змінюється на TRUE, коли виконання інструкції завершено. Тимчасом *зайнятий* і *активний* змінюються на FALSE. Через один цикл *Done* змінюється на FALSE.

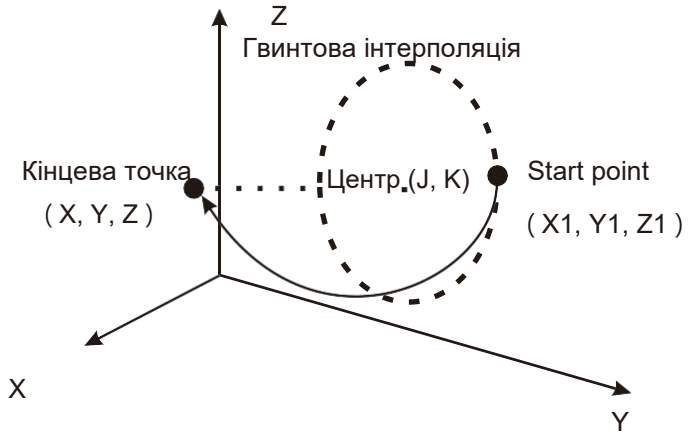
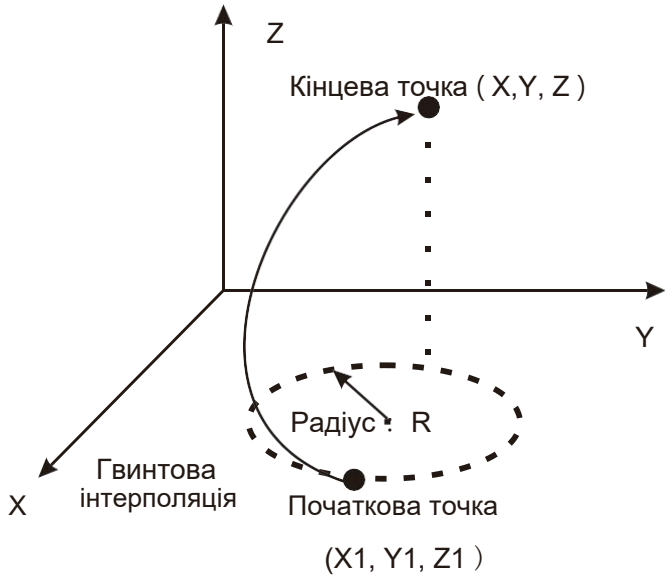
- **функція**

DMC_MoveCircularRelative використовується для осей для виконання кругової інтерполяції.
Прошивка V1.01 і вище підтримує цю функцію.

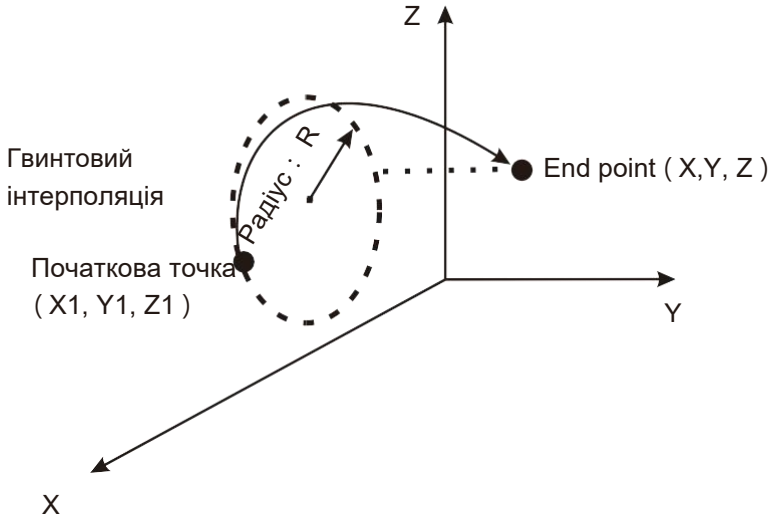
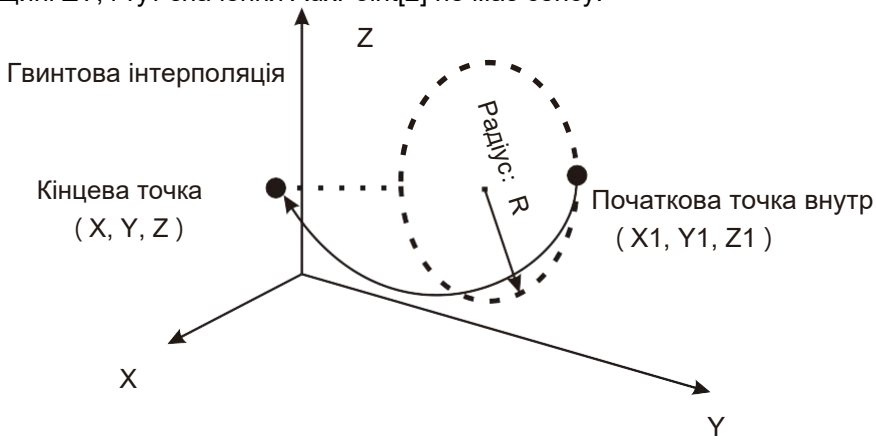
11

■CirMode (режим кругової інтерполяції) Нижче наведено шість режимів CirMode.

Значення CirMode	опис
0	Накресліть дугу через центр площини XY. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях X. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Y.  На малюнку вище Координати для центру: I=X1+AuxPoint[1], J=Y1+AuxPoint[2], Координати для кінцевої точки: X=X1+Кінцева точка[1], Y=Y1+Кінцева точка[2] ， Z=Z1+Кінцева точка[3].
1	Накресліть дугу через центр на площині ZX. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях X. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Z.  Координати для центру: I=X1+AuxPoint[1] ， K=Z1+AuxPoint[2] 。 Координати для кінцевої точки: X=X1+Кінцева точка[1], Y=Y1+Кінцева точка[2], Z=Z1+Кінцева точка[3].
2	Накресліть дугу через центр площини YZ. AuxPoint[1] — це значення зміщення центру на основі початкової точки на осях Y. AuxPoint[2] — це значення зміщення центру на основі початкової точки на осях Z.

Значення CirMode	опис
	 Координати для центру: J=Y1+AuxPoint[1] ， K=Z1+AuxPoint[2] 。 Координати для кінцевої точки: X=X1+EndPoint[1] ， Y=Y1+EndPoint[2] ， Z=Z1 +Кінцева точка[3] 。
3	Накресліть дугу через радіус на площині XY. Значення AuxPoint[1] — це радіус кола на площині XY, і тут значення AuxPoint[2] не має сенсу. Якщо значення AuxPoint[1] більше 0, це друга дуга. Якщо значення AuxPoint[1] менше 0, це велика дуга.  На малюнку вище радіус: R=AuxPoint[1], Координати кінцевої точки: X=X1+Кінцева точка[1], Y=Y1+Кінцева точка[2], Z=Z1+Кінцева точка[3].
4	Накресліть дугу через радіус на площині ZX. Значення AuxPoint[1] є радіусом кола на площині ZX, і тут значення AuxPoint[2] не має сенсу.

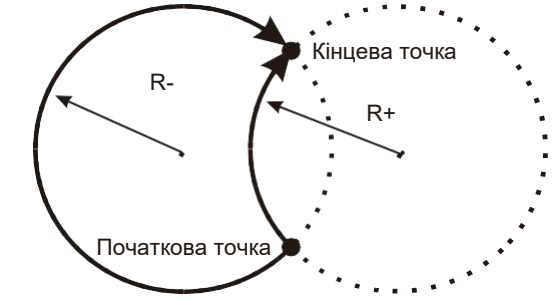
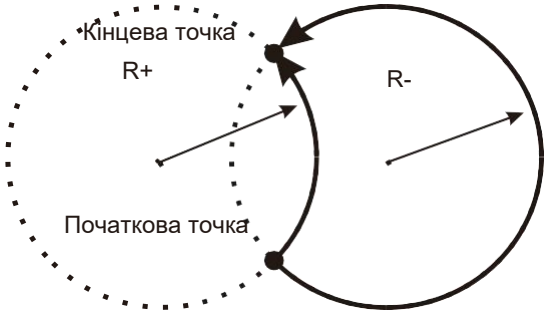
11

Значення CirMode	опис
	 На малюнку вище Радіус: R=AuxPoint[1] 。 Координати кінцевої точки: X=X1+Кінцева точка[1] ， Y=Y1+Кінцева точка[2] ， Z=Z1+Кінцева точка[3] 。
5	Накресліть дугу через радіус на площині ZY. Значення AuxPoint[1] є радіусом кола на площині ZY, і тут значення AuxPoint[2] не має сенсу.  На малюнку вище радіус: R=AuxPoint[1] Координати кінцевої точки: X=X1+Кінцева точка[1], Y=Y1+Кінцева точка[2], Z=Z1+Кінцева точка[3]

■ PathChoice

Параметр визначає напрямок для кругової інтерполяції. Подробиці дивіться нижче.

Значення PathChoice	опис
0	Група осей проводить кругову інтерполяцію за годинниковою стрілкою на дугах заданої площини. Візьмемо, наприклад, радіусний метод, шлях для кругового руху показаний нижче.

Значення PathChoice	опис
	 Рух за годинниковою стрілкою
1	Група осей проводить кругову інтерполяцію проти годинникової стрілки на дугах заданої площини. Візьмемо, наприклад, метод радіуса, шлях для кругового руху показано нижче.  Рух проти годинникової стрілки

■ BufferMode

Вкажіть режим буфера між поточною інструкцією та попередньою інструкцією інтерполяції. Подробиці див. нижче.

Значення BufferMode	опис
mcBuffered （ 1 ）	Поточна інструкція чекатиме і не буде виконана, доки не завершиться виконання попередньої інструкції інтерполяції.
mcBlendingПопередній （ 3 ）	Перехід між поточною інструкцією та попередньою інструкцією інтерполяції здійснюється в режимі, визначеному *TransitionMode* . BufferMode DMC_MoveCircularRelative може вибрати цей режим лише тоді , коли поточна інструкція є інструкцією лінійної інтерполяції.

Приклад програмування

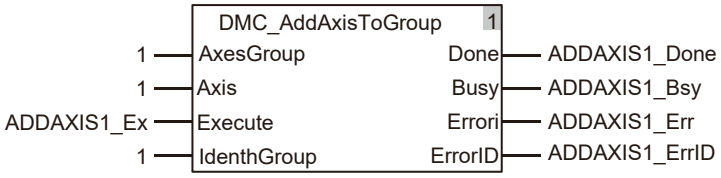
Приклад виконання однієї інструкції DMC_MoveCircularRelative показаний наступним чином.

1. Таблиця змінних і програма

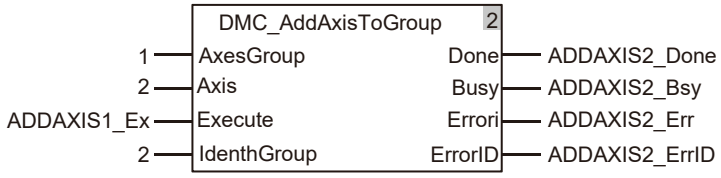
Ім'я змінної	Тип даних	Початкове значення
M1	BOOL	
ADDAXIS1	DMC_AddAxisToGroup	
ADDAXIS1_Ex	BOOL	
ADDAXIS1_Done	BOOL	
ADDAXIS1_Bsy	BOOL	
ADDAXIS1_Err	BOOL	

Ім'я змінної	Тип даних	Початкове значення
ADDAXIS1_ErrID	WORD	
ADDAXIS2	DMC_AddAxisToGroup	
ADDAXIS2_Done	BOOL	
ADDAXIS2_Bsy	BOOL	
ADDAXIS2_Err	BOOL	
ADDAXIS2_ ErrID	WORD	
DMC_GroupEnable0	DMC_GroupEnable	
GE_En	BOOL	
GE_Vel	ARRAY [1..8] LREAL	
GE_Acc	ARRAY [1..8] LREAL	
GE_Dec	ARRAY [1..8] LREAL	
GE_Jerk	ARRAY [1..8] LREAL	
GE_Status	BOOL	
GE_Bsy	BOOL	
GE_Abt	BOOL	
GE_Err	BOOL	
GE_ErrID	WORD	
CIRREL	DMC_MoveCircularRelative	
CIRREL_Ex	BOOL	
CIRREL_Mode	INT	0
CIRREL_Auxp	ARRAY [1..2] LREAL	
CIRREL_Endp	ARRAY [1..8] LREAL	
CIRREL_T	UINT	0
CIRREL_PC	INT	0
CIRREL_Done	BOOL	
CIRREL_Bsy	BOOL	
CIRREL_Act	BOOL	
CIRREL_Abt	BOOL	
CIRREL_Err	BOOL	
CIRREL_ErrID	WORD	

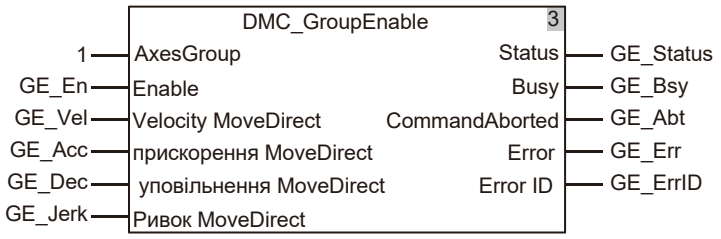
ADDAX IS1



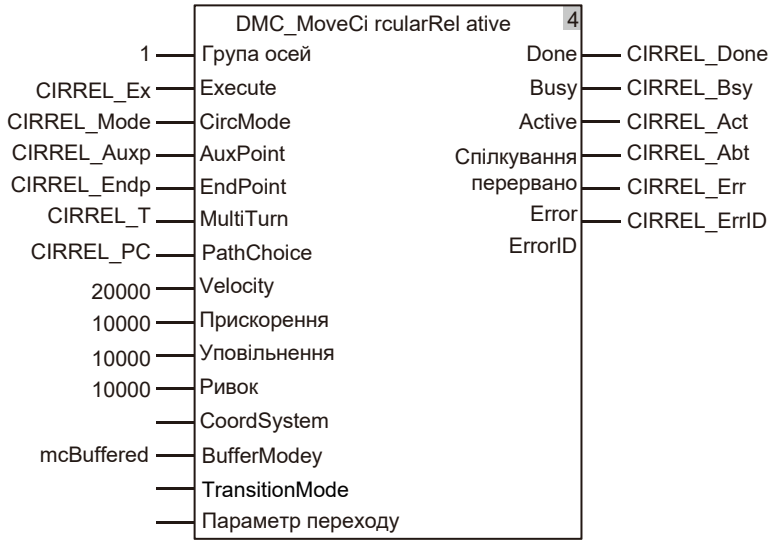
ADDAXIS2



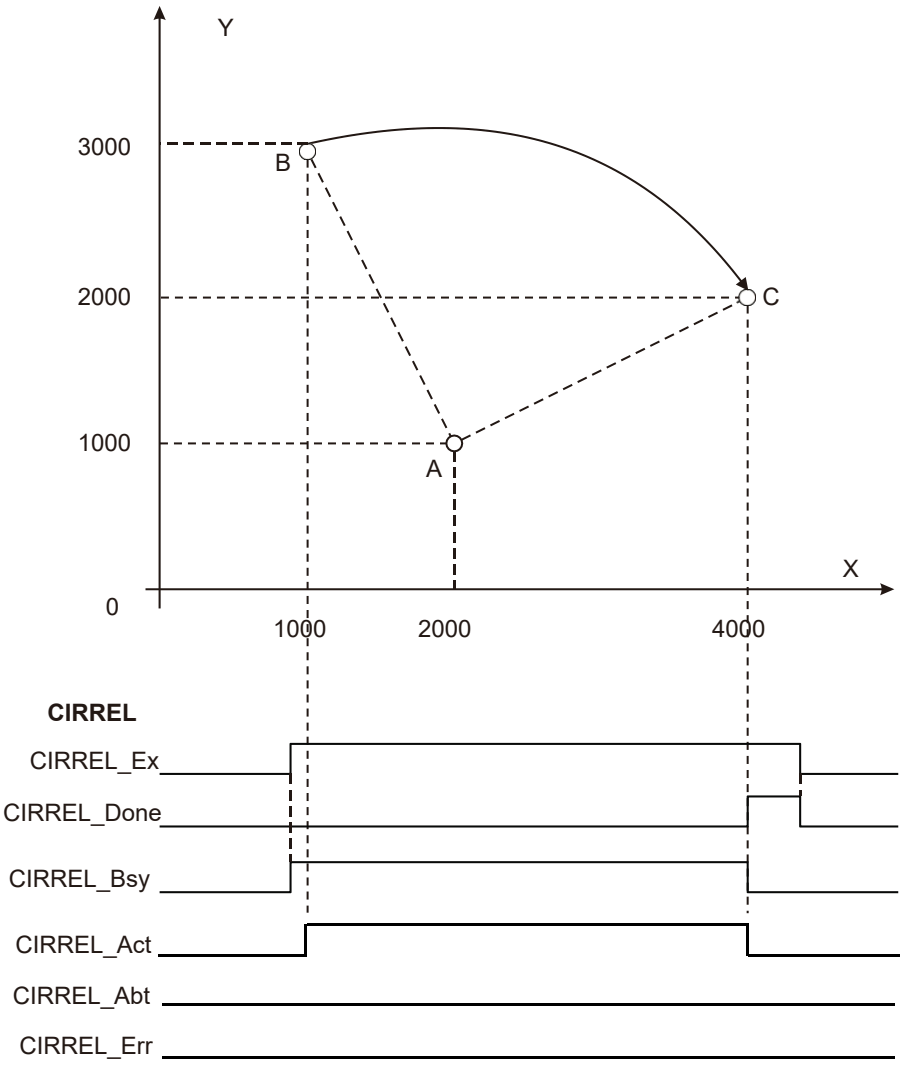
DMC_GroupEnable0



СІ ЗАЛІЗНИЦЯ



2. Крива руху в системі координат XY і часова діаграма

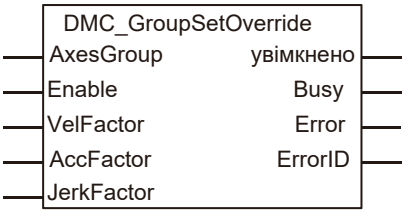


- Значення CIRREL_Mode встановлено на 0. Значення CIRREL_ Auxp[1] встановлено на 1000. Значення CIRREL_Auxp[2] встановлено на -2000. Значення CIRREL_Endp[1] має значення 3000. Значення CIRREL_Endp[2] має значення -1000. Значення CIRREL_Mode встановлено на 0.
- Точка B є початковою точкою дуги з координатами (1000, 3000). Точка A є центром кола, де знаходиться дуга з координатами (1000+CIRREL_Auxp[1]=2000, 3000+CIRREL_Auxp[2]=1000). Точка C є кінцевою точкою дуги з координатами (1000+CIRREL_Endp[1]=4000, 3000+CIRREL_Endp[2]=2000).
- Після виконання інструкції DMC_MoveCircularRelative виконується кругова інтерполяція за годинниковою стрілкою, починаючи з точки B і вважаючи точку A центром. Виконання інструкції закінчено, коли досягнуто точки C.

11.7.14 DMC_GroupSetOverride

FB/FC	Пояснення	Застосовна модель
FB	Інструкція DMC_GroupSetOverride використовується для встановлення значення перевизначення для скоординованого руху групи осей.	AS516E-B AS532EST-B AS564EST-B

DMC_ GroupSetOverride_insta nce



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
ВелФактор	Перевизначене значення, одиниця: %, наприклад, «100» означає 100%.	LREAL	0~500 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
AccFactor	Зарезервований	-	-	-
JerkFactor	Зарезервований	-	-	-

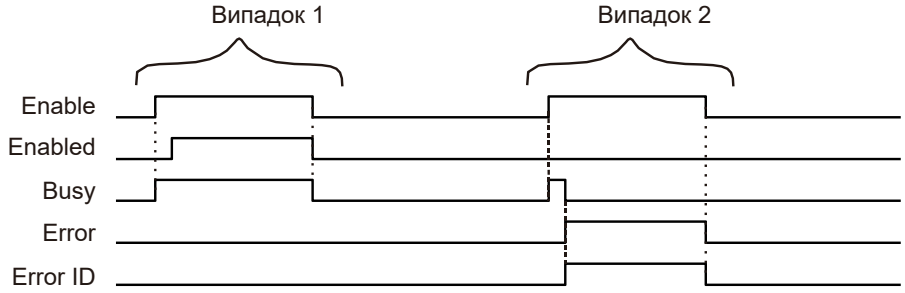
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Увімкнено	TRUE, коли група осей керується.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Увімкнено	◆ Коли інструкція починає виконуватися.	◆ Коли *Enable* змінюється на FALSE ◆ Коли *помилка* змінюється на TRUE
Busy	◆ Коли *Enable* змінюється на TRUE	◆ Коли *Enable* змінюється на FALSE ◆ Коли *помилка* змінюється на TRUE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один цикл *Enabled* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Enabled* і *Busy* змінюються на FALSE.

Випадок 2 : коли під час виконання інструкції виникає помилка, *Error* змінюється на TRUE, *ErrorID* показує відповідні коди помилок, а *Busy тим часом* змінюється на FALSE. Коли *Enable* змінюється на FALSE, *Error* змінюється на FALSE, а значення в *ErrorID* скидається на 0.

- **функція**

Інструкція DMC_GroupSetOverride використовується для встановлення значення перевизначення для скоординованого руху групи осей. Прошивка V1.01 і вище підтримує цю функцію.

1. Цільові швидкості цих інструкцій можна змінити, включаючи DMC_MoveDirectAbsolute, DMC_MoveDirectRelative, DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute і DMC_MoveCircularRelative.
2. Одиницею *VelFactor* є %. «100» означає «100%». Допустимий діапазон значення VelFactor становить 0~500. При перевищенні діапазону виникне помилка у виконанні інструкції.
3. Цільові швидкості осей у групі осей = цільові швидкості поточних осей * значення перевизначення, якщо інструкція DMC_MoveDirectAbsolute або DMC_MoveDirectRelative виконується після виконання інструкції DMC_GroupSetOverride.

   Нова максимальна швидкість групи осей = поточна максимальна швидкість групи осей* значення перевизначення, якщо інструкція DMC_MoveLinearAbsolute, DMC_MoveLinearRelative, DMC_MoveCircularAbsolute або DMC_MoveCircularRelative виконується після Інструкція DMC_GroupSetOverride виконана.
4. Після виконання інструкції DMC_GroupSetOverride група осей прискорюватиметься або сповільнюватиметься відповідно до швидкості прискорення чи уповільнення поточної інструкції, що виконується, до досягнення цільової швидкості після модифікації.
5. Error виникне в осях, якщо цільова швидкість після модифікації перевищує максимальну швидкість обертання.

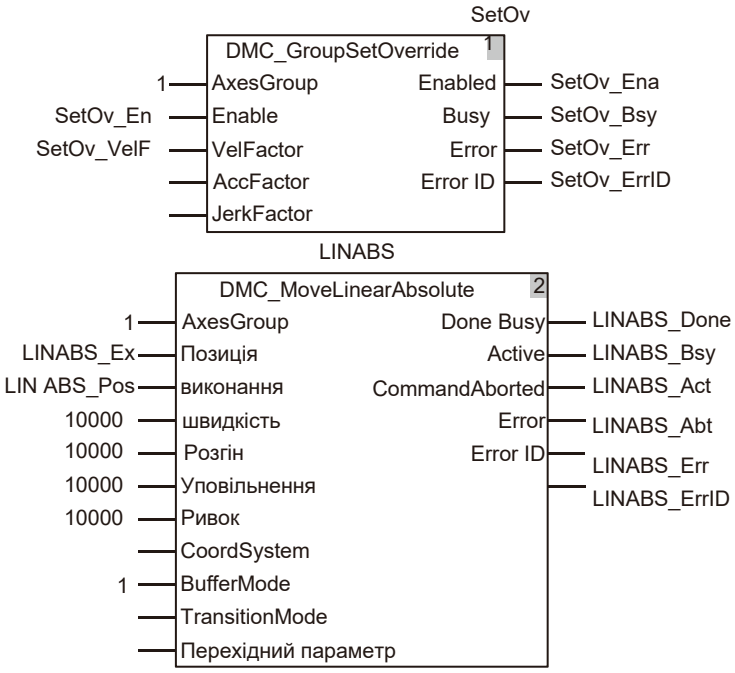
6. Цільова швидкість стає 0, а група осей діє зі швидкістю 0, коли значення *VelFactor* встановлено 0.
7. Коли *Enable* змінюється на TRUE, нещодавно змінене значення VelFactor набуде чинності негайно, і користувачам не потрібно перезапускати інструкцію. Коли *Enable* змінюється на TRUE, у інструкції виникне помилка, і цільова швидкість повернеться до 100%, якщо значення *VelFactor* перевищить допустимий діапазон.
8. Коли *Enable* змінюється на TRUE, змінене значення *Velfactor* набуде чинності негайно, і не буде потреби перезапускати інструкцію. Коли *параметр «Увімкнути»* змінюється на «TRUE», у інструкції буде негайно повідомлено про помилку, а цільова швидкість повернеться до 100%, якщо змінене значення *Velfactor* перевищить дійсний діапазон.
9. Коли *Enable* змінюється на FALSE, група осей прискорюватиметься або сповільнюватиметься, враховуючи VelFactor=100 як ціль.

Приклад програмування

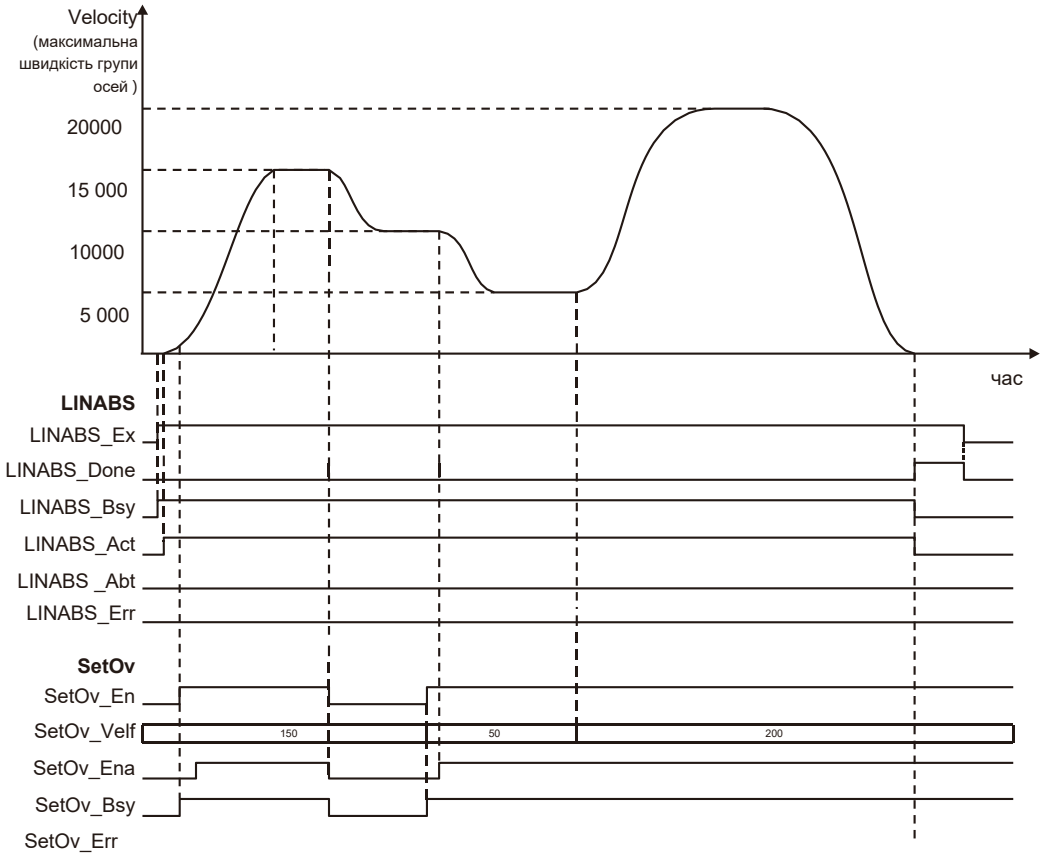
Приклад виконання інструкції DMC_GroupSetOverride виглядає наступним чином.

1. Таблиця змінних і програма

Ім'я змінної	Тип даних	Початкове значення
M1	BOOL	
SetOv	DMC_GroupSetOverride	
SetOv_En	BOOL	
SetOv_Velf	LREAL	
SetOv_Ena	BOOL	
SetOv_Bsy	BOOL	
SetOv_Err	BOOL	
SetOv_ErrID	WORD	
LINABS	DMC_MoveLinearAbsolute	
LINABS_Ex	BOOL	
LINABS_Pos	ARRAY [1..8] LREAL	
LINABS_Done	BOOL	
LINABS_Bsy	BOOL	
LINABS_Act	BOOL	
LINABS_Abt	BOOL	
LINABS_Err	BOOL	
LINABS_Id	WORD	



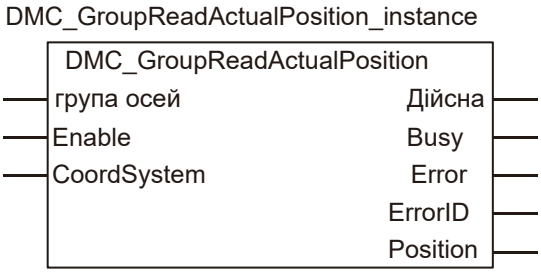
2. Крива руху та часовий графік



- Коли LINABS_Ex змінюється на TRUE, LINABS_Bsy змінюється на TRUE. Через два цикли LINABS_Act змінюється на TRUE, і група осей починає рухатися. Коли група осей не досягла цільової швидкості, а SetOv_En встановлено на TRUE, набуває чинності інструкція DMC_GroupSetOverride, і цільова швидкість групи осей стає новою цільовою швидкістю.
- Коли SetOv_En змінюється на FALSE, цільова швидкість групи осей стає 10000.
- Якщо значення SetOv_Velf змінено під час виконання інструкції DMC_GroupSetOverride, значення SetOv_Velf набуде чинності негайно. Цільова швидкість інструкції DMC_MoveLinearAbsolute відповідно зміниться.

11.7.15 DMC_GroupReadActualPosition

FB/FC	Пояснення	Застосовна модель
FB	Інструкцію DMC_GroupReadActualPosition можна використовувати для читання положення осей у групі осей.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
AxesGroup	Номер групи осей	USINT	1~8 (Потрібно встановити значення змінної)	Коли *Enable* змінюється на TRUE.
Enable	Інструкція виконується, коли *Enable* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
CoorSystem	Зарезервований	-	-	-

- **Вихідні параметри**

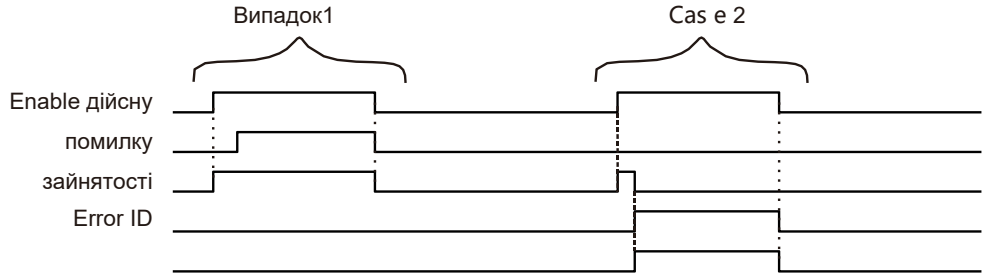
Назва параметра	функція	Тип даних	Допустимий діапазон
Vaild	TRUE, коли вихід інструкції дійсний.	BOOL	TRUE / FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE / FALSE
Error	TRUE, коли при виконанні інструкції виникає помилка.	BOOL	TRUE / FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Позиція	Розташування осей у групі осей	ARRAY [1..8] LREAL	

- **Час оновлення виводу**

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Vaild	◆ Коли інструкція читає позиції осей у групі осей	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється на TRUE
Busy	◆ Коли вказівка виконана	◆ Коли *Enable* змінюється з TRUE на FALSE ◆ Коли *помилка* змінюється на TRUE

Ім'я	Час зміни на TRUE	Час зміни на FALSE
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Enable* змінюється з TRUE на FALSE

- **Діаграма часу оновлення вихідних даних**



3. Коли *Enable* змінюється з FALSE на TRUE, *Busy* змінюється на TRUE. Через один цикл *Valid* змінюється на TRUE. Коли *Enable* змінюється з TRUE на FALSE, *Valid* і *Busy* змінюються на FALSE одночасно.
4. Коли в інструкції виникає помилка введення та *Enable* змінюється з FALSE на TRUE, Busy змінюється на TRUE, через один цикл *Error* змінюється на TRUE, а *ErrorID* показує коди помилок, а тим часом *Busy* змінюється на FALSE. Коли *Enable* змінюється з TRUE на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* очищається до 0.

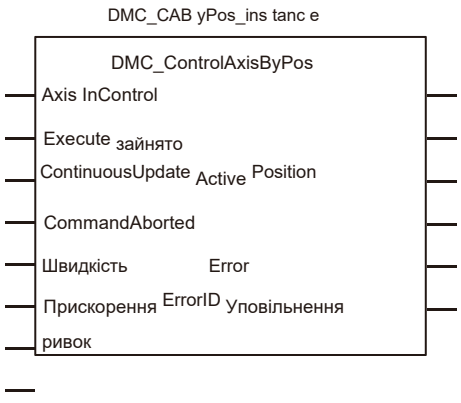
- **функція**

Інструкція DMC_GroupReadActualPosition використовується для читання поточних позицій осей у групі осей. Значення вихідної *позиції* є масивом. Кожен елемент масиву відповідає одній позиції осі. Наприклад, Position[1] відповідає положенню на осі X, Position[2] відповідає положенню на осі Y тощо. Прошивка V1.01 і вище підтримує цю функцію.

11.8 Інструкція з координації

11.8.1 DMC_ControlAxisByPos

FB/FC	Пояснення	Застосовна модель
FB	DMC_ControlAxisByPos керує рухом осі, надсилаючи поступове положення осі.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Axis	Вкажіть номер осі, якою потрібно керувати.	USINT	Див. "Осі № Діапазон, що Контролери Відповідають" . (Потрібно встановити значення змінної)	Коли *Execute* змінюється з FALSE на TRUE
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	–
ContinuousUpdate	Зарезервований	–	–	–
Позиція	Укажіть позицію збільшення.	LREAL	Додатне число, від'ємне число і 0 (0)	Коли *Execute* змінюється з FALSE на TRUE
швидкість	Вкажіть цільову швидкість	LREAL	Додатне число (Необхідно встановити значення змінної)	
Прискорення	Вкажіть цільове прискорення.	LREAL	Додатне число (Необхідно встановити значення змінної)	
Уповільнення	Вкажіть цільове уповільнення.	LREAL	Додатне число (Необхідно встановити значення змінної)	

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
ривок	Вкажіть швидкість зміни прискорення або уповільнення цілі.	LREAL	Додатне число (Необхідно встановити значення змінної)	

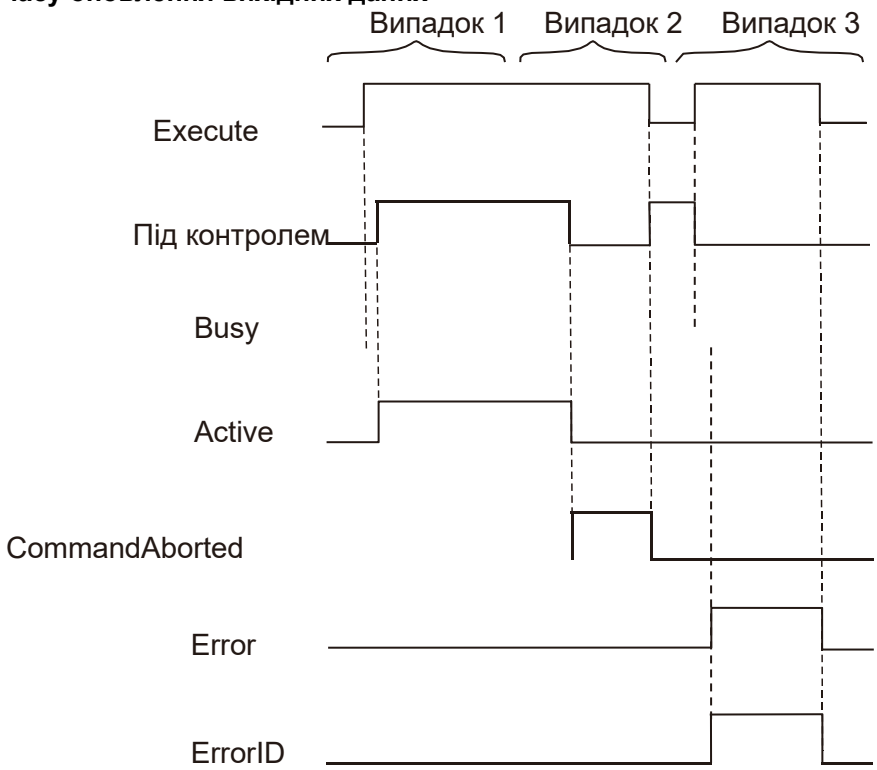
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Під контролем	TRUE, коли вісь контролюється інструкцією.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли вісь керується інструкцією.	BOOL	TRUE/FALSE
CommandAborted	TRUE, коли виконання інструкції перервано.	BOOL	TRUE/FALSE
Error	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	TRUE/FALSE
ErrorID	Містить коди помилок, коли виникає помилка. Будь ласка, зверніться до розділу 12.2 для відповідних кодів помилок.	WORD	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Під контролем	◆ Коли вісь керується інструкцією.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
Active	◆ Коли інструкція починає керувати віссю.	◆ Коли *CommandAborted* змінюється на TRUE. ◆ Коли *помилка* змінюється на TRUE.
CommandAborted	◆ Коли ця інструкція переривається в процесі виконання.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється на TRUE. Кілька циклів пізніше, *Active* і *InControl* змінюється на TRUE.

Випадок 2 : коли *Execute* змінюється з FALSE на TRUE і інструкція переривається іншою інструкцією, *CommandAborted* змінюється на TRUE, а тим часом *Busy* і *Active* змінюються на FALSE. *CommandAborted* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

Випадок 3 : коли виникає помилка, наприклад аварійний сигнал осі або Offline після того, як *Execute* змінюється з FALSE на TRUE, *Error змінюється* на TRUE, а *ErrorID* показує відповідні коди помилок. Тим часом *параметри «Зайнятий»* і *«Активний»* змінюються на «ХИБНІСТЬ», а *InControl* — на «ХИБНІСТЬ». *Error* змінюється на FALSE, коли *Execute* змінюється з TRUE на FALSE.

- **функція**

1. Інструкція DMC_ControlAxisByPos виконується шляхом зміни *Execute* з FALSE на TRUE. Зміна *значення Execute* інструкції з TRUE на FALSE під час виконання інструкції не впливає на виконання інструкції.
2. Хоча *Execute* знову змінюється з FALSE на TRUE під час виконання DMC_ControlAxisByPos, це все одно не впливає на виконання інструкції.
3. Швидкість, прискорення та уповільнення мають бути більшими за 0. Жодне з них не використовується в розрахунку контролю положення.
4. Позиція в цій інструкції означає відносну інкрементну позицію . Посилальна позиція для інкрементної позиції – це значення *Postion* , коли *Execute* інструкції змінюється з FALSE на TRUE.

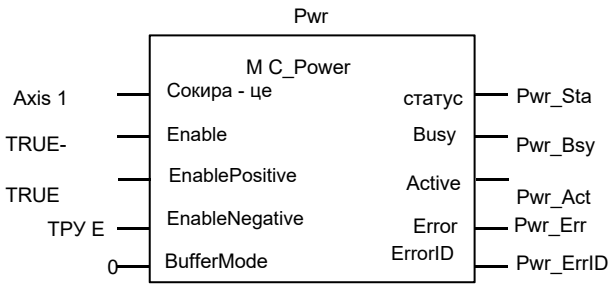
   Наприклад, значення *Position* — 5000, а позиція осі — 500. Змініть *Execute* інструкції з FALSE на TRUE. Тоді вісь залишається нерухомою, а положення осі – незмінним. Змініть значення *Position* на 6000 (збільшення на 1000), вісь завершена

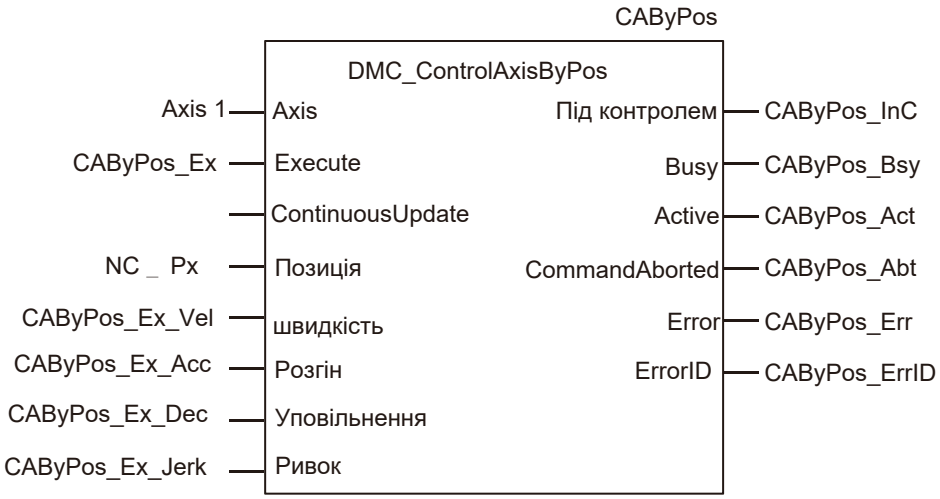
руху за один цикл SYNC. Положення осі становить 1500 (збільшено на 1000) після завершення руху.

Приклад програмування

Приклад програмування такий, коли використовується одна інструкція

Ім'я змінної	Тип даних	Початкове значення
Pwr	MC_Power	
Axis1	USINT	1
Pwr_Sta	BOOL	FALSE
Pwr_Bsy	BOOL	FALSE
Pwr_Act	BOOL	FALSE
Pwr_Err	BOOL	FALSE
Pwr_ErrID	WORD	0
CAByPos	DMC_ControlAxisByPos	
CAByPos_Ex	BOOL	FALSE
CAByPos_Ex_Pos	LREAL	
CAByPos_Ex_Vel	LREAL	1
CAByPos_Ex_Acc	LREAL	1
CAByPos_Ex_Dec	LREAL	1
CAByPos_Ex_Jerk	LREAL	1
CAByPos_InC	BOOL	FALSE
CAByPos_Bsy	BOOL	FALSE
CAByPos_Act	BOOL	FALSE
CAByPos_Abt	BOOL	FALSE
CAByPos_Err	BOOL	FALSE
CAByPos_ErrID	WORD	0

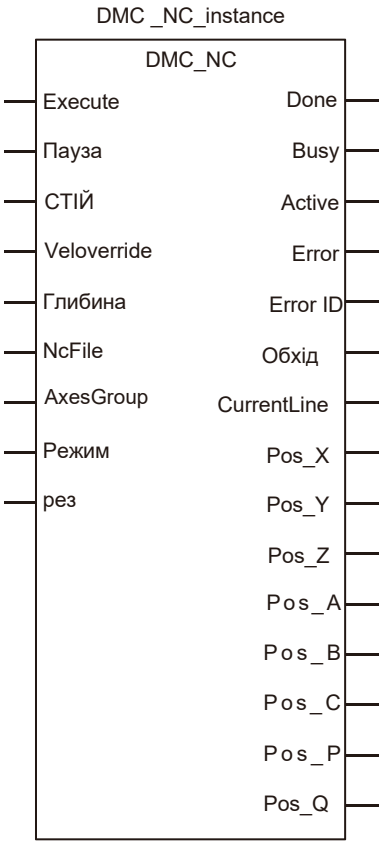




CAByPos_Ex змінюється на TRUE після ввімкнення живлення осі. Тим часом CAByPos_Bsy є TRUE у тому самому циклі. CAByPos_Act має значення TRUE у наступному циклі. Потім збільште значення CAByPos_Ex_Pos на 10 за цикл. Припустимо, що цикл SYNC становить 2 мс, є 500 циклів SYNC протягом 1 секунди, а швидкість командної осі становить 10*500=5000 одиниць/секунду.

11.8.2 DMC_NC

FB/FC	Пояснення	Застосовна модель
FB	Інструкція DMC_NC використовується для аналізу позиції кожної осі у файлі CNC кожного циклу SYNC.	AS516E-B AS532EST-B AS564EST-B



- **Вхідні параметри**

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
Execute	Інструкція виконується, коли *Execute* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	-
Пауза	Розбір файлу CNC за допомогою DMC_NC тимчасово припиняється, коли *пауза* змінюється з FALSE на TRUE.	BOOL -	Правда чи брехня (FALSE) -	-
СТІЙ	Розбір файлу CNC за допомогою DMC_NC припиняється, коли *зупинка* змінюється з FALSE на TRUE.	BOOL	Правда чи брехня (FALSE)	
VelOverride	Перевизначення швидкості , одиниця: % , наприклад, «100» означає 100%	LREAL	Негативний, 0-500 (0)	Коли *Execute* правда
Глибина	Введіть тут 1; Зарезервований.	LREAL	1-50 (0)	Коли *Execute* змінюється з FALSE на TRUE

Назва параметра	функція	Тип даних	Дійсний діапазон (за замовчуванням)	Час перевірки
NCFile	Номер файлу ЧПУ	LREAL	1-64 (0)	Коли *Execute* змінюється з FALSE на TRUE
AxesGroup	Кількість групи осей	LREAL	1-8 (0)	Коли *Execute* змінюється з FALSE на TRUE
Режим	Введіть тут 0; Зарезервований.			
рез	Зарезервований			

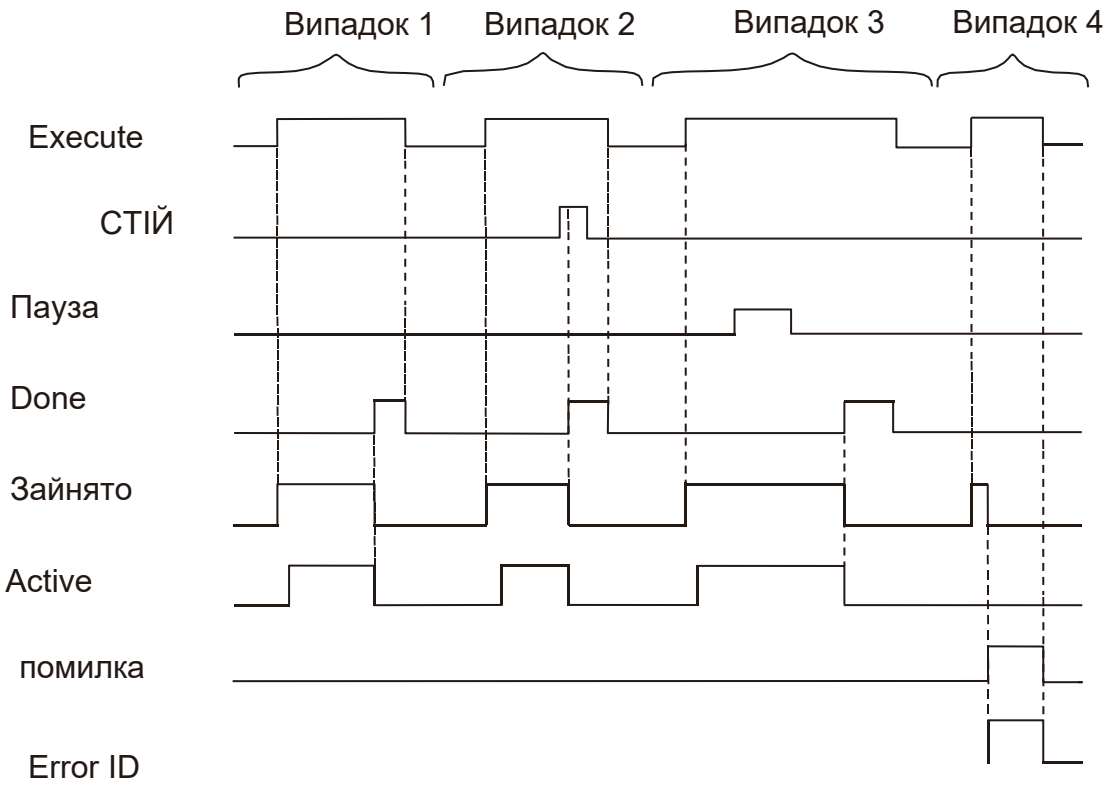
- **Вихідні параметри**

Назва параметра	функція	Тип даних	Допустимий діапазон
Done	TRUE, коли аналіз файлу ЧПУ завершено.	BOOL	TRUE/FALSE
Busy	TRUE, коли інструкція виконується.	BOOL	TRUE/FALSE
Active	TRUE, коли інструкція розбирає файл CNC.	BOOL	TRUE/FALSE
Error	TRUE, коли виникає помилка під час виконання інструкції.	BOOL	TRUE/FALSE
ErrorID	Містить код помилки, коли виникає помилка. Будь ласка, зверніться до розділу 12.2.	WORD	
Обхід	Зарезервований	BOOL	
CurrentLine	Номер рядка, де в даний момент виконується код G.	UDINT	
Pos_X	Положення осі X, яке аналізується	LREAL	
Pos_Y	Положення осі Y, яке аналізується	LREAL	
Pos_Z	Положення осі Z, яке аналізується	LREAL	
Pos_A	Положення осі A, яке аналізується	LREAL	
Pos_B	Положення Baxis, яке аналізується	LREAL	
Pos_C	Положення осі C, яке аналізується	LREAL	
Pos_П	Положення осі P, яке аналізується	LREAL	
Pos_Q	Положення осі Q, яке аналізується	LREAL	

- **Час оновлення виводу**

Назва параметра	Час зміни на TRUE	Час зміни на FALSE
Done	◆ Коли розбір файлу ЧПУ закінчено. ◆ Вхід *Stop* змінюється на TRUE.	◆ Коли *Execute* змінюється з TRUE на FALSE.
Busy	◆ Коли *Execute* змінюється на TRUE.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *«Готово»* змінюється на TRUE.
Active	◆ Коли інструкція починає розбирати файл CNC.	◆ Коли *помилка* змінюється на TRUE. ◆ Коли *«Готово»* змінюється на TRUE.
Error	◆ Коли під час виконання інструкції виникає помилка або вхідні параметри для інструкції є недопустимими.	◆ Коли *Execute* змінюється з TRUE на FALSE.

- **Діаграма часу оновлення вихідних даних**



Випадок 1 : *Busy* змінюється на TRUE, коли *Execute* змінюється на TRUE. Кілька циклів пізніше *Active* змінюється на TRUE. Після завершення синтаксичного аналізу файлу ЧПУ *значення Done* змінюється на TRUE. Тим часом *«Зайнятий»* і *«Активний»* змінюються на FALSE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FALSE.

Випадок 2 : *Busy* змінюється на TRUE, коли *Execute* змінюється на TRUE. Кілька циклів пізніше *Active* змінюється на TRUE. Змініть стан *Stop* на TRUE під час синтаксичного аналізу файлу CNC. У наступному циклі *Done* змінюється на TRUE, тим часом *Busy* та *Active* змінюються на FALSE, розбір файлу ЧПУ завершується, а виведення *CurrentLine* і положення осі припиняється. Змінюючи стан *Stop* на FALSE, *Done* залишається TRUE. Коли *Execute* змінюється на FALSE, *Done* змінюється на FLASE.

Випадок 3 : *Busy* змінюється на TRUE, коли *Execute* змінюється на TRUE. Кілька циклів пізніше *Active* змінюється на TRUE. змініть стан *Pause* на TRUE під час аналізу файлу CNC. Інструкція тимчасово припиняє синтаксичний аналіз файлу ЧПУ, а вихід *поточної лінії* та положення осі припиняється. Якщо змінити стан *Pause* на FALSE, синтаксичний аналіз файлу CNC продовжується, а вихідні дані *поточної лінії* та положення осі відновлюються. Після завершення *аналізу Done* змінюється на TRUE. Тим часом *«Зайнятий»* і *«Активний»* змінюються на FALSE. *Внесено* зміни до FALSE, змінивши *Execute* на FALSE.

Випадок 4 : коли *Execute* змінюється на TRUE і одне зі значень параметра для інструкції є неправильним, *Busy* змінюється на TRUE. Через один цикл *Error* змінюється на TRUE, *ErrorID* виводить ідентифікатор помилки, а *Busy* змінюється на FALSE. Коли *Execute* змінюється на FALSE, *Error* змінюється на FALSE, а значення *ErrorID* змінюється на 0.

- **функція**

Інструкція DMC_NC використовується для аналізу файлу CNC.
Інструкція аналізує положення кожної осі та виводить положення кожної осі у відповідні вихідні параметри інструкції (наприклад, Pos_X, Pos_Y тощо) кожного циклу SYNC. Насправді значення Pos_X і Pos_Y не керують осями, і це лише позиції, які аналізуються. Якщо ви хочете керувати будь-якою віссю за допомогою цієї інструкції, позицію, яку аналізує інструкція, можна призначити *позиції* інструкції DMC_ControlAxisByPos за допомогою інструкції DMC_NC разом із інструкцією DMC_ControlAxisByPos. Таким чином можна керувати рухом осі відповідно до шляху, запланованого у файлі ЧПУ.

1. Інструкція DMC_NC виконується шляхом зміни *Execute* з FALSE на TRUE. Немає жодного впливу на виконання інструкції шляхом зміни *параметра Execute* інструкції з TRUE на FALSE під час виконання інструкції.
2. зміна *Execute* з FALSE на TRUE під час виконання інструкції DMC_NC не впливає на виконання інструкції.
3. Встановлюючи *Stop* на TRUE під час виконання інструкції DMC_NC, розбір файлу CNC завершиться і *Зроблено* зміни на TRUE.
4. Встановлюючи для *паузи значення* TRUE під час виконання інструкції DMC_NC, синтаксичний аналіз файлу ЧПУ зупиниться на деякий час і продовжиться після того, як *для паузи* встановлено значення FALSE.
5. Вхід *NCFile* визначає номер файлу ЧПУ, який потрібно виконати , тобто . це ідентифікатор файлу ЧПУ, створеного в програмному забезпеченні для програмування.
6. Вхід *AxesGroup* визначає номер групи осей, яка має виконати файл ЧПУ.

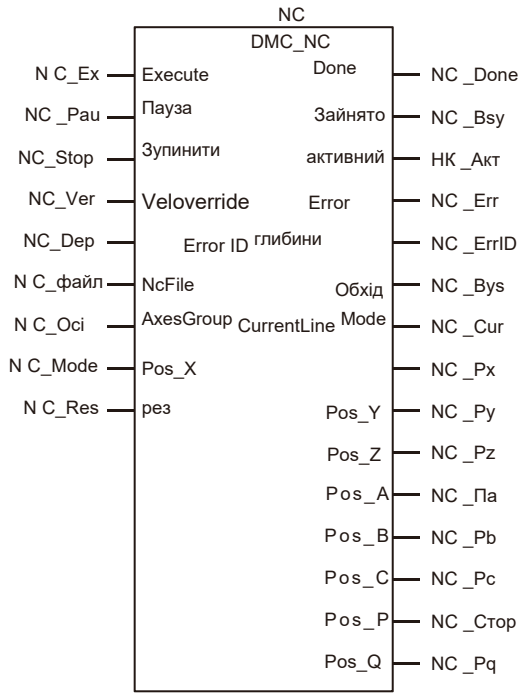
Приклад програмування

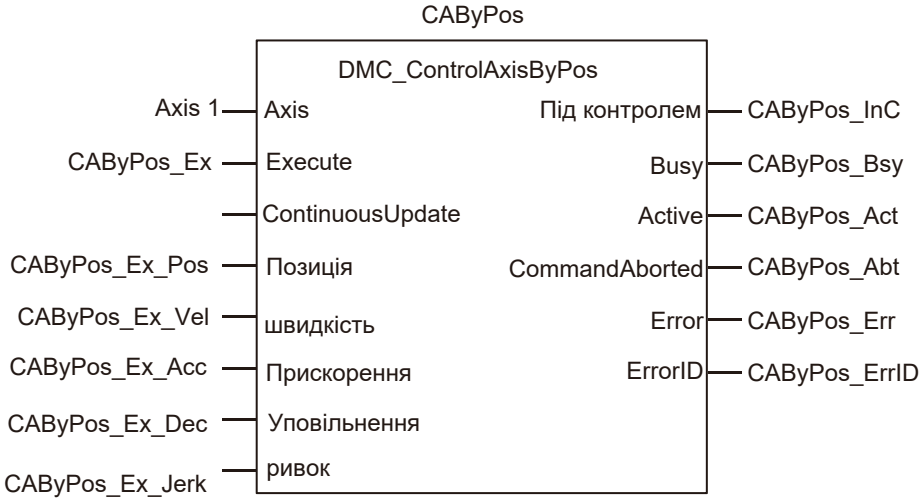
Осі керуються для переміщення відповідно до шляху, запланованого у файлі ЧПУ, за допомогою інструкцій DMC_NC і DMC_ControlBypos. Axis 1 і вісь 2 налаштовуються в програмі. Початкові положення обох осей становлять 10000 одиниць. Axis 1 – це вісь X, а вісь 2 – вісь Y.
Новий файл ЧПУ створюється в програмі з ідентифікатором файлу,
встановленим на 1. Перегляньте програму наступним чином.

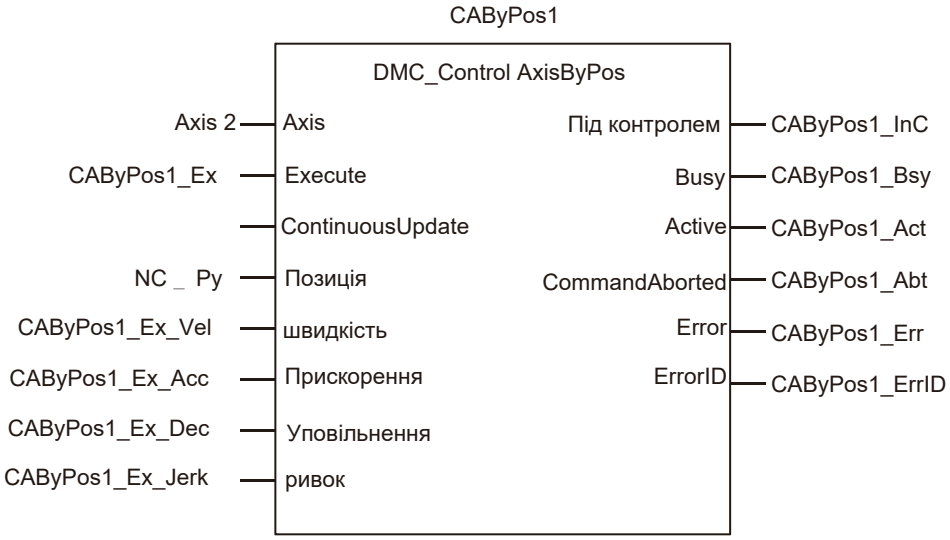
```
N00 G91
N01 G0 X40000 Y90000
```

Результат виконання файлу ЧПУ:
Після виконання G91 кінцева позиція кожної осі в G-кодах нижче G91 обчислюється за додатковим значенням, починаючи з поточної позиції. Після виконання G0 вісь X переміщається на 40 000 одиниць із поточної позиції (10000) до кінцевої позиції (50 000). Вісь Y переміщує 90000 одиниць з поточної позиції (10000) до кінцевої позиції (100000)

Ім'я змінної	Тип даних	Початкове значення
CAByPos	DMC_ControlAxisByPos	
Axis1	USINT	1
CAByPos_Ex	BOOL	FALSE
CAByPos_Ex_Vel	LREAL	1
CAByPos_Ex_Acc	LREAL	1
CAByPos_Ex_Dec	LREAL	1
CAByPos_Ex_Jerk	LREAL	1
CAByPos_InC	BOOL	FALSE
CAByPos_Bsy	BOOL	FALSE
CAByPos_Act	BOOL	FALSE
CAByPos_Abt	BOOL	FALSE
CAByPos_Err	BOOL	FALSE
CAByPos_ErrID	WORD	0
CAByPos1	DMC_ControlAxisByPos	
Axis2	USINT	2
CAByPos1_Ex	BOOL	FALSE
CAByPos1_Ex_Vel	LREAL	1
CAByPos1_Ex_Acc	LREAL	1
CAByPos1_Ex_Dec	LREAL	1
CAByPos1_Ex_Jerk	LREAL	1
CAByPos1_InC	BOOL	FALSE
CAByPos1_Bsy	BOOL	FALSE
CAByPos1_Act	BOOL	FALSE
CAByPos1_Abt	BOOL	FALSE
CAByPos1_Err	BOOL	FALSE
CAByPos1_ErrID	WORD	0
NC	DMC_NC	
NC_EX	BOOL	FALSE
NC_Pau	BOOL	FALSE
NC_Stop	BOOL	FALSE
NC_Ver	LREAL	100
NC_Dep	UINT	1
NC_файл	UINT	1
Oci NC	UINT	1
NC_Mode	INT	0
NC_Res	LREAL	0
NC_Act	BOOL	FALSE
NC_Err	BOOL	FALSE
NC_ErrID	WORD	0
NC_Bys	BOOL	0
NC_Cur	UDINT	0
NC_Px	LREAL	0
NC_Py	LREAL	0
NC_Pz	LREAL	0
NC_Pa	LREAL	0
NC_Pb	LREAL	0
NC_Pc	LREAL	0
NC_Pp	LREAL	0
NC_Pq	LREAL	0


NC
DMC_NC
N C_Ex
Execute
Done
NC _Done
NC _Pau
Пауза
Зайнято
NC _Bsy
NC_Stop
Зупинити
активний
НК _Акт
NC_Ver
Veloverride
Error
NC _Err
NC_Dep
Error ID
глибини
NC _ErrID
N C_файл
NcFile
Обхід
NC _Bys
N C_Oci
AxesGroup
CurrentLine
Mode
NC _Cur
N C_Mode
Pos_X
NC _Px
N C_Res
рез
Pos_Y
NC _Py
Pos_Z
NC _Pz
Pos_A
NC _Па
Pos_B
NC _Pb
Pos_C
NC _Pc
Pos_P
NC _Стор
Pos_Q
NC _Pq


CAByPos
DMC_ControlAxisByPos
Axis 1
Axis
Під контролем
CAByPos_InC
CAByPos_Ex
Execute
Busy
CAByPos_Bsy
ContinuousUpdate
Active
CAByPos_Act
CAByPos_Ex_Pos
Позиція
CommandAborted
CAByPos_Abt
CAByPos_Ex_Vel
швидкість
Error
CAByPos_Err
CAByPos_Ex_Acc
Прискорення
ErrorID
CAByPos_ErrID
CAByPos_Ex_Dec
Уповільнення
CAByPos_Ex_Jerk
ривок

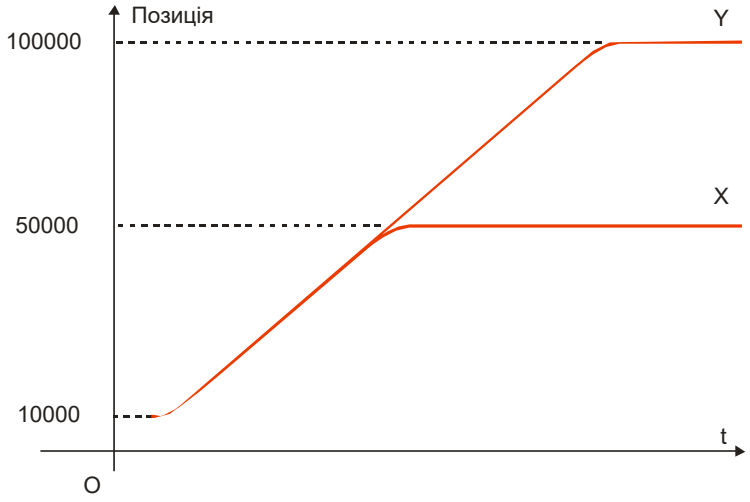

CAByPos1
DMC_Control AxisByPos
Axis 2
Axis
Під контролем
CAByPos1_InC
CAByPos1_Ex
Execute
Busy
CAByPos1_Bsy
ContinuousUpdate
Active
CAByPos1_Act
NC _ Py
Позиція
CommandAborted
CAByPos1_Abt
CAByPos1_Ex_Vel
швидкість
Error
CAByPos1_Err
CAByPos1_Ex_Acc
Прискорення
ErrorID
CAByPos1_ErrID
CAByPos1_Ex_Dec
Уповільнення
CAByPos1_Ex_Jerk
ривок

Коли CAByPos_Ex і CAByPos1_Ex встановлено на TRUE після ввімкнення живлення осі 1 і осі 2, *Busy* і *Активність* інструкцій CAByPos і CAByPos1 змінюється на TRUE.

Після того, як змінна NC_Ex змінюється на TRUE, NC_Bsy змінюється на TRUE, NC_Act змінюється на TRUE, NC_Cur відображає номер рядка, у якому наразі аналізується файл CNC. NC_Px, NC_Py, NC_Pz, NC_Pa, NC_Pb, NC_Pc, NC_Pp і NC_Pq виводять положення осей відповідно після аналізу файлу ЧПУ у поточному циклі.

CAByPos контролює переміщення осі X відповідно до значення NC_Px, яке аналізується, а CAByPos1 керує віссю Y відповідно до значення NC_Py, яке аналізується.

- Після виконання програми крива «Позиція/Час» для всього процесу руху показана нижче .



Пам'ятка

12

Розділ 12 Усунення несправностей

Зміст

12.1 Explanation of LED Indicators 12-2
12.2 Table of Error IDs in Motion Instructions 12-7
12.3 System Trouble Diagnosis through System Error Codes 12-19

У цьому розділі представлено в основному методи усунення несправностей у разі виникнення будь-якої несправності в контролері.

12.1 Пояснення світлодіодних індикаторів

- **Індикатор живлення**

Індикатор POWER вказує на стан живлення контролера руху.

стан світлодіода	Пояснення	Як боротися з
Блакитне світло увімкнено	Живлення в нормі	Без виправлення
Світлодіод вимкнений або миготливий	Ненормальне живлення	Перевірте, чи нормальне живлення для контролера руху.

- **Індикатор RUN**

Світлодіод RUN показує стан виконання програми в контролері руху.

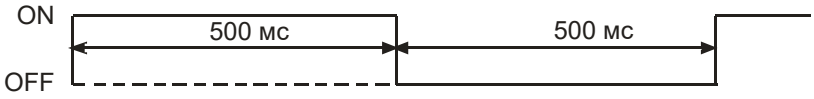
стан світлодіода	Пояснення	Як боротися з
Горить зелене світло	Контролер руху знаходиться в стані RUN.	Без виправлення
Світлодіод вимкнено	Контролер руху знаходиться в стані STOP.	Переведіть ПЛК у стан RUN відповідно до вимог

- **Індикатор ПОМИЛКА**

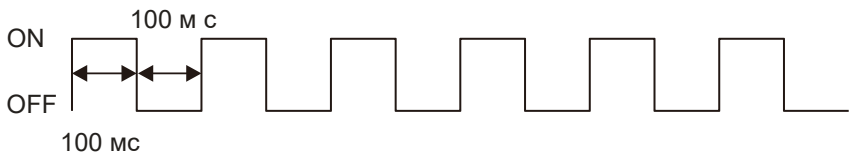
Світлодіод ERR вказує на стан помилки контролера руху.

стан світлодіода	Пояснення	Як боротися з
Світлодіод вимкнено	Контролер руху знаходиться в стані нормальної роботи.	Без виправлення
Блимає червоне світло	Помилки в програмі або конфігурації.	Отримайте детальну інформацію про помилку за допомогою функції діагностики помилок.
Горить червоне світло	Помилки в апаратному забезпеченні	Зверніться до місцевих техніків.

■ **Світлодіод ERROR: блимає червоне світло.（ 1 Гц ）**



■ **Індикатор ПОМИЛКИ: червоне світло швидко блимає.（ 10Гц ）**



- **Індикатор BAT LOW**

Світлодіодний індикатор BAT LOW вказує на стан кнопкової батареї контролера руху 3 В

стан світлодіода	Пояснення	Як боротися з
OFF	Кнопкова батарея 3 В подає живлення нормально.	Без виправлення
Горить червоне світло	Кнопкова батарея 3В не встановлена або перебуває під напругою.	Замініть кнопковий акумулятор на новий.

- **SD LED**

SD LED використовується для відображення стану SD-карти в контролері руху.

стан світлодіода	Пояснення	Як боротися з
OFF	1. SD-карта не вставлена в контролер руху. 2. Під час читання та запису документа виникає помилка	Вставте SD-карту чи ні відповідно до фактичної потреби
Швидко блимає зелене світло	SD-карта в контролері руху обмінюється даними	Без виправлення
Горить зелене світло	Немає обміну даними для SD-карти в контролері руху.	

- **Світлодіод EtherCAT**

Світлодіод EtherCAT відображає стан порту EtherCAT контролера.

СВІТЛОДІОДНИЙ	Держава	Індикація
Зелене світло	ON	Порт EtherCAT підключено до мережі EtherCAT.
	OFF	Порт EtherCAT ще не підключено до мережі EtherCAT.
Жовте світло	Блимає	Дані передаються або приймаються через порт EtherCAT
	OFF	Жодні дані не передаються та не приймаються через порт EtherCAT.

- **Світлодіод Ethernet**

Індикатор Ethernet вказує стан мережі контролера на порту Ethernet.

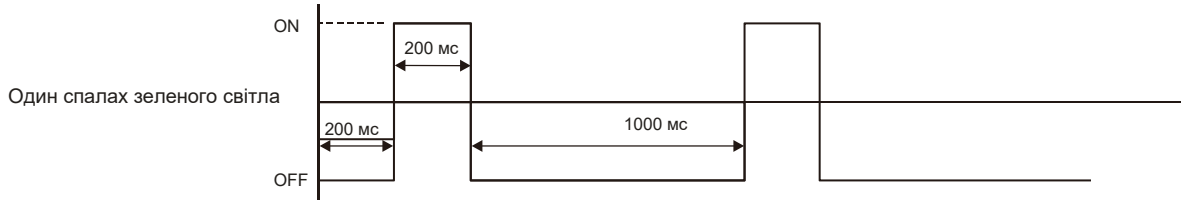
СВІТЛОДІОДНИЙ	Держава	Індикація
Зелене світло	ON	Швидкість передачі даних Ethernet становить 100 Мбіт/с.
	OFF	Швидкість передачі даних Ethernet становить 10 Мбіт/с, або контролер не підключено до мережі Ethernet.
Жовте світло	Блимає	Дані передаються або приймаються через порт Ethernet контролера руху.
	OFF	Жодні дані не передаються та не приймаються через порт Ethernet контролера руху.

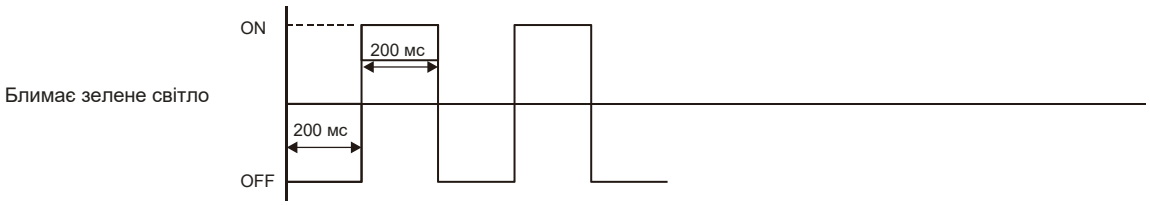
● CANopen LED

■ Індикатор RUN

стан світлодіода	Пояснення	Як боротися з
Один спалах зеленого світла	Комунікаційний порт CANopen знаходиться в стані STOP.	ПК завантажує дані конфігурації мережі. Зачекайте, поки завантаження завершиться.
Блимає зелене світло	Комунікаційний порт CANopen знаходиться в передопераційному стані.	1. Перевірте правильність підключення кабелю мережевої шини CANopen. 2. Перевірте, чи кабель шини CANopen є кабелем стандарту Delta CANopen. 3. Перевірте, чи обидва кінці шини CANopen під'єднано кінцевий резистор відповідно. 4. Перевірте, чи швидкість передачі даних головного пристрою є такою ж, як і інших підлеглих пристроїв. 5. Перевірте, чи налаштовані підлеглі підключені до мережі. 6. Перевірте, чи якийсь підлеглий не може встановити з'єднання з головним. 7. Перевірте, чи якийсь раб не в мережі.
Горить зелене світло	Комунікаційний порт CANopen перебуває в стані RUN.	Без виправлення

■ Світлодіод RUN: Зелене світло блимає один раз і блимає, як показано нижче.

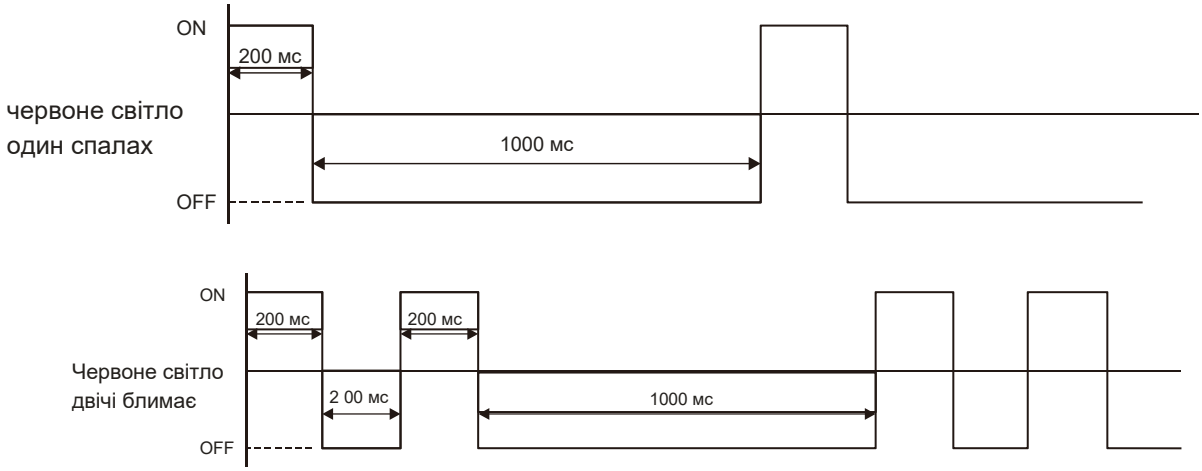




■ **Світлодіод ERR**

стан світлодіода	Пояснення	Як боротися з
Світлодіод вимкнено	Модуль PLC знаходиться в стані нормальної роботи.	Без виправлення
Червоне світло двічі блимає	Якийсь раб офлайн.	1. Перевірте, чи є кабель шини CANopen стандартним кабелем Delta. 2. Перевірте, чи обидва кінці шини CANopen під'єднано кінцевий резистор відповідно. 3. Перевірте, чи налаштовані підлеглі підключені до мережі. 4. Перевірте, чи не надто сильні перешкоди навколо кабелю шини CANopen.
Один спалах червоного світла	Помилка шини поза межами рівня сповіщення.	1. Перевірте правильність підключення кабелю шини CANopen. 2. Перевірте, чи кабель шини CANopen є Delta стандартний кабель. 3. Перевірте, чи кожен із двох кінців шини CANopen під'єднано кінцевий резистор відповідно. 4. Перевірте швидкість передачі даних порту CANopen так само, як і в інших рабів. 5. Перевірте, чи не надто сильні перешкоди навколо кабелю шини CANopen.
Горить червоне світло	Автобус викл	

■ **Світлодіод ERR: Червоне світло горить один раз і двічі спалахує, як показано нижче.**



- **Світлодіод RS232**

Світлодіод RS232, індикатор зв'язку RS-232 контролера руху вказує на стан зв'язку порту RS-232 контролера руху.

стан світлодіода	Індикація
Блимає жовте світло	Є відповідні дані через порт RS-232.
Світлодіод вимкнено	Немає відповідних даних через порт RS-232.

- **Світлодіод RS485**

  Світлодіод RS485, індикатор зв’язку RS-485 контролера руху вказує на стан зв’язку порту RS-485 контролера руху.

стан світлодіода	Індикація
Блимає жовте світло	Дані відповіді надходять через порт RS-485.
Світлодіод вимкнено	Немає відповідних даних через порт RS-485.

- **Світлодіод точки входу**

  Існує 16 світлодіодних індикаторів точок введення, які показують, увімкнено чи вимкнено цифрові вхідні точки контролера руху.

стан світлодіода	Індикація
Горить червоне світло	Точка введення включена.
Світлодіод вимкнено	Точка введення вимкнена.

- **Світлодіод вихідної точки**

  Існує 8 світлодіодних індикаторів вихідних точок, які показують, увімкнено чи вимкнено цифрові вихідні точки контролера руху.

стан світлодіода	Індикація
Горить червоне світло	Вихідна точка увімкнена.
Світлодіод вимкнено	Вихідна точка вимкнена.

12.2 Таблиця ідентифікаторів помилок в інструкціях щодо руху

Коли в інструкції руху виникає помилка, значення ErrorID можна побачити таким чином для аналізу причини та усунення несправностей.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
1001	4097	Номер осі перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Axis знаходиться в допустимому діапазоні.
1002	4098	Прискорення перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Acceleration є додатним числом.
1003	4099	Уповільнення перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Deceleration є додатним числом.
1004	4100	Швидкість зміни прискорення перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Jerk є додатним числом.
1005	4101	Швидкість перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Velocity є ненульовим значенням.
1006	4102	Значення позиції перевищує допустимий діапазон.	Переконайтеся, що значення вхідної змінної Position MC_MoveAbsolute не перевищує значення Modulo серед параметрів осі.
1007	4103	Значення напрямку перевищує допустимий діапазон.	Змініть значення вхідної змінної Direction на те, яке можна встановити в інструкції.
1008	4104	Значення буферного режиму перевищує допустимий діапазон.	Змініть значення вхідної змінної BufferMode на те, яке можна встановити в інструкції.
1009	4105	Введене значення для типу контрольної позиції неправильне.	Змініть значення вхідної змінної ReferenceType на те, яке можна встановити в інструкції.
100A	4106	Час для виконання MC_SetPosition є неправильним.	Змінити час виконання MC_SetPosition. Не виконуйте MC_SetPosition під час виконання MC_Home або MC_Stop.
100B	4107	Номер таблиці електронної камери неправильний.	Змініть значення вхідної змінної CamTable на CamId, встановлене в програмному забезпеченні.
100C	4108	Номер осі головної осі неправильний.	Переконайтеся, що значення вхідної змінної Axis знаходиться в допустимому діапазоні.
100D	4109	Введене значення режиму залучення неправильне.	Змініть значення вхідної змінної StartMode на те, яке можна встановити в інструкції.
100E	4110	Значення головного масштабування неправильне.	Переконайтеся, що значення вхідної змінної MasterScaling є додатним числом.
100F	4111	Значення масштабування підпорядкованого пристрою неправильне.	Переконайтеся, що значення вхідної змінної SlaveScaling є ненульовим значенням.
1010	4112	Вибране джерело положення головної осі неправильне.	Змініть значення вхідної змінної MasterValueSource на те, яке можна встановити для інструкції.
1011	4113	Конфлікт в осі № головної та веденої осей.	Змініть значення вхідних змінних, Master і Slave на різні значення.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
1012	4114	Неправильне значення чисельника e-gear	Змініть значення вхідної змінної Numerator на ненульове значення.
1013	4115	Неправильне значення знаменника e-gear	Змініть значення вхідної змінної, Знаменник, на ненульове значення.
1014	4116	Значення VelFactor неправильне.	Змініть значення вхідної змінної VelFactor на те, яке можна встановити в інструкції.
1015	4117	Час очікування SDO у мережі CANopen (або мережі EtherCAT)	Перевірте, чи існує підлеглий пристрій, вказаний в інструкції. Перевірте, чи нормальне з'єднання між підлеглим пристроєм, до якого здійснюється доступ, і портом CANopen або портом EtherCAT. Перевірте, чи однакові швидкості передачі даних порту CANopen або EtherCAT і підпорядкованого пристрою, до якого здійснюється доступ.
1016	4118	Помилка вхідного параметра інструкції SDO	Перевірте, чи прийнятні налаштування вхідних параметрів інструкції SDO. Наприклад, подивіться, чи існують індекс і піднідекс, до яких здійснюється доступ, і чи допустиме значення DataType.
1017	4119	Інші помилки в SDO в мережі CANopen (або мережі EtherCAT).	Перевірте, чи раби нормально працюють.
1018	4120	Значення TriggerInput інструкції захоплення позиції DMC_TouchProbe неправильне.	Змініть значення вхідної змінної TriggerInput. Значення можна встановити в діапазоні 0~15 відповідно, що представляє I0~I7 та I10~I15.
1019	4121	Точка введення, визначена TriggerInput інструкції DMC_TouchProbe, була використана в іншому DMC_TouchProbe.	Змініть значення TriggerInput інструкції на одне значення, яке ще не використовувалося.
101A	4122	Лише вікно DMC_TouchProbe є ненормальним.	Змініть значення Firststops і Laststops у межах допустимого діапазону.
101B	4123	Дві інструкції DMC_TouchProbe виконуються для захоплення положення однієї осі одночасно.	Запобігайте одночасному захопленню позиції однієї осі двома інструкціями DMC_TouchProbe.
101C	4124	Значення налаштування режиму DMC_TouchProbe неправильне.	Змініть значення вхідної змінної Mode на те, яке можна встановити в інструкції.
101D	4125	Вісь, визначена DMC_TouchProbe, не є віссю кодера.	Змініть значення вхідної змінної Axis на номер осі налаштованої осі кодера.
101E	4126	Значення ActivationPosition для MC_CamIn є неправильним.	Змініть значення вхідної змінної ActivationPosition на те, яке можна встановити в інструкції.
1020	4128	Використана вісь не налаштована на мережу EtherCAT у програмному забезпеченні або тип осі неправильний.	Змініть значення вхідної змінної Axis на номер осі, який було налаштовано в мережі EtherCAT.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
1021	4129	Неправильний радіус осі обертання.	Змініть значення вхідної змінної RotaryAxisRadius. Воно має бути більше 0.
1022	4130	Радіус осі подачі неправильний.	Змініть значення вхідної змінної FeedAxisRadius. Воно має бути більше 0.
1023	4131	Неправильна довжина різання.	Змініть значення вхідної змінної CutLenth APF_RotaryCut_Init. Воно має бути більше 0.
1024	4132	Значення SyncStartPos неправильне.	Змініть значення вхідної змінної SyncStartPos APF_RotaryCut_Init. Воно має бути між 0 і довжиною різання.
1025	4133	Значення SyncStopPos неправильне.	Змініть значення вхідної змінної SyncStopPos APF_RotaryCut_Init. Воно має бути між 0 і довжиною різання.
1026	4134	Параметри SyncStopPos і SyncStartPos неправильні.	Значення вхідної змінної SyncStopPos має бути меншим за значення SyncStartPos інструкції.
1027	4135	Значення RotCutID неправильне.	Значення вхідної змінної RotCutID має бути в діапазоні від 1 до 8.
1028	4136	Значення RotaryAxisKnifeNum неправильне.	Значення вхідної змінної RotaryAxisKnifeNum має бути в діапазоні 1~16.
1029	4137	Внутрішній стан обертального різання є недопустимим.	Змініть значення параметрів для ініціалізації ротаційного різання.
103A	4154	Помилка ініціалізації ротаційного різання.	Оскільки APF_RotaryCut_Init не було виконано, спочатку виконайте APF_RotaryCut_Init, а потім виконайте APF_RotaryCut_In.
103B	4155	Вісь не в мережі, і функцію захоплення не можна виконати	Виконайте інструкцію захоплення після нормального підключення осі.
103C	4156	Значення MasterOffset для MC_CamIn більше, ніж діапазон циклу кулачка головної осі.	Змініть значення MasterOffset на значення між від'ємним числом і додатним числом діапазону циклу кулачка головної осі. (Діапазон циклу кулачка головної осі = максимальний цикл кулачка головної осі - мінімальний цикл кулачка головної осі)
103D	4157	Значення SlaveOffset MC_CamIn більше, ніж діапазон циклу кулачка ведомої осі.	Змініть значення SlaveOffset на значення між від'ємним числом і додатним числом діапазону циклів кулачка веденої осі. (Діапазон циклу кулачка веденої осі = максимальний цикл кулачка веденої осі - мінімальний цикл кулачка веденої осі)
103E	4158	Значення глибини інструкції виходить за межі діапазону.	Змініть значення введеної глибини, щоб не перевищувати діапазон.
103F	4159	Діапазон значень VelOverride інструкції є неприпустимим.	Змініть значення вхідного параметра VelOverride, щоб не виходити за межі діапазону.
1040	4160	Код файлу є незаконним.	Змініть значення вхідного NCFile на належне значення коду.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
1041	4161	DMC_SetTorque виконується, коли вісь не перебуває в стані зупинки.	Переконайтеся, що DMC_SetTorque виконується, коли вісь перебуває в стані зупинки.
1042	4162	Виконання MC_Reset не вдається.	Перевірте, чи існує вісь, визначена MC_Reset. MC_Reset виконується після видалення аварійного сигналу сервоприводу.
1043	4163	Виконання інструкції призводить до того, що положення осі виходить за межі встановленого програмним забезпеченням діапазону.	Змініть інструкцію, щоб переконатися, що кінцева позиція не перевищує діапазон програмного обмеження.
1044	4164	Крива кулачка, визначена MC_CamIn, не вбудована в програмне забезпечення.	Перевірте, чи може значення CamTable MC_CamIn відповідати кулачковій кривій, вбудованій у програмне забезпечення.
1045	4165	Помилка ідентифікатора групи осей	Перевірте, чи значення GroupID знаходиться в діапазоні від 1 до 8.
1046	4166	Помилка введеного значення режиму	Значення Mode для інструкції може бути встановлено лише на 0
1047	4167	Номер інструкції «Від/До» неправильний.	Перевірте правильність значення станції та номера інструкції.
1048	4168	Помилка в кількості регістрів CR, які читаються та записуються функцією From/To.	Перевірте, чи значення Num знаходиться в діапазоні 1~64.
1049	4169	Змінна не встановлена для контакту введення інструкції	Встановіть змінну для вхідного виводу.
104A	4170	Немає відповіді на інструкцію Від/До	Перевірте, чи належне з'єднання між модулями та чи нормально працює модуль розширення.
104B	4171	Порожній файл ЧПК	Перевірте, чи правильне значення NCFile і чи відповідний файл CNC порожній.
104C	4172	Помилка вирішення шляху	Переконайтеся, що параметри G-кодів або групових інструкцій для шляху правильні.
104D	4173	Інструкція захоплення позиції не отримала сигнал захоплення в діапазоні вікна, і захоплення не вдалося.	Переконайтеся, що діапазон вікна встановлено належним чином.
104E	4174	Помилка ідентифікації коду G	Переконайтеся, що файл G-коду записаний належним чином.
104F	4175	Неправильний формат попередньо зчитаного коду G	Переконайтеся, що файл G-коду записаний належним чином.
1050	4176	Помилка попереднього читання G-коду	Переконайтеся, що файл G-коду записаний належним чином.
1051	4177	Помилка запису шляху	Переконайтеся, що шлях написаний правильно.
1052	4178	Помилка налаштування для позиції приводу осі, указаного в інструкції MC_Home. Можливо, дисковод не підтримує параметр.	Перевірте, чи підтримує привод параметр Position.
1053	4179	Інструкція MC_Home не підтримує вісь кодувальника, яка приймає джерела даних як режим кодування та вісь абсолютного кодувальника SSI.	Змініть тип осі на інший тип осі.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
1054	4180	Забагато рівнів вкладеності G26 у G-кодах	Перевірте, чи кількість рівнів вкладеності G26 перевищує 16.
1057	4183	Кількість осей, які виконують багатоосьові інструкції, перевищує макс. кількість осей для конкретної моделі.	Зберігайте кількість осей, які виконують багатоосьові інструкції, у діапазоні номерів осей для конкретної моделі; або замініть контролер на іншу модель.
1400	5120	Модуль не підтримує функцію.	Змініть значення вхідної змінної *Module* інструкції як кількість AS02PU/AS04PU, фактично підключених на правій стороні контролера AS500.
1401	5121	Деякі вхідні значення інструкції, пов'язаної з модулем PU, є недопустимими або поза діапазоном.	Змініть значення вхідних змінних інструкції та переконайтеся, що вони знаходяться в межах дозволеного діапазону.
1402	5122	Тайм-аут зв'язку виникає між головним і його правим модулем.	Переконайтеся, що з'єднання між контролером AS500 і його правим модулем AS02PU/AS04PU є правильним. Будь ласка, зверніться до місцевих техніків, якщо помилка не зникає після повторного ввімкнення.
1403	5123	Такого номера осі в модулі PU немає.	Змініть значення вхідної змінної *Axis* інструкції та переконайтеся, що воно знаходиться в межах від 1 до 4.
1404	5124	Вихідна швидкість осі модуля ПУ є недопустимою.	Змініть значення швидкості виведення в інструкції та переконайтеся, що воно знаходиться між - 100 000 і 100 000.
1405	5125	У той час як зазначена вісь керується інструкцією, пов'язаною з модулем PU, іншою інструкцією є керування тією ж віссю.	Дочекайтеся виконання інструкції, що виконується, а потім виконайте іншу інструкцію. Або змініть *Enable* інструкції, що виконується, на FALSE, щоб потім виконати іншу інструкцію.
2001	8193	Вісь вимикається за допомогою інструкції MC_Power, коли вона не перебуває в стані зупинки.	Вимкніть вісь за допомогою інструкції MC_Power, коли вісь перебуває в стані зупинки.
2002	8194	Інструкція не може бути виконана через обмеження напрямку руху.	Установіть для EnablePositive і EnableNegative MC_Power значення TRUE, щоб скасувати обмеження напрямку руху осі.
2004	8196	MC_HaltSuperimposed не можна виконати, якщо MC_MoveSuperimposed ще не виконано.	Змінити послідовність виконання MC_HaltSuperimposed. Виконання MC_HaltSuperimposed повинно проводитися в процесі виконання MC_MoveSuperimposed.
2100	8448	Кінцевий автомат обмежує, що функція не може бути виконана.	Змінити час виконання інструкції. Зверніться до кінцевого автомата в розділі 10.4 щодо виконання інструкцій руху.
2101	8449	Буферний регістр заповнений.	BufferMode інструкції керування рухом підтримує лише один перемикач для зміни часу виконання поточної інструкції та уникнення того, що інша інструкція також очікує на виконання (BufferMode не 0), поки один

ErrorID		Значення	Як боротися з
Hex	Десятковий		
			інструкція очікує на виконання (BufferMode не 0).
2102	8450	Функція буфера не може бути виконана в інструкції.	Інструкцією не можна керувати в режимі буфера.
3001	12289	Помилка в налаштуванні типу осі	Змініть тип осі у вікні конфігурації осі.
3002	12290	Серво сигналізація	Керуйте сервоприводом після очищення сигналізації сервоприводу.
3003	12291	Час очікування сервоприводу	Перевірте, чи в порядку з'єднання між контролером і сервоприводом.
3004	12292	Положення команди перевищує граничне положення, встановлене в програмному забезпеченні.	Перевірте правильність установленого програмного обмеження або вимкніть програмне обмеження.
3005	12293	Процес від RUN до STOP відбувається в контролері (під час виконання інструкції руху)	Очистіть помилку за допомогою інструкції MC_Reset, а потім виконайте іншу інструкцію руху.
600D	24589	Помилка з'єднання	Перевірте, чи правильний кабель зв'язку.
6200	25088	Помилка віддаленого IP TCP	Перевірте, чи правильний формат віддаленої IP-адреси TCP.
6201	25089	Помилка віддаленого порту TCP	Перевірте, чи налаштування віддаленого порту TCP виходить за допустимий діапазон.
6203	25091	Помилка адреси реєстру надсилання даних TCP	Перевірте, чи адреса реєстру надсилання даних знаходиться в допустимому діапазоні.
6206	25094	Помилка адреси реєстру отримання даних TCP	Перевірте, чи адреса реєстру, що отримує дані, знаходиться в допустимому діапазоні.
6208	25096	Дані, які насправді отримує майстер TCP, перевищують встановлену довжину.	Змінити вказану довжину отриманих даних.
6209	25097	Помилка віддаленої IP-адреси UDP	Перевірте, чи правильний формат віддаленої IP-адреси UDP.
620A	25098	Помилка порту зв'язку UDP	Змінити порт зв'язку UDP.
620C	25100	Помилка адреси реєстру надсилання UDP	Перевірте, чи адреса реєстру надсилання даних знаходиться в допустимому діапазоні.
620F	25103	Помилка адреси реєстру отримання UDP	Перевірте, чи адреса реєстру, що отримує дані, знаходиться в допустимому діапазоні.
6210	25104	Фактично отримані дані UDP перевищують встановлену довжину.	Змінити вказану довжину даних, які потрібно отримати.
6212	25106	Час очікування з'єднання Ethernet	Змініть час очікування або перевірте, чи віддалений пристрій підключено нормально.
6213	25107	Дані, які фактично отримує підлеглий пристрій TCP, перевищують встановлену довжину.	Змінити вказану довжину надісланих даних.
6214	25108	Посилання вимкнено через виняток підключення.	Переконайтеся, що віддалений пристрій працює.
6215	25109	Підключення не вдається або ще не ввімкнено.	Перевірте, чи правильна послідовність дій інструкції.
6220	25120	Час вийшов	Змініть час очікування або переконайтеся, що віддалений пристрій підключено нормально.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
6300	25344	Кількість підключень перевищує ліміт.	Перевірте, чи номер підключення знаходиться в межах дозволеного діапазону.
8000	32768	Інструкцію можна використовувати для діагностики лише тоді, коли контролер працює як CANopen master.	Змініть локальний контролер як головний CANopen перед використанням інструкції.
8800	34816	Номер пріоритету завдання більше 31.	Встановіть номер пріоритету завдання на значення менше 31.
8801	34817	Функцію сторожового таймера для завдання ще не ввімкнено.	Увімкніть функцію сторожового таймера перед виконанням інструкції.
8808	34824	Ведучий не налаштував ведений пристрій для діагностики.	Налаштуйте slave перед діагностикою.
8810	34832	Помилка типу діагностики	Змінити тип діагнозу
8818	34840	Вісь, яку потрібно діагностувати, не налаштована.	Налаштуйте вісь перед діагностикою.
8820	34848	Помилка типу діагностики	Змінити тип діагнозу
9000	36864	Кількість каналів Ethernet перевищує діапазон 1~16.	Змініть номер посилання Ethernet на 1~16
9001	36865	Довжина записаних даних, налаштована для з'єднання Ethernet, перевищує максимальну.	Змініть довжину записаних даних, налаштовану для каналу Ethernet, у межах дозволеного діапазону.
9002	36866	Довжина даних для читання, налаштована для з'єднання Ethernet, перевищує максимальну.	Змініть довжину даних для читання, налаштовану для каналу Ethernet, у межах дозволеного діапазону.
9003	36867	Помилка фізичного підключення Ethernet	Перевірте, чи з'єднання мережевого обладнання в нормі, наприклад, з'єднання мережевого кабелю.
9004	36868	Номер розетки перевищує допустимий діапазон	Змініть номер сокета на 1~4.
9005	36869	Довжина надісланих даних, налаштована для функції Socket, перевищує допустиме максимальне значення.	Змініть довжину надісланих даних, налаштовану для функції Socket, як значення в межах 0–200.
9006	36870	Довжина отриманих даних, налаштована для функції Socket, перевищує допустиме максимальне значення.	Змініть довжину отриманих даних, налаштовану для функції Socket як 0~200.
9007	36871	Налаштування тайм-ауту зв'язку в конфігурації каналу Ethernet неправильне.	Змініть час очікування на значення більше 0.
9008	36872	Довжина надісланих і отриманих даних, налаштованих для функції Socket, дорівнює 0.	Змініть довжину надісланих і отриманих даних, налаштованих для функції Socket, як значення, відмінне від 0.
9010	36880	Номер посилання ПЛК RS485 перевищує діапазон 1-24.	Змініть номер посилання ПЛК як значення в діапазоні 1-24.
9011	36881	Довжина записаних даних, налаштована для зв'язку ПЛК RS485, перевищує максимально допустиме значення.	Змініть довжину записаних даних, налаштовану для зв'язку ПЛК RS485, як значення, яке знаходиться в межах дозволеного діапазону.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
9012	36882	Довжина зчитування даних, налаштована для зв'язку ПЛК RS485, перевищує максимально допустиме значення.	Змініть довжину даних читання, налаштовану для зв'язку ПЛК RS485, як значення в межах дозволеного діапазону.
9013	36883	Довжина надісланих даних, налаштована для функції вільного протоколу RS485, перевищує максимально допустиме значення.	Змініть довжину надісланих даних, налаштовану для функції вільного протоколу RS485, як значення в межах дозволеного діапазону.
9014	36884	Довжина отриманих даних, налаштована для функції вільного протоколу RS485, перевищує максимально допустиме значення.	Змініть довжину отриманих даних, налаштовану для функції вільного протоколу RS485, як значення в межах дозволеного діапазону.
9015	36885	Номер RS232 PLC Link перевищує діапазон 1~24.	Змініть номер RS232 PLC Link як значення в діапазоні 1-24.
9016	36886	Довжина записаних даних, налаштована для зв'язку ПЛК RS232, перевищує максимально допустиме значення.	Змініть довжину записаних даних, налаштовану для зв'язку ПЛК RS232, як значення в межах дозволеного діапазону.
9017	36887	Довжина даних для читання, налаштована для зв'язку ПЛК RS232, перевищує максимально допустиме значення.	Змініть довжину даних для читання, налаштовану для зв'язку ПЛК RS232, як значення в межах дозволеного діапазону.
9018	36888	Довжина надісланих даних, налаштована для функції вільного протоколу RS232, перевищує максимально допустиме значення.	Змініть довжину надісланих даних, налаштовану для функції вільного протоколу RS232, як значення в межах дозволеного діапазону.
9019	36889	Довжина отриманих даних, налаштована для функції вільного протоколу RS232, перевищує максимально допустиме значення.	Змініть довжину отриманих даних, налаштовану для функції вільного протоколу RS232, як значення в межах дозволеного діапазону.
901A	36890	Налаштування часу очікування зв'язку в конфігурації зв'язку ПЛК RS485/RS232 неправильне.	Змініть час очікування на значення більше 0.
901B	36891	Довжина прочитаних і записаних даних, налаштованих для каналу Ethernet/RS485/RS232, дорівнює 0.	Змініть будь-яку довжину прочитаних даних і записаних даних, налаштованих як значення, яке не дорівнює 0.
901C	36892	Довжина надісланих і отриманих даних, налаштованих для функції вільного протоколу RS485/RS232, дорівнює 0.	Змініть будь-яку довжину надісланих і отриманих даних, налаштованих як значення, яке не дорівнює 0.
9020	36896	Локальний буфер типу word для запису даних не має достатньо місця, щоб відповідати вказаній довжині даних. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера, щоб у ньому було достатньо місця для відповідності вказаній довжині даних.
9021	36897	Початкова адреса локального буфера типу слова для запису даних виходить за межі дозволеної області реєстру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера в межах дозволеної області реєстру слів.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
9022	36898	Початкова адреса локального буфера типу слова для запису даних знаходиться в межах дозволеної області реєстру слів, але не може відповідати вирівнюванню адрес реєстру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера або довжину зміщення.
9023	36899	У локальному буфері типу word для читання даних недостатньо місця, щоб відповідати заданій довжині даних. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера, щоб у ньому було достатньо місця для відповідності вказаній довжині даних.
9024	36900	Початкова адреса локального буфера типу слова для запису даних виходить за межі дозволеної області реєстру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера в межах дозволеної області реєстру слів.
9025	36901	Початкова адреса локального буфера типу слів для читання даних знаходиться в межах дозволеної області реєстру слів, але не може відповідати вирівнюванню адрес реєстру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера або довжину зміщення.
9026	36902	Локальний буфер бітового типу для запису даних не має достатньо місця, щоб відповідати вказаній довжині даних. (Діапазон: %QX0.0~%QX127.7,%MX0.0~%MX65535.7)	Змініть початкову адресу локального буфера, щоб у ньому було достатньо місця для відповідності вказаній довжині даних.
9027	36903	Початкова адреса локального буфера бітового типу для запису даних виходить за межі дозволеної області. （ Діапазон: %QX0.0~%QX127.7,%MX0.0~%MX65535.7 ）	Змініть початкову адресу локального буфера в межах дозволеної області.
9028	36904	У локальному буфері бітового типу для читання даних недостатньо місця, щоб задовольнити вказану довжину даних. (%QX0.0~%QX127.7,%MX0.0~% MX65535.7)	Змініть початкову адресу локального буфера, щоб у ньому було достатньо місця для відповідності вказаній довжині даних.
9029	36905	Початкова адреса локального буфера бітового типу для читання даних виходить за межі дозволеної області. (%QX0.0~%QX127.7,%MX0.0~% MX65535.7)	Змініть початкову адресу локального буфера в межах дозволеної області.
9030	36912	Помилка типу об'єкта	Змініть значення вхідного параметра ObjType.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
9031	36913	Зазначений код функції для читання даних перевищує дозволений діапазон.	Укажіть новий код функції, який знаходиться в межах дозволеного діапазону.
9040	36928	У локальному буфері для надсилання даних недостатньо місця, щоб відповідати вказаній довжині даних. (%MW0~%MW32767)	Змініть початкову адресу локального буфера або вказану довжину даних.
9042	36930	Початкова адреса локального буфера для надсилання даних виходить за межі дозволеної області реєстру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера.
9043	36931	У локальному буфері, визначеному для отримання даних, недостатньо місця, щоб відповідати вказаній довжині даних. (%MW0~%MW32767)	Змініть початкову адресу локального буфера або вказану довжину даних.
9045	36933	Початкова адреса локального буфера, указана для отримання даних, знаходиться поза дозволеною областю регістру слів. (Дійсно діапазон: %MW0~%MW32767)	Змініть початкову адресу локального буфера.
9100	37120	Значення параметра інструкції сокета перевищує допустимий діапазон.	Змініть значення параметра інструкції Socket у межах дозволеного діапазону.
9101	37121	Фізичне підключення Ethernet розірвано.	Перевірте правильність підключення мережевого кабелю.
9102	37122	Помилка віддаленої IP-адреси TCP	Змініть налаштування для віддаленої IP-адреси.
9103	37123	Помилка порту TCP	Змініть налаштування віддаленого порту.
9105	37125	Виникає помилка в адресах реєстру для надсилання даних TCP.	Змінити адресу реєстру для надсилання даних TCP.
9106	37126	Триває отримання даних TCP/UDP.	Останнє отримання даних не було завершено, тому нове отримання не може бути запущено.
9107	37127	Помилка адреси регістру отримання TCP	Змініть адресу реєстру для отримання даних TCP.
9108	37128	Довжина отриманих даних перевищує встановлену довжину в режимі TCP-сервера.	Встановіть довжину отриманих даних на довжину, що перевищує або дорівнює кількості байтів перших отриманих даних.
9109	37129	Довжина отриманих даних перевищує встановлену довжину при передачі UDP.	Встановіть довжину отриманих даних на довжину, яка перевищує або дорівнює кількості байтів першої частини отриманих даних.
910A	37130	Помилка віддаленої IP-адреси UDP	Змініть налаштування віддаленої IP-адреси.
910B	37131	Помилка порту UDP	Локальний порт і віддалений порт не можуть мати значення 0 одночасно.
910C	37132	Сталася помилка в адресах реєстру для надісланих даних UDP.	Змінити адреси реєстру для надісланих даних.
910D	37133	Сталася помилка в реєстрі адрес для отриманих даних UDP.	Змінити адреси реєстру для зберігання отриманих даних.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
910E	37134	Тайм-аут з'єднання TCP	Перевірте, чи правильна конфігурація роз'єму, чи віддалений пристрій працює нормально.
910F	37135	Довжина отриманих даних перевищує встановлену довжину в режимі клієнта TCP.	Встановіть довжину отриманих даних на довжину, яка перевищує або дорівнює кількості байтів першої частини отриманих даних.
9110	37136	TCP-з'єднання відхилено віддаленим пристроєм.	Перевірте, чи віддалений пристрій працює нормально, або повторіть спробу підключення до віддаленого пристрою.
9111	37137	Посилання TCP/UDP не ввімкнено.	Переконайтеся, що посилання активовано.
9112	37138	З'єднання TCP/UDP було запущено.	Посилання створюється, тому його не можна повторно запустити.
9113	37139	Надсилання даних TCP/UDP було запущено.	Останнє надсилання даних не було завершено, тому надсилання не можна запустити повторно.
9114	37140	З'єднання TCP/UDP створено.	Посилання було створено, тому повторний тригер не може створити посилання.
9115	37141	Посилання TCP/UDP вимкнено.	Посилання вимикається, тому вимкнення посилання не може бути повторно активовано.
9116	37142	Посилання TCP/UDP не вимкнено.	Конфігурацію параметрів Socket можна проводити тільки при вимкненому посиланні.
9117	37143	Значення параметра для довжини надісланих даних TCP/UDP перевищує обмеження.	Змініть довжину надісланих даних у межах дозволеного діапазону.
9118	37144	Значення параметра для довжини отриманих даних TCP/UDP перевищує обмеження.	Змініть довжину отриманих даних у межах дозволеного діапазону.
A000	40960	Помилка номера точки камери	Змініть номер таблиці кулачків.
A001	40961	Помилка номера точки штовхання	Змінити номер точки штовхача.
A008	40968	Помилка номера коду M	Змінити номер коду M.
A010	40976	Помилка кінцевого автомата групи Axes	Змінити час виконання інструкції.
A011	40977	Номер групи осей перевищує допустимий діапазон.	Змініть номер групи осей.
A012	40978	Значення вхідного параметра TransitionMode неправильне.	Змінити введене значення.
A013	40979	Значення вхідного параметра TransitionParameter неправильне.	Змінити введене значення.
A014	40980	Значення параметра BufferMode не відповідає значенню TransitionMode.	Змінити введене значення.
A015	40981	Значення CircMode неправильне.	Змінити введене значення.
A016	40982	Значення PathChoice неправильне.	Змінити введене значення.
A800	43008	Механічне передавальне число не може бути 0 або від'ємним числом.	Змінити механічне передавальне число.
A801	43009	Значення UnitsPerRotation не може бути 0 або від'ємним числом.	Змініть значення UnitsPerRotation.
A802	43010	Помилка типу осі	Змініть тип осі.

ErrorID		Значення	Як боротися з
Hex	Десятковий		
A803	43011	Значення модуля не може бути недодатним числом.	Змініть значення модуля.
A808	43016	Ex_Move виконується знову, коли виконання інструкції ще не завершено.	Змінити час виконання інструкції.
A809	43017	Перед виконанням Ex_Stop потрібно спочатку виконати Ex_Move.	Змінити час виконання інструкції.
A80A	43018	Значення RoundPhase має бути більше 0.	Змініть значення RoundPhase на значення, яке перевищує 0.
A80B	43019	Значення StopPhase має бути більше 0 і менше значення RoundPhase.	Змініть значення StopPhase на значення, яке більше за 0 і менше значення RoundPhase.
A810	43024	Значення TorqueRamp дорівнює 0 або від'ємне число.	Змініть значення TorqueRamp.
A818	43032	Значення Lag є від'ємним числом.	Змініть значення Lag як невід'ємне число.
A819	43033	Значення HoldTime є від'ємним числом.	Змініть значення HoldTime як невід'ємне число.
A820	43040	Кількість точок штовхача досягає максимального значення.	Переконайтеся, що кількість точок штовхача не перевищує максимального значення.
A821	43041	Значення MasterPos перевищує допустимий діапазон.	Переконайтеся, що головне положення точки штовхача знаходиться в межах діапазону для головної осі, встановленого в програмному забезпеченні.
A830x	43056	Було використано ідентифікаційний номер у групі осей.	Змінити ідентифікаційний номер.
A831	43057	Вісь було налаштовано як вісь у поточній групі осей.	Змініть номер осі.
A838	43064	Робочий стан групи осей перервано.	Змінити час виконання інструкції.
A839	43065	Коли інструкція виконується, параметри Pause і Stop мають мати значення FALSE.	Змінити час виконання інструкції.
A83A	43066	У групі осей не виділено жодної осі.	Змінити час виконання інструкції.
A83B	43067	Осі в групі осей не знаходяться в стані зупинки.	Змінити час виконання інструкції.

12.3 Діагностика системних несправностей за допомогою кодів системних помилок

Коли індикатор ERR контролера руху блимає або постійно горить, ви можете дізнатися причину помилки та вирішити проблему за допомогою значень змінних системної помилки.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
1000	4096	Не вдалося виявити внутрішню оперативну пам’ять	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного ввімкнення.
1001	4097	Не вдалося виявити внутрішній Flash	
1002	4098	Не вдалося виявити порт розширення	
1003	4099	Внутрішня напруга ненормальна (LV)	Відрегулюйте вхідну напругу до 24 В на порту живлення.
1004	4100	Помилка ініціалізації Flash	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного ввімкнення.
1005	4101	Не вдалося виявити Flash ID.	
1007	4103	Помилка доступу до флеш-пам’яті в зоні Ethernet.	Зверніться до місцевих техніків, якщо проблема не зникає після повторного завантаження програми та відновлення заводських налаштувань.
1008	4104	Не вдалося отримати доступ до флеш-пам’яті в зоні розширення.	
1009	4105	Збій доступу до флеш-пам'яті в програмній області.	
100A	4106	Помилка доступу до флеш-пам’яті в області конфігурації осі.	
100B	4107	Помилка доступу до флеш-пам'яті в області конфігурації завдань.	
100C	4108	Помилка доступу до флеш-пам’яті в області конфігурації CANopen.	
100D	4109	Не вдалося отримати доступ до флеш-пам’яті в області конфігурації обладнання.	
100E	4110	Помилка доступу до флеш-пам’яті в області CAM.	
100F	4111	Помилка доступу до флеш-пам’яті (таблиці керування флеш-пам’яттю).	
1010	4112	Помилка доступу до флеш-пам’яті (аркуш 1 у таблиці керування флеш-пам’яттю).	
1011	4113	Доступ до флеш-пам’яті (аркуш 2 у таблиці керування флеш-пам’яттю) не вдається.	
1012	4114	Зчитування флеш-пам'яті не вдалося.	
1013	4115	Запис у флеш не вдалося.	
1014	4116	Не вдалося видалити вміст флеш-пам’яті.	
1015	4117	Ідентифікатор файлу ЧПУ виходить за межі допустимого діапазону	Перевірте, чи ідентифікатор файлу ЧПК перевищує 64. Оновіть програмне забезпечення та повторно завантажте програму, якщо помилка не зникає після повторного завантаження програми.
1016	4118	Розмір файлу ЧПК перевищує діапазон	Файл CNC завеликий за розміром. Зменшіть розмір і перезавантажте програму.
1017	4119	Положення інкрементного кодера 1 різко змінюється за короткий час.	Перевірте, чи не надто швидкий вхід кодера, або збільште роздільну здатність кодера.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
1018	4120	Положення інкрементного кодера 2 різко змінюється за короткий час.	Перевірте, чи не надто швидкий вхід кодера, або збільште роздільну здатність кодера.
1019	4121	Системний стек використано.	У програмі занадто багато проміжних змінних. Змініть програму.
101A	4122	Файл Retain завеликий.	Забагато змінних Retain. Зменшіть кількість змінних Retain, а потім повторно завантажте програму.
101B	4123	Помилка доступу до файлу збереження.	Повторно завантажте програму після відновлення системи до заводських налаштувань.
101C	4124	Помилка завантаження даних конфігурації EIP.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
101D	4125	Помилка читання таблиці Cam.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1020	4128	Неприпустимий тип файлу конфігурації	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1021	4129	Помилка зберігання файлів	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1022	4130	Не вдалося прочитати файл	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1023	4131	Помилка перевірки файлу	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1024	4132	Розмір файлу перевищує допустимий діапазон.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1025	4133	Не вдалося прочитати програмний файл.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1026	4134	Не вдалося прочитати файл конфігурації CANopen	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1027	4135	Не вдалося прочитати файл конфігурації руху	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1028	4136	Не вдалося прочитати файл конфігурації системи	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1029	4137	Не вдалося прочитати файл конфігурації завдання	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102A	4138	Не вдалося прочитати файл конфігурації розширення	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102B	4139	Не вдалося прочитати файл Cam.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102C	4140	Не вдалося прочитати файл RETAIN.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102D	4141	Не вдалося прочитати файл ID.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102E	4142	Не вдалося прочитати зашифрований файл.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
102F	4143	Не вдалося прочитати файл ЧПК.	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
1030	4144	Обладнання не відповідає.	Зверніться до місцевих техніків.
1031	4145	Помилка обробки аналізатора позиції	Зверніться до місцевих техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
1401	5121	Помилка ініціалізації Ethernet LAN1.	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного ввімкнення.
1402	5122	Буфер Ethernet LAN1 переповнюється	
1403	5123	Помилка надсилання даних через Ethernet LAN1.	
1404	5124	Не вдалося надіслати розподіл буферної пам’яті через Ethernet.	
1405	5125	IP-адреса іншого пристрою збігається з IP-адресою ПЛК у мережі Ethernet.	Переконайтеся, що в мережі немає ідентичних IP-адрес, змінивши IP-адресу іншого пристрою або контролера.
1601	5633	Помилка ініціалізації Ethernet LAN2.	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного ввімкнення.
1602	5634	Буфер Ethernet LAN2 переповнюється.	
1603	5635	Повна буферна область для зв'язку через USB	Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
3000	12288	Кількість входів або кількість виходів перевищує обмеження 32.	Скиньте кількість вхідних і вихідних змінних у самовизначеному POU та переконайтеся, що кількість вхідних і вихідних змінних не перевищує 32.
3001	12289	Ємність одного POU більше 65535 байт.	Змініть ємність змінних у POU, щоб зменшити зайнятість змінних у пам’яті.
3002	12290	Кількість POU більше 1000.	Зменшіть кількість POU, викликаних завданням, і повторно завантажте програму.
3003	12291	Тип POU є незаконним.	Оновіть програмне забезпечення, якщо помилка все ще існує після повторної компіляції та повторного завантаження програми та повторного увімкнення продукту.
3004	12292	Типи параметрів у програмі є недопустимими.	
3005	12293	Помилка адреси зміщення змінної в програмі	
3006	12294	Типи даних параметрів є недопустимими в програмі.	
3007	12295	Діапазон стрибків у програмі заборонений.	
3008	12296	Вирівнювання розподілу пам'яті програми неправильне.	
3009	12297	Неправильне вирівнювання пам’яті кодера віртуальної осі.	
300A	12298	Біт, до якого звернулися, перевищує діапазон. (Лише Bit0~Bit7 можна отримати доступ.)	Оновіть програмне забезпечення, якщо помилка все ще існує після повторної компіляції та повторного завантаження програми та повторного увімкнення продукту.
300B	12299	Виявлено, що типи даних у програмі є незаконними.	Оновіть програмне забезпечення, якщо помилка все ще існує після повторної компіляції та повторного завантаження програми та повторного увімкнення продукту.
300C	12300	Довжина типу даних String завелика.	Кількість символів у типі даних String завелика. Оновіть програмне забезпечення, якщо помилка все ще існує після модифікації програми, повторної компіляції та повторного завантаження програми.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
300D	12301	Неприпустимий метод адресації для змінних	Оновіть програмне забезпечення, якщо помилка все ще існує після повторної компіляції та повторного завантаження програми та повторного увімкнення продукту.
300E	12302	Неправильна постановка пріоритету завдання	Встановіть належний пріоритет завдання та час циклу для завдання
300F	12303	Виняток виникає під час доступу до пам'яті контролера	Перевірте, чи якийсь індекс не виходить за межі масиву в програмі. Вказівник використовується, не присвоюючи йому значення, або використовується, присвоюючи йому неправильне значення. Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
3020	12320	Контрольна сума завантаженої конфігурації осі є недопустимою.	Переконайтеся, що конфігурація осі правильна, а потім повторно скомпілюйте та повторно завантажте конфігурацію.
3021	12321	Контрольна сума конфігурації завантаженого розширення є незаконною.	Переконайтеся, що конфігурація розширення правильна, а потім повторно скомпілюйте та повторно завантажте конфігурацію.
3022	12322	Контрольна сума завантаженої програми є незаконною.	Переконайтеся, що програма правильна, а потім знову скомпілюйте та завантажте програму.
3023	12323	Контрольна сума завантажених даних завдання є незаконною.	Переконайтеся, що налаштування завдання правильні, а потім повторно скомпілюйте та повторно завантажте конфігурацію.
3024	12324	Контрольна сума завантаженої конфігурації CANopen є недопустимою.	Переконайтеся, що конфігурація CANopen правильна, а потім повторно скомпілюйте та завантажте конфігурацію.
3025	12325	Контрольна сума завантаженої конфігурації апаратного забезпечення є незаконною.	Переконайтеся, що апаратна конфігурація правильна, а потім повторно скомпілюйте та повторно завантажте конфігурацію.
3026	12326	Сторожовий тайм-аут	Перевірте, чи програма правильна, чи існує цикл, з якого виконання програми не може вийти, коли настає час очікування виконання програми.
3027	12327	Помилка виклику кінцевого автомата осі.	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного завантаження програми та відновлення заводських налаштувань.
3028	12328	Помилка аналізу списку ЧПУ	Перевірте, чи правильний файл CNC, і повторно завантажте програму.
3029	12329	Помилка аналізу файлу ЧПУ	Перевірте, чи правильний файл CNC, і повторно завантажте програму.
302A	12330	Помилка аналізу файлу Cam.	Перевірте, чи правильний файл cam. Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
3031	12337	Помилка аналізу файлу вихідного коду.	Перевірте, чи правильний файл вихідного коду. Зверніться до місцевих технічних спеціалістів, якщо проблема не зникає після повторного завантаження програми.
3050	12368	Фактичний час виконання завдання з пріоритетом 0 перевищує встановлений час тайм-ауту сторожового таймера.	Зверніться до місцевих дистриб'юторів

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
3051	12369	Фактичний час виконання завдання з пріоритетом 1 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3052	12370	Фактичний час виконання завдання priority2 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3053	12371	Фактичний час виконання завдання з пріоритетом 3 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3054	12372	Фактичний час виконання завдання з пріоритетом 4 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3055	12373	Фактичний час виконання завдання з пріоритетом 5 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3056	12374	Фактичний час виконання завдання з пріоритетом 6 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перегляньте програму або повторно завантажте оновлену програму.
3057	12375	Фактичний час виконання завдання з пріоритетом 7 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перегляньте програму або повторно завантажте оновлену програму.
3058	12376	Фактичний час виконання завдання з пріоритетом 8 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
3059	12377	Фактичний час виконання завдання з пріоритетом 9 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305A	12378	Фактичний час виконання завдання з пріоритетом 10 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305B	12379	Фактичний час виконання завдання з пріоритетом 11 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305C	12380	Фактичний час виконання завдання з пріоритетом 12 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305D	12381	Фактичний час виконання завдання з пріоритетом 13 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305E	12382	Фактичний час виконання завдання з пріоритетом 14 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
305F	12383	Фактичний час виконання завдання з пріоритетом 15 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3060	12384	Фактичний час виконання завдання з пріоритетом 16 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3061	12385	Фактичний час виконання завдання з пріоритетом 17 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
3062	12386	Фактичний час виконання завдання з пріоритетом 18 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3063	12387	Фактичний час виконання завдання з пріоритетом 19 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3064	12388	Фактичний час виконання завдання з пріоритетом 20 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3065	12389	Фактичний час виконання завдання з пріоритетом 21 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3066	12390	Фактичний час виконання завдання з пріоритетом 22 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3067	12391	Фактичний час виконання завдання з пріоритетом 23 перевищує встановлений тайм-аут сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3068	12392	Фактичний час виконання завдання з пріоритетом 24 перевищує встановлений час тайм-ауту сторожового таймера.	1. Скиньте сторожовий час до більшого значення для завдання. 2. Перевірте, чи є нескінченний цикл у програмі, яку викликає завдання. 3. Перезавантажте його після зміни програми.
3069	12393	Фактичний час виконання завдання з пріоритетом 25 перевищує встановлений час тайм-ауту сторожового таймера.	Зверніться до місцевих техніків.
306A	12394	Фактичний час виконання завдання з пріоритетом 26 перевищує встановлений час тайм-ауту сторожового таймера.	Зверніться до місцевих техніків.
306B	12395	Фактичний час виконання завдання з пріоритетом 27 перевищує встановлений тайм-аут сторожового таймера.	Зверніться до місцевих техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
306C	12396	Фактичний час виконання завдання з пріоритетом 28 перевищує встановлений тайм-аут сторожового таймера.	Зверніться до місцевих техніків.
306D	12397	Фактичний час виконання завдання з пріоритетом 29 перевищує встановлений тайм-аут сторожового таймера.	Зверніться до місцевих техніків.
306E	12398	Фактичний час виконання завдання з пріоритетом 30 перевищує встановлений час тайм-ауту сторожового таймера.	Зверніться до місцевих техніків.
306F	12399	Фактичний час виконання завдання з пріоритетом 31 перевищує встановлений тайм-аут сторожового таймера.	Зверніться до місцевих техніків.
4000	16384	Помилка читання та запису даних SD-карти ПЛК.	1. Перевірте, чи правильно встановлено SD-карту. 2. Перевірте, чи SD-карта не пошкоджена.
4003	16387	Файл на SD-карті доступний лише для читання.	Змініть файл на SD-карті як файл читання-запису
4100	16640	Дані у файлі резервної копії проекту змінено або формат даних неправильний.	1. Переконайтеся, що дані у файлі резервної копії проекту не змінено. 2. Переконайтеся, що файл резервної копії проекту створено CANopen Builder.
4101	16641	Формат даних файлу резервної копії змінної RETAIN неправильний.	1. Переконайтеся, що дані у файлі резервної копії змінної RETAIN не змінено. 2. Переконайтеся, що файл резервної копії змінної RETAIN створено CANopen Builder.
4102	16642	Під час відновлення дані у файлі резервної копії змінної RETAIN частково відрізняються від даних у ПЛК.	Переконайтеся, що дані у файлі резервної копії змінної RETAIN узгоджуються з даними в ПЛК.
4103	16643	Під час відновлення дані у файлі резервної копії змінної RETAIN повністю відрізняються від даних у ПЛК.	Переконайтеся, що дані у файлі резервної копії змінної RETAIN узгоджуються з даними в ПЛК.
4104	16644	Під час відновлення модель контролера у файлі резервної копії проекту відрізняється від моделі фактично підключеного ПЛК.	Переконайтеся, що модель контролера у файлі резервної копії проекту збігається з моделлю фактично підключеного ПЛК.
4105	16645	Ідентифікатор програми у файлі резервної копії відрізняється від ідентифікатора поточного ПЛК.	Переконайтеся, що ідентифікатор програми у файлі резервної копії збігається з ідентифікатором поточного ПЛК.
4106	16646	Пароль ПЛК у файлі резервної копії відрізняється від пароля поточного ПЛК.	Переконайтеся, що пароль ПЛК у файлі резервної копії збігається з паролем поточного ПЛК.
5000	20480	Помилка перевірки зв'язку з розширенням.	Зверніться до місцевих технічних спеціалістів, якщо помилка не зникає після повторного ввімкнення.
5001	20481	Час очікування внутрішнього зв'язку	
5100	20736	Фактично підключений модуль не відповідає налаштованому в програмному забезпеченні.	Переконайтеся, що модуль, фактично підключений до правої сторони ПЛК, відповідає модулю, налаштованому в програмному забезпеченні, а потім повторно завантажте конфігурацію.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
5200	20992	Буфер для отримання даних CANopen заповнений.	Налаштуйте конфігурацію CANopen і перевірте налаштування завдання.
5201	20993	Буфер для надсилання даних CANopen заповнений.	
5300	21248	Буфер для отримання даних руху заповнений.	Налаштуйте конфігурацію руху та перевірте налаштування завдання.
5301	21249	Буфер для надсилання даних руху заповнений.	
C000	49152	Ідентифікатор помилки для 1-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C001	49153	Ідентифікатор помилки для 1-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C002	49154	Ідентифікатор помилки для 1-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C003	49155	Ідентифікатор помилки для 1-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C004	49156	Ідентифікатор помилки для 1-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C005	49157	Ідентифікатор помилки для 1-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C007	49159	Ідентифікатор помилки для 1-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C008	49160	Ідентифікатор помилки для 1-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C100	49408	Ідентифікатор помилки для 2-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C101	49409	Ідентифікатор помилки для 2-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C102	49410	Ідентифікатор помилки для 2-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C103	49411	Ідентифікатор помилки для 2-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C104	49412	Ідентифікатор помилки для 2-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
C105	49413	Ідентифікатор помилки для 2-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C107	49415	Ідентифікатор помилки для 2-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C108	49416	Ідентифікатор помилки для 2-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C200	49664	Ідентифікатор помилки для 3-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C201	49665	Ідентифікатор помилки для 3-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C202	49666	Ідентифікатор помилки для 3-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C203	49667	Ідентифікатор помилки для 3-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C204	49668	Ідентифікатор помилки для 3-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C205	49669	Ідентифікатор помилки для 3-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C207	49671	Ідентифікатор помилки для 3-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C208	49672	Ідентифікатор помилки для 3-го правого модуля, 16#1608: калібрування або CJC неправильні.	Поверніть модуль на завод для калібрування.
C300	49920	Ідентифікатор помилки для 4-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C301	49921	Ідентифікатор помилки для 4-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C302	49922	Ідентифікатор помилки для 4-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C303	49923	Ідентифікатор помилки для 4-го правого модуля, 16#1603: не вдається увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
C304	49924	Ідентифікатор помилки для 4-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C305	49925	Ідентифікатор помилки для 4-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C307	49927	Ідентифікатор помилки для 4-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C308	49928	Ідентифікатор помилки для 4-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C400	50176	Ідентифікатор помилки для 5-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C401	50177	Ідентифікатор помилки для 5-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C402	50178	Ідентифікатор помилки для 5-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C403	50179	Ідентифікатор помилки для 5-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C404	50180	Ідентифікатор помилки для 5-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C405	50181	Ідентифікатор помилки для 5-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C407	50183	Ідентифікатор помилки для 5-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C408	50184	Ідентифікатор помилки для 5-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C500	50432	Ідентифікатор помилки для 6-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C501	50433	Ідентифікатор помилки для 6-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C502	50434	Ідентифікатор помилки для 6-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
C503	50435	Ідентифікатор помилки для 6-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C504	50436	Ідентифікатор помилки для 6-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C505	50437	Ідентифікатор помилки для 6-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C507	50439	Ідентифікатор помилки для 6-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C508	50440	Ідентифікатор помилки для 6-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C600	50688	Ідентифікатор помилки для 7-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C601	50689	Ідентифікатор помилки для 7-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C602	50690	Ідентифікатор помилки для 7-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C603	50691	Ідентифікатор помилки для 7-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C604	50692	Ідентифікатор помилки для 7-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C605	50693	Ідентифікатор помилки для 7-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C607	50695	Ідентифікатор помилки для 7-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C608	50696	Ідентифікатор помилки для 7-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C700	50944	Ідентифікатор помилки для 8-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C701	50945	Ідентифікатор помилки для 8-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
C702	50946	Ідентифікатор помилки для 8-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C703	50947	Ідентифікатор помилки для 8-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C704	50948	Ідентифікатор помилки для 8-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C705	50949	Ідентифікатор помилки для 8-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C707	50951	Ідентифікатор помилки для 8-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C708	50952	Ідентифікатор помилки для 8-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C800	51200	Ідентифікатор помилки для 9-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C801	51201	Ідентифікатор помилки для 9-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C802	51202	Ідентифікатор помилки для 9-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C803	51203	Ідентифікатор помилки для 9-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C804	51204	Ідентифікатор помилки для 9-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C805	51205	Ідентифікатор помилки для 9-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C807	51207	Ідентифікатор помилки для 9-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C808	51208	Ідентифікатор помилки для 9-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
C900	51456	Ідентифікатор помилки для 10-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
C901	51457	Ідентифікатор помилки для 10-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C902	51458	Ідентифікатор помилки для 10-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C903	51459	Ідентифікатор помилки для 10-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C904	51460	Ідентифікатор помилки для 10-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
C905	51461	Ідентифікатор помилки для 10-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
C907	51463	Ідентифікатор помилки для 10-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
C908	51464	Ідентифікатор помилки для 10-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
CA00	51712	Ідентифікатор помилки для 11-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CA01	51713	Ідентифікатор помилки для 11-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CA02	51714	Ідентифікатор помилки для 11-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CA03	51715	Ідентифікатор помилки для 11-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CA04	51716	Ідентифікатор помилки для 11-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CA05	51717	Ідентифікатор помилки для 11-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
CA07	51719	Ідентифікатор помилки для 11-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
CA08	51720	Ідентифікатор помилки для 11-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
CB00	51968	Ідентифікатор помилки для 12-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CB01	51969	Ідентифікатор помилки для 12-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CB02	51970	Ідентифікатор помилки для 12-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CB03	51971	Ідентифікатор помилки для 12-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CB04	51972	Ідентифікатор помилки для 12-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CB05	51973	Ідентифікатор помилки для 12-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
CB07	51975	Ідентифікатор помилки для 12-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
CB08	51976	Ідентифікатор помилки для 12-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
CC00	52224	Ідентифікатор помилки для 13-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CC01	52225	Ідентифікатор помилки для 13-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CC02	52226	Ідентифікатор помилки для 13-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CC03	52227	Ідентифікатор помилки для 13-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CC04	52228	Ідентифікатор помилки для 13-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CC05	52229	Ідентифікатор помилки для 13-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
CC07	52231	Ідентифікатор помилки для 13-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
CC08	52232	Ідентифікатор помилки для 13-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
CD00	52480	Ідентифікатор помилки для 14-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CD01	52481	Ідентифікатор помилки для 14-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CD02	52482	Ідентифікатор помилки для 14-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CD03	52483	Ідентифікатор помилки для 14-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CD04	52484	Ідентифікатор помилки для 14-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CD05	52485	Ідентифікатор помилки для 14-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
CD07	52487	Ідентифікатор помилки для 14-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
CD08	52488	Ідентифікатор помилки для 14-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
CE00	52736	Ідентифікатор помилки для 15-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CE01	52737	Ідентифікатор помилки для 15-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CE02	52738	Ідентифікатор помилки для 15-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CE03	52739	Ідентифікатор помилки для 15-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CE04	52740	Ідентифікатор помилки для 15-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CE05	52741	Ідентифікатор помилки для 15-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
CE07	52743	Ідентифікатор помилки для 15-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
CE08	52744	Ідентифікатор помилки для 15-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
CF00	52992	Ідентифікатор помилки для 16-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CF01	52993	Ідентифікатор помилки для 16-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CF02	52994	Ідентифікатор помилки для 16-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CF03	52995	Ідентифікатор помилки для 16-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CF04	52996	Ідентифікатор помилки для 16-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
CF05	52997	Ідентифікатор помилки для 16-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
CF07	52999	Ідентифікатор помилки для 16-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
CF08	53000	Ідентифікатор помилки для 16-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D000	53248	Ідентифікатор помилки для 17-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D001	53249	Ідентифікатор помилки для 17-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D002	53250	Ідентифікатор помилки для 17-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D003	53251	Ідентифікатор помилки для 17-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D004	53252	Ідентифікатор помилки для 17-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
D005	53253	Ідентифікатор помилки для 17-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D007	53255	Ідентифікатор помилки для 17-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D008	53256	Ідентифікатор помилки для 17-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D100	53504	Ідентифікатор помилки для 18-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D101	53505	Ідентифікатор помилки для 18-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D102	53506	Ідентифікатор помилки для 18-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D103	53507	Ідентифікатор помилки для 18-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D104	53508	Ідентифікатор помилки для 18-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D105	53509	Ідентифікатор помилки для 18-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D107	53511	Ідентифікатор помилки для 18-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D108	53512	Ідентифікатор помилки для 18-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D200	53760	Ідентифікатор помилки для 19-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D201	53761	Ідентифікатор помилки для 19-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D202	53762	Ідентифікатор помилки для 19-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D203	53763	Ідентифікатор помилки для 19-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
D204	53764	Ідентифікатор помилки для 19-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D205	53765	Ідентифікатор помилки для 19-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D207	53767	Ідентифікатор помилки для 19-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D208	53768	Ідентифікатор помилки для 19-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D300	54016	Ідентифікатор помилки для 20-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D301	54017	Ідентифікатор помилки для 20-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D302	54018	Ідентифікатор помилки для 20-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D303	54019	Ідентифікатор помилки для 20-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D304	54020	Ідентифікатор помилки для 20-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D305	54021	Ідентифікатор помилки для 20-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D307	54023	Ідентифікатор помилки для 20-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D308	54024	Ідентифікатор помилки для 20-го правого модуля, 16#1608: калібрування або CJC є ненормальними.	Поверніть модуль на завод для калібрування.
D400	54272	Ідентифікатор помилки для 21-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D401	54273	Ідентифікатор помилки для 21-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D402	54274	Ідентифікатор помилки для 21-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
D403	54275	Ідентифікатор помилки для 21-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D404	54276	Ідентифікатор помилки для 21-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D405	54277	Ідентифікатор помилки для 21-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D407	54279	Ідентифікатор помилки для 21-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D408	54280	Ідентифікатор помилки для 21-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D500	54528	Ідентифікатор помилки для 22-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D501	54529	Ідентифікатор помилки для 22-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D502	54530	Ідентифікатор помилки для 22-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D503	54531	Ідентифікатор помилки для 22-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D504	54532	Ідентифікатор помилки для 22-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D505	54533	Ідентифікатор помилки для 22-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D507	54535	Ідентифікатор помилки для 22-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D508	54536	Ідентифікатор помилки для 22-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D600	54784	Ідентифікатор помилки для 23-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D601	54785	Ідентифікатор помилки для 23-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
D602	54786	Ідентифікатор помилки для 23-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D603	54787	Ідентифікатор помилки для 23-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D604	54788	Ідентифікатор помилки для 23-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D605	54789	Ідентифікатор помилки для 23-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D607	54791	Ідентифікатор помилки для 23-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D608	54792	Ідентифікатор помилки для 23-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D700	55040	Ідентифікатор помилки для 24-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D701	55041	Ідентифікатор помилки для 24-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D702	55042	Ідентифікатор помилки для 24-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D703	55043	Ідентифікатор помилки для 24-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D704	55044	Ідентифікатор помилки для 24-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D705	55045	Ідентифікатор помилки для 24-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D707	55047	Ідентифікатор помилки для 24-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D708	55048	Ідентифікатор помилки для 24-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D800	55296	Ідентифікатор помилки для 25-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
D801	55297	Ідентифікатор помилки для 25-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D802	55298	Ідентифікатор помилки для 25-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D803	55299	Ідентифікатор помилки для 25-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D804	55300	Ідентифікатор помилки для 25-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D805	55301	Ідентифікатор помилки для 25-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D807	55303	Ідентифікатор помилки для 25-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D808	55304	Ідентифікатор помилки для 25-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
D980	55680	Ідентифікатор помилки для 26-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D981	55681	Ідентифікатор помилки для 26-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D982	55682	Ідентифікатор помилки для 26-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D983	55683	Ідентифікатор помилки для 26-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D984	55684	Ідентифікатор помилки для 26-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
D985	55685	Ідентифікатор помилки для 26-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
D987	55687	Ідентифікатор помилки для 26-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
D988	55688	Ідентифікатор помилки для 26-го правого модуля, 16#1608: калібрування або CJC є ненормальними.	Поверніть модуль на завод для калібрування.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
DA00	55808	Ідентифікатор помилки для 27-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DA01	55809	Ідентифікатор помилки для 27-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DA02	55810	Ідентифікатор помилки для 27-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DA03	55811	Ідентифікатор помилки для 27-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DA04	55812	Ідентифікатор помилки для 27-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DA05	55813	Ідентифікатор помилки для 27-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
DA07	55815	Ідентифікатор помилки для 27-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
DA08	55816	Ідентифікатор помилки для 27-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
DB00	56064	Ідентифікатор помилки для 28-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DB01	56065	Ідентифікатор помилки для 28-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DB02	56066	Ідентифікатор помилки для 28-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DB03	56067	Ідентифікатор помилки для 28-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DB04	56068	Ідентифікатор помилки для 28-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DB05	56069	Ідентифікатор помилки для 28-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
DB07	56071	Ідентифікатор помилки для 28-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
DB08	56072	Ідентифікатор помилки для 28-го правого модуля, 16#1608: калібрування або CJC є ненормальними.	Поверніть модуль на завод для калібрування.
DC00	56320	Ідентифікатор помилки для 29-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DC01	56321	Ідентифікатор помилки для 29-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DC02	56322	Ідентифікатор помилки для 29-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DC03	56323	Ідентифікатор помилки для 29-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DC04	56324	Ідентифікатор помилки для 29-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DC05	56325	Ідентифікатор помилки для 29-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
DC07	56327	Ідентифікатор помилки для 29-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
DC08	56328	Ідентифікатор помилки для 29-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
DD00	56576	Ідентифікатор помилки для 30-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DD01	56577	Ідентифікатор помилки для 30-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DD02	56578	Ідентифікатор помилки для 30-го правого модуля, 16#1602: Ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DD03	56579	Ідентифікатор помилки для 30-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DD04	56580	Ідентифікатор помилки для 30-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DD05	56581	Ідентифікатор помилки для 30-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
DD07	56583	Ідентифікатор помилки для 30-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
DD08	56584	Ідентифікатор помилки для 30-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
DE00	56832	Ідентифікатор помилки для 31-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DE01	56833	Ідентифікатор помилки для 31-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DE02	56834	Ідентифікатор помилки для 31-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DE03	56835	Ідентифікатор помилки для 31-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DE04	56836	Ідентифікатор помилки для 31-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DE05	56837	Ідентифікатор помилки для 31-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
DE07	56839	Ідентифікатор помилки для 31-го правого модуля, 16#1607: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
DE08	56840	Ідентифікатор помилки для 31-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
DF00	57088	Ідентифікатор помилки для 32-го правого модуля, 16#1600: Ідентифікатор модуля розширення перевищує діапазон.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DF01	57089	Ідентифікатор помилки для 32-го правого модуля, 16#1601: ідентифікатор модуля розширення не може бути встановлений.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DF02	57090	Ідентифікатор помилки для 32-го правого модуля, 16#1602: ідентифікатор модуля розширення дублюється.	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DF03	57091	Ідентифікатор помилки для 32-го правого модуля, 16#1603: не вдалося увійти в режим RUN	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.
DF04	57092	Ідентифікатор помилки для 32-го правого модуля, 16#1604: час очікування зв'язку модуля	Переконайтеся, що модуль належним чином підключено до модуля ЦП, і знову УВІМКНІТЬ живлення. Якщо помилка все ще виникає, зверніться до техніків.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
DF05	57093	Ідентифікатор помилки для 32-го правого модуля, 16#1605: збій обладнання	Поверніть модуль на завод для ремонту.
DF07	57095	Ідентифікатор помилки для 32-го правого модуля, 16#1607: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму є нормальним.
DF08	57096	Ідентифікатор помилки для 32-го правого модуля, 16#1608: калібрування або CJC не відповідає нормі.	Поверніть модуль на завод для калібрування.
E001	57345	Ідентифікатор помилки для 1-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E002	57346	Ідентифікатор помилки для 1-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E004	57348	Ідентифікатор помилки для 1-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E007	57351	Ідентифікатор помилки для 1-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E008	57352	Ідентифікатор помилки для 1-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E009	57353	Ідентифікатор помилки для 1-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
E00A	57354	Ідентифікатор помилки для 1-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E00B	57355	Ідентифікатор помилки для 1-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E00C	57356	Ідентифікатор помилки для 1-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E00D	57357	Ідентифікатор помилки для 1-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E00E	57358	Ідентифікатор помилки для 1-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
E00F	57359	Ідентифікатор помилки для 1-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E0F1	57585	Ідентифікатор помилки для 1-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E0F2	57586	Ідентифікатор помилки для 1-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E0F7	57591	Ідентифікатор помилки для 1-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E0F8	57592	Ідентифікатор помилки для 1-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E0F9	57593	Ідентифікатор помилки для 1-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E0FA	57594	Ідентифікатор помилки для 1-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E0FB	57595	Ідентифікатор помилки для 1-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E0FC	57596	Ідентифікатор помилки для 1-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E0FD	57597	Ідентифікатор помилки для 1-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E101	57601	Ідентифікатор помилки для 2-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E102	57602	Ідентифікатор помилки для 2-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E104	57604	Ідентифікатор помилки для 2-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E107	57607	Ідентифікатор помилки для 2-го правого модуля, 16#1807: Помилка температури CJC	Поверніть модуль на завод для ремонту.
E108	57608	Ідентифікатор помилки для 2-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E109	57609	Ідентифікатор помилки для 2-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E10A	57610	Ідентифікатор помилки для 2-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E10B	57611	Ідентифікатор помилки для 2-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E10C	57612	Ідентифікатор помилки для 2-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E10D	57613	Ідентифікатор помилки для 2-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
E10E	57614	Ідентифікатор помилки для 2-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
E10F	57615	Ідентифікатор помилки для 2-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
E1F1	57841	Ідентифікатор помилки для 2-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E1F2	57842	Ідентифікатор помилки для 2-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E1F7	57847	Ідентифікатор помилки для 2-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E1F8	57848	Ідентифікатор помилки для 2-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E1F9	57849	Ідентифікатор помилки для 2-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E1FA	57850	Ідентифікатор помилки для 2-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E1FB	57851	Ідентифікатор помилки для 2-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN є неправильною.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E1FC	57852	Ідентифікатор помилки для 2-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E1FD	57853	Ідентифікатор помилки для 2-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E201	57857	Ідентифікатор помилки для 3-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E202	57858	Ідентифікатор помилки для 3-го правого модуля, 16#1802: збій апаратного забезпечення модуля	Поверніть модуль на завод для ремонту.
E204	57860	Ідентифікатор помилки для 3-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E207	57863	Ідентифікатор помилки для 3-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E208	57864	Ідентифікатор помилки для 3-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
E209	57865	Ідентифікатор помилки для 3-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E20A	57866	Ідентифікатор помилки для 3-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E20B	57867	Ідентифікатор помилки для 3-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E20C	57868	Ідентифікатор помилки для 3-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
E20D	57869	Ідентифікатор помилки для 3-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E20E	57870	Ідентифікатор помилки для 3-го правого модуля, 16#180E: Сигнал (температура), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
E20F	57871	Ідентифікатор помилки для 3-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E2F1	58097	Ідентифікатор помилки для 3-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E2F2	58098	Ідентифікатор помилки для 3-го правого модуля, 16#18F2: збій апаратного забезпечення модуля	Поверніть модуль на завод для ремонту.
E2F7	58103	Ідентифікатор помилки для 3-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E2F8	58104	Ідентифікатор помилки для 3-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E2F9	58105	Ідентифікатор помилки для 3-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E2FA	58106	Ідентифікатор помилки для 3-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E2FB	58107	Ідентифікатор помилки для 3-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN є неправильною.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E2FC	58108	Ідентифікатор помилки для 3-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E2FD	58109	Ідентифікатор помилки для 3-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E301	58113	Ідентифікатор помилки для 4-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E302	58114	Ідентифікатор помилки для 4-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E304	58116	Ідентифікатор помилки для 4-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E307	58119	Ідентифікатор помилки для 4-го правого модуля, 16#1807: Помилка температури CJC	Поверніть модуль на завод для ремонту.
E308	58120	Ідентифікатор помилки для 4-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E309	58121	Ідентифікатор помилки для 4-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E30A	58122	Ідентифікатор помилки для 4-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E30B	58123	Ідентифікатор помилки для 4-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E30C	58124	Ідентифікатор помилки для 4-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E30D	58125	Ідентифікатор помилки для 4-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E30E	58126	Ідентифікатор помилки для 4-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
E30F	58127	Ідентифікатор помилки для 4-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
E3F1	58353	Ідентифікатор помилки для 4-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E3F2	58354	Ідентифікатор помилки для 4-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E3F7	58359	Ідентифікатор помилки для 4-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E3F8	58360	Ідентифікатор помилки для 4-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E3F9	58361	Ідентифікатор помилки для 4-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E3FA	58362	Ідентифікатор помилки для 4-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E3FB	58363	Ідентифікатор помилки для 4-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E3FC	58364	Ідентифікатор помилки для 4-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E3FD	58365	Ідентифікатор помилки для 4-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E401	58369	Ідентифікатор помилки для 5-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E402	58370	Ідентифікатор помилки для 5-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E404	58372	Ідентифікатор помилки для 5-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E407	58375	Ідентифікатор помилки для 5-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E408	58376	Ідентифікатор помилки для 5-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
E409	58377	Ідентифікатор помилки для 5-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E40A	58378	Ідентифікатор помилки для 5-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E40B	58379	Ідентифікатор помилки для 5-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E40C	58380	Ідентифікатор помилки для 5-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E40D	58381	Ідентифікатор помилки для 5-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E40E	58382	Ідентифікатор помилки для 5-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
E40F	58383	Ідентифікатор помилки для 5-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E4F1	58609	Ідентифікатор помилки для 5-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E4F2	58610	Ідентифікатор помилки для 5-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E4F7	58615	Ідентифікатор помилки для 5-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E4F8	58616	Ідентифікатор помилки для 5-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E4F9	58617	Ідентифікатор помилки для 5-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E4FA	58618	Ідентифікатор помилки для 5-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E4FB	58619	Ідентифікатор помилки для 5-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E4FC	58620	Ідентифікатор помилки для 5-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E4FD	58621	Ідентифікатор помилки для 5-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E501	58625	Ідентифікатор помилки для 6-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E502	58626	Ідентифікатор помилки для 6-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E504	58628	Ідентифікатор помилки для 6-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E507	58631	Ідентифікатор помилки для 6-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E508	58632	Ідентифікатор помилки для 6-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E509	58633	Ідентифікатор помилки для 6-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E50A	58634	Ідентифікатор помилки для 6-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E50B	58635	Ідентифікатор помилки для 6-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E50C	58636	Ідентифікатор помилки для 6-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E50D	58637	Ідентифікатор помилки для 6-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E50E	58638	Ідентифікатор помилки для 6-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
E50F	58639	Ідентифікатор помилки для 6-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E5F1	58865	Ідентифікатор помилки для 6-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E5F2	58866	Ідентифікатор помилки для 6-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E5F7	58871	Ідентифікатор помилки для 6-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E5F8	58872	Ідентифікатор помилки для 6-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E5F9	58873	Ідентифікатор помилки для 6-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E5FA	58874	Ідентифікатор помилки для 6-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E5FB	58875	Ідентифікатор помилки для 6-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E5FC	58876	Ідентифікатор помилки для 6-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E5FD	58877	Ідентифікатор помилки для 6-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E601	58881	Ідентифікатор помилки для 7-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E602	58882	Ідентифікатор помилки для 7-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E604	58884	Ідентифікатор помилки для 7-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E607	58887	Ідентифікатор помилки для 7-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E608	58888	Ідентифікатор помилки для 7-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E609	58889	Ідентифікатор помилки для 7-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E60A	58890	Ідентифікатор помилки для 7-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E60B	58891	Ідентифікатор помилки для 7-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E60C	58892	Ідентифікатор помилки для 7-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
E60D	58893	Ідентифікатор помилки для 7-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E60E	58894	Ідентифікатор помилки для 7-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
E60F	58895	Ідентифікатор помилки для 7-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
E6F1	59121	Ідентифікатор помилки для 7-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E6F2	59122	Ідентифікатор помилки для 7-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E6F7	59127	Ідентифікатор помилки для 7-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E6F8	59128	Ідентифікатор помилки для 7-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E6F9	59129	Ідентифікатор помилки для 7-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E6FA	59130	Ідентифікатор помилки для 7-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E6FB	59131	Ідентифікатор помилки для 7-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E6FC	59132	Ідентифікатор помилки для 7-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E6FD	59133	Ідентифікатор помилки для 7-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E701	59137	Ідентифікатор помилки для 8-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E702	59138	Ідентифікатор помилки для 8-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E704	59140	Ідентифікатор помилки для 8-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E707	59143	Ідентифікатор помилки для 8-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E708	59144	Ідентифікатор помилки для 8-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E709	59145	Ідентифікатор помилки для 8-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
E70A	59146	Ідентифікатор помилки для 8-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E70B	59147	Ідентифікатор помилки для 8-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E70C	59148	Ідентифікатор помилки для 8-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
E70D	59149	Ідентифікатор помилки для 8-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
E70E	59150	Ідентифікатор помилки для 8-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
E70F	59151	Ідентифікатор помилки для 8-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E7F1	59377	Ідентифікатор помилки для 8-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E7F2	59378	Ідентифікатор помилки для 8-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E7F7	59383	Ідентифікатор помилки для 8-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E7F8	59384	Ідентифікатор помилки для 8-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E7F9	59385	Ідентифікатор помилки для 8-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E7FA	59386	Ідентифікатор помилки для 8-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E7FB	59387	Ідентифікатор помилки для 8-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E7FC	59388	Ідентифікатор помилки для 8-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E7FD	59389	Ідентифікатор помилки для 8-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E801	59393	Ідентифікатор помилки для 9-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E802	59394	Ідентифікатор помилки для 9-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E804	59396	Ідентифікатор помилки для 9-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E807	59399	Ідентифікатор помилки для 9-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E808	59400	Ідентифікатор помилки для 9-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
E809	59401	Ідентифікатор помилки для 9-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
E80A	59402	Ідентифікатор помилки для 9-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
E80B	59403	Ідентифікатор помилки для 9-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E80C	59404	Ідентифікатор помилки для 9-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
E80D	59405	Ідентифікатор помилки для 9-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E80E	59406	Ідентифікатор помилки для 9-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
E80F	59407	Ідентифікатор помилки для 9-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E8F1	59633	Ідентифікатор помилки для 9-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E8F2	59634	Ідентифікатор помилки для 9-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E8F7	59639	Ідентифікатор помилки для 9-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E8F8	59640	Ідентифікатор помилки для 9-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E8F9	59641	Ідентифікатор помилки для 9-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E8FA	59642	Ідентифікатор помилки для 9-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E8FB	59643	Ідентифікатор помилки для 9-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E8FC	59644	Ідентифікатор помилки для 9-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E8FD	59645	Ідентифікатор помилки для 9-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
E901	59649	Ідентифікатор помилки для 10-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E902	59650	Ідентифікатор помилки для 10-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E904	59652	Ідентифікатор помилки для 10-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
E907	59655	Ідентифікатор помилки для 10-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
E908	59656	Ідентифікатор помилки для 10-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
E909	59657	Ідентифікатор помилки для 10-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
E90A	59658	Ідентифікатор помилки для 10-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
E90B	59659	Ідентифікатор помилки для 10-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E90C	59660	Ідентифікатор помилки для 10-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
E90D	59661	Ідентифікатор помилки для 10-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
E90E	59662	Ідентифікатор помилки для 10-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
E90F	59663	Ідентифікатор помилки для 10-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
E9F1	59889	Ідентифікатор помилки для 10-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
E9F2	59890	Ідентифікатор помилки для 10-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
E9F7	59895	Ідентифікатор помилки для 10-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
E9F8	59896	Ідентифікатор помилки для 10-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
E9F9	59897	Ідентифікатор помилки для 10-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
E9FA	59898	Ідентифікатор помилки для 10-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
E9FB	59899	Ідентифікатор помилки для 10-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
E9FC	59900	Ідентифікатор помилки для 10-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
E9FD	59901	Ідентифікатор помилки для 10-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
EA01	59905	Ідентифікатор помилки для 11-го правого модуля, 16#1801: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EA02	59906	Ідентифікатор помилки для 11-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EA04	59908	Ідентифікатор помилки для 11-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
EA07	59911	Ідентифікатор помилки для 11-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
EA08	59912	Ідентифікатор помилки для 11-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
EA09	59913	Ідентифікатор помилки для 11-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
EA0A	59914	Ідентифікатор помилки для 11-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
EA0B	59915	Ідентифікатор помилки для 11-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EA0C	59916	Ідентифікатор помилки для 11-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
EA0D	59917	Ідентифікатор помилки для 11-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
EA0E	59918	Ідентифікатор помилки для 11-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
EA0F	59919	Ідентифікатор помилки для 11-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
EAF1	60145	Ідентифікатор помилки для 11-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EAF2	60146	Ідентифікатор помилки для 11-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EAF7	60151	Ідентифікатор помилки для 11-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
EAF8	60152	Ідентифікатор помилки для 11-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
EAF9	60153	Ідентифікатор помилки для 11-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
EAFA	60154	Ідентифікатор помилки для 11-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EAFB	60155	Ідентифікатор помилки для 11-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
EAFC	60156	Ідентифікатор помилки для 11-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
EAFD	60157	Ідентифікатор помилки для 11-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
EB01	60161	Ідентифікатор помилки для 12-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EB02	60162	Ідентифікатор помилки для 12-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EB04	60164	Ідентифікатор помилки для 12-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
EB07	60167	Ідентифікатор помилки для 12-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
EB08	60168	Ідентифікатор помилки для 12-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
EB09	60169	Ідентифікатор помилки для 12-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
EB0A	60170	Ідентифікатор помилки для 12-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
EB0B	60171	Ідентифікатор помилки для 12-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EB0C	60172	Ідентифікатор помилки для 12-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
EB0D	60173	Ідентифікатор помилки для 12-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
EB0E	60174	Ідентифікатор помилки для 12-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
EB0F	60175	Ідентифікатор помилки для 12-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
EBF1	60401	Ідентифікатор помилки для 12-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EBF2	60402	Ідентифікатор помилки для 12-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EBF7	60407	Ідентифікатор помилки для 12-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
EBF8	60408	Ідентифікатор помилки для 12-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
EBF9	60409	Ідентифікатор помилки для 12-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
EBFA	60410	Ідентифікатор помилки для 12-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EBFB	60411	Ідентифікатор помилки для 12-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
EBFC	60412	Ідентифікатор помилки для 12-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
EBFD	60413	Ідентифікатор помилки для 12-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
EC01	60417	Ідентифікатор помилки для 13-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EC02	60418	Ідентифікатор помилки для 13-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EC04	60420	Ідентифікатор помилки для 13-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
EC07	60423	Ідентифікатор помилки для 13-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
EC08	60424	Ідентифікатор помилки для 13-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
EC09	60425	Ідентифікатор помилки для 13-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
EC0A	60426	Ідентифікатор помилки для 13-го правого модуля, 16#180A: сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
EC0B	60427	Ідентифікатор помилки для 13-го правого модуля, 16#180B: сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EC0C	60428	Ідентифікатор помилки для 13-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
EC0D	60429	Ідентифікатор помилки для 13-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
EC0E	60430	Ідентифікатор помилки для 13-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
EC0F	60431	Ідентифікатор помилки для 13-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
ECF1	60657	Ідентифікатор помилки для 13-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
ECF2	60658	Ідентифікатор помилки для 13-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
ECF7	60663	Ідентифікатор помилки для 13-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
ECF8	60664	Ідентифікатор помилки для 13-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
ECF9	60665	Ідентифікатор помилки для 13-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
ECFA	60666	Ідентифікатор помилки для 13-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
ECFB	60667	Ідентифікатор помилки для 13-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
ECFC	60668	Ідентифікатор помилки для 13-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
ECFD	60669	Ідентифікатор помилки для 13-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
ED01	60673	Ідентифікатор помилки для 14-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
ED02	60674	Ідентифікатор помилки для 14-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
ED04	60676	Ідентифікатор помилки для 14-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
ED07	60679	Ідентифікатор помилки для 14-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
ED08	60680	Ідентифікатор помилки для 14-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
ED09	60681	Ідентифікатор помилки для 14-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
ED0A	60682	Ідентифікатор помилки для 14-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
ED0B	60683	Ідентифікатор помилки для 14-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
ED0C	60684	Ідентифікатор помилки для 14-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
ED0D	60685	Ідентифікатор помилки для 14-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
ED0E	60686	Ідентифікатор помилки для 14-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
ED0F	60687	Ідентифікатор помилки для 14-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
EDF1	60913	Ідентифікатор помилки для 14-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EDF2	60914	Ідентифікатор помилки для 14-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EDF7	60919	Ідентифікатор помилки для 14-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
EDF8	60920	Ідентифікатор помилки для 14-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
EDF9	60921	Ідентифікатор помилки для 14-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
EDFA	60922	Ідентифікатор помилки для 14-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EDFB	60923	Ідентифікатор помилки для 14-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
EDFC	60924	Ідентифікатор помилки для 14-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
EDFD	60925	Ідентифікатор помилки для 14-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
EE01	60929	Ідентифікатор помилки для 15-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EE02	60930	Ідентифікатор помилки для 15-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EE04	60932	Ідентифікатор помилки для 15-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
EE07	60935	Ідентифікатор помилки для 15-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
EE08	60936	Ідентифікатор помилки для 15-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
EE09	60937	Ідентифікатор помилки для 15-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
EE0A	60938	Ідентифікатор помилки для 15-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
EE0B	60939	Ідентифікатор помилки для 15-го правого модуля, 16#180B: сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EE0C	60940	Ідентифікатор помилки для 15-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
EE0D	60941	Ідентифікатор помилки для 15-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
EE0E	60942	Ідентифікатор помилки для 15-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
EE0F	60943	Ідентифікатор помилки для 15-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
EEF1	61169	Ідентифікатор помилки для 15-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EEF2	61170	Ідентифікатор помилки для 15-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EEF7	61175	Ідентифікатор помилки для 15-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
EEF8	61176	Ідентифікатор помилки для 15-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
EEF9	61177	Ідентифікатор помилки для 15-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
EEFA	61178	Ідентифікатор помилки для 15-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EEFB	61179	Ідентифікатор помилки для 15-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
EEFC	61180	Ідентифікатор помилки для 15-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
EEFD	61181	Ідентифікатор помилки для 15-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
EF01	61185	Ідентифікатор помилки для 16-го правого модуля, 16#1801: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EF02	61186	Ідентифікатор помилки для 16-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EF04	61188	Ідентифікатор помилки для 16-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
EF07	61191	Ідентифікатор помилки для 16-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
EF08	61192	Ідентифікатор помилки для 16-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
EF09	61193	Ідентифікатор помилки для 16-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
EF0A	61194	Ідентифікатор помилки для 16-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
EF0B	61195	Ідентифікатор помилки для 16-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EF0C	61196	Ідентифікатор помилки для 16-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
EF0D	61197	Ідентифікатор помилки для 16-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
EF0E	61198	Ідентифікатор помилки для 16-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
EF0F	61199	Ідентифікатор помилки для 16-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
EFF1	61425	Ідентифікатор помилки для 16-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
EFF2	61426	Ідентифікатор помилки для 16-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
EFF7	61431	Ідентифікатор помилки для 16-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
EFF8	61432	Ідентифікатор помилки для 16-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
EFF9	61433	Ідентифікатор помилки для 16-го правого модуля, 16#18F9: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
EFFA	61434	Ідентифікатор помилки для 16-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
EFFB	61435	Ідентифікатор помилки для 16-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
EFFC	61436	Ідентифікатор помилки для 16-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
EFFD	61437	Ідентифікатор помилки для 16-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F001	61441	Ідентифікатор помилки для 17-го правого модуля, 16#1801: збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F002	61442	Ідентифікатор помилки для 17-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F004	61444	Ідентифікатор помилки для 17-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F007	61447	Ідентифікатор помилки для 17-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F008	61448	Ідентифікатор помилки для 17-го правого модуля, 16#1808: сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F009	61449	Ідентифікатор помилки для 17-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F00A	61450	Ідентифікатор помилки для 17-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F00B	61451	Ідентифікатор помилки для 17-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F00C	61452	Ідентифікатор помилки для 17-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F00D	61453	Ідентифікатор помилки для 17-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F00E	61454	Ідентифікатор помилки для 17-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F00F	61455	Ідентифікатор помилки для 17-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
F0F1	61681	Ідентифікатор помилки для 17-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F0F2	61682	Ідентифікатор помилки для 17-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F0F7	61687	Ідентифікатор помилки для 17-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F0F8	61688	Ідентифікатор помилки для 17-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F0F9	61689	Ідентифікатор помилки для 17-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F0FA	61690	Ідентифікатор помилки для 17-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F0FB	61691	Ідентифікатор помилки для 17-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F0FC	61692	Ідентифікатор помилки для 17-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F0FD	61693	Ідентифікатор помилки для 17-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F101	61697	Ідентифікатор помилки для 18-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F102	61698	Ідентифікатор помилки для 18-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F104	61700	Ідентифікатор помилки для 18-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F107	61703	Ідентифікатор помилки для 18-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F108	61704	Ідентифікатор помилки для 18-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F109	61705	Ідентифікатор помилки для 18-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F10A	61706	Ідентифікатор помилки для 18-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F10B	61707	Ідентифікатор помилки для 18-го правого модуля, 16#180B: сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F10C	61708	Ідентифікатор помилки для 18-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
F10D	61709	Ідентифікатор помилки для 18-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
F10E	61710	Ідентифікатор помилки для 18-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
F10F	61711	Ідентифікатор помилки для 18-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
F1F1	61937	Ідентифікатор помилки для 18-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F1F2	61938	Ідентифікатор помилки для 18-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F1F7	61943	Ідентифікатор помилки для 18-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F1F8	61944	Ідентифікатор помилки для 18-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F1F9	61945	Ідентифікатор помилки для 18-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F1FA	61946	Ідентифікатор помилки для 18-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F1FB	61947	Ідентифікатор помилки для 18-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F1FC	61948	Ідентифікатор помилки для 18-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F1FD	61949	Ідентифікатор помилки для 18-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F201	61953	Ідентифікатор помилки для 19-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F202	61954	Ідентифікатор помилки для 19-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F204	61956	Ідентифікатор помилки для 19-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F207	61959	Ідентифікатор помилки для 19-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F208	61960	Ідентифікатор помилки для 19-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F209	61961	Ідентифікатор помилки для 19-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
F20A	61962	Ідентифікатор помилки для 19-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F20B	61963	Ідентифікатор помилки для 19-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F20C	61964	Ідентифікатор помилки для 19-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
F20D	61965	Ідентифікатор помилки для 19-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F20E	61966	Ідентифікатор помилки для 19-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F20F	61967	Ідентифікатор помилки для 19-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
F2F1	62193	Ідентифікатор помилки для 19-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F2F2	62194	Ідентифікатор помилки для 19-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F2F7	62199	Ідентифікатор помилки для 19-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F2F8	62200	Ідентифікатор помилки для 19-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F2F9	62201	Ідентифікатор помилки для 19-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F2FA	62202	Ідентифікатор помилки для 19-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F2FB	62203	Ідентифікатор помилки для 19-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F2FC	62204	Ідентифікатор помилки для 19-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F2FD	62205	Ідентифікатор помилки для 19-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F301	62209	Ідентифікатор помилки для 20-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F302	62210	Ідентифікатор помилки для 20-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F304	62212	Ідентифікатор помилки для 20-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F307	62215	Ідентифікатор помилки для 20-го правого модуля, 16#1807: Помилка температури CJC	Поверніть модуль на завод для ремонту.
F308	62216	Ідентифікатор помилки для 20-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F309	62217	Ідентифікатор помилки для 20-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
F30A	62218	Ідентифікатор помилки для 20-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
F30B	62219	Ідентифікатор помилки для 20-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F30C	62220	Ідентифікатор помилки для 20-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
F30D	62221	Ідентифікатор помилки для 20-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F30E	62222	Ідентифікатор помилки для 20-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F30F	62223	Ідентифікатор помилки для 20-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
F3F1	62449	Ідентифікатор помилки для 20-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F3F2	62450	Ідентифікатор помилки для 20-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F3F7	62455	Ідентифікатор помилки для 20-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F3F8	62456	Ідентифікатор помилки для 20-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F3F9	62457	Ідентифікатор помилки для 20-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F3FA	62458	Ідентифікатор помилки для 20-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F3FB	62459	Ідентифікатор помилки для 20-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F3FC	62460	Ідентифікатор помилки для 20-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F3FD	62461	Ідентифікатор помилки для 20-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F401	62465	Ідентифікатор помилки для 21-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F402	62466	Ідентифікатор помилки для 21-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F404	62468	Ідентифікатор помилки для 21-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F407	62471	Ідентифікатор помилки для 21-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F408	62472	Ідентифікатор помилки для 21-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
F409	62473	Ідентифікатор помилки для 21-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F40A	62474	Ідентифікатор помилки для 21-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F40B	62475	Ідентифікатор помилки для 21-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F40C	62476	Ідентифікатор помилки для 21-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F40D	62477	Ідентифікатор помилки для 21-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F40E	62478	Ідентифікатор помилки для 21-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F40F	62479	Ідентифікатор помилки для 21-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
F4F1	62705	Ідентифікатор помилки для 21-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F4F2	62706	Ідентифікатор помилки для 21-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F4F7	62711	Ідентифікатор помилки для 21-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F4F8	62712	Ідентифікатор помилки для 21-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F4F9	62713	Ідентифікатор помилки для 21-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F4FA	62714	Ідентифікатор помилки для 21-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F4FB	62715	Ідентифікатор помилки для 21-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F4FC	62716	Ідентифікатор помилки для 21-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F4FD	62717	Ідентифікатор помилки для 21-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F501	62721	Ідентифікатор помилки для 22-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F502	62722	Ідентифікатор помилки для 22-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F504	62724	Ідентифікатор помилки для 22-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F507	62727	Ідентифікатор помилки для 22-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F508	62728	Ідентифікатор помилки для 22-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F509	62729	Ідентифікатор помилки для 22-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
F50A	62730	Ідентифікатор помилки для 22-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
F50B	62731	Ідентифікатор помилки для 22-го правого модуля, 16#180B: сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F50C	62732	Ідентифікатор помилки для 22-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F50D	62733	Ідентифікатор помилки для 22-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
F50E	62734	Ідентифікатор помилки для 22-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F50F	62735	Ідентифікатор помилки для 22-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
F5F1	62961	Ідентифікатор помилки для 22-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F5F2	62962	Ідентифікатор помилки для 22-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F5F7	62967	Ідентифікатор помилки для 22-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F5F8	62968	Ідентифікатор помилки для 22-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F5F9	62969	Ідентифікатор помилки для 22-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F5FA	62970	Ідентифікатор помилки для 22-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F5FB	62971	Ідентифікатор помилки для 22-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F5FC	62972	Ідентифікатор помилки для 22-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F5FD	62973	Ідентифікатор помилки для 22-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F601	62977	Ідентифікатор помилки для 23-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F602	62978	Ідентифікатор помилки для 23-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F604	62980	Ідентифікатор помилки для 23-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F607	62983	Ідентифікатор помилки для 23-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F608	62984	Ідентифікатор помилки для 23-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F609	62985	Ідентифікатор помилки для 23-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F60A	62986	Ідентифікатор помилки для 23-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F60B	62987	Ідентифікатор помилки для 23-го правого модуля, 16#180B: сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F60C	62988	Ідентифікатор помилки для 23-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F60D	62989	Ідентифікатор помилки для 23-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F60E	62990	Ідентифікатор помилки для 23-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F60F	62991	Ідентифікатор помилки для 23-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
F6F1	63217	Ідентифікатор помилки для 23-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F6F2	63218	Ідентифікатор помилки для 23-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F6F7	63223	Ідентифікатор помилки для 23-го правого модуля, 16#18F7: помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F6F8	63224	Ідентифікатор помилки для 23-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F6F9	63225	Ідентифікатор помилки для 23-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F6FA	63226	Ідентифікатор помилки для 23-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F6FB	63227	Ідентифікатор помилки для 23-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F6FC	63228	Ідентифікатор помилки для 23-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F6FD	63229	Ідентифікатор помилки для 23-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F701	63233	Ідентифікатор помилки для 24-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F702	63234	Ідентифікатор помилки для 24-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F704	63236	Ідентифікатор помилки для 24-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F707	63239	Ідентифікатор помилки для 24-го правого модуля, 16#1807: Помилка температури CJC	Поверніть модуль на завод для ремонту.
F708	63240	Ідентифікатор помилки для 24-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
F709	63241	Ідентифікатор помилки для 24-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F70A	63242	Ідентифікатор помилки для 24-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
F70B	63243	Ідентифікатор помилки для 24-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F70C	63244	Ідентифікатор помилки для 24-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
F70D	63245	Ідентифікатор помилки для 24-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
F70E	63246	Ідентифікатор помилки для 24-го правого модуля, 16#180E: сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
F70F	63247	Ідентифікатор помилки для 24-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 8.
F7F1	63473	Ідентифікатор помилки для 24-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F7F2	63474	Ідентифікатор помилки для 24-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F7F7	63479	Ідентифікатор помилки для 24-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F7F8	63480	Ідентифікатор помилки для 24-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F7F9	63481	Ідентифікатор помилки для 24-го правого модуля, 16#18F9: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F7FA	63482	Ідентифікатор помилки для 24-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F7FB	63483	Ідентифікатор помилки для 24-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F7FC	63484	Ідентифікатор помилки для 24-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F7FD	63485	Ідентифікатор помилки для 24-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F801	63489	Ідентифікатор помилки для 25-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F802	63490	Ідентифікатор помилки для 25-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F804	63492	Ідентифікатор помилки для 25-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F807	63495	Ідентифікатор помилки для 25-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F808	63496	Ідентифікатор помилки для 25-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
F809	63497	Ідентифікатор помилки для 25-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F80A	63498	Ідентифікатор помилки для 25-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F80B	63499	Ідентифікатор помилки для 25-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F80C	63500	Ідентифікатор помилки для 25-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F80D	63501	Ідентифікатор помилки для 25-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F80E	63502	Ідентифікатор помилки для 25-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
F80F	63503	Ідентифікатор помилки для 25-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
F8F1	63729	Ідентифікатор помилки для 25-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F8F2	63730	Ідентифікатор помилки для 25-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F8F7	63735	Ідентифікатор помилки для 25-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F8F8	63736	Ідентифікатор помилки для 25-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F8F9	63737	Ідентифікатор помилки для 25-го правого модуля, 16#18F9: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F8FA	63738	Ідентифікатор помилки для 25-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F8FB	63739	Ідентифікатор помилки для 25-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F8FC	63740	Ідентифікатор помилки для 25-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F8FD	63741	Ідентифікатор помилки для 25-го правого модуля, 16#18FD: CH2 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
F901	63745	Ідентифікатор помилки для 26-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F902	63746	Ідентифікатор помилки для 26-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F904	63748	Ідентифікатор помилки для 26-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
F907	63751	Ідентифікатор помилки для 26-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
F908	63752	Ідентифікатор помилки для 26-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
F909	63753	Ідентифікатор помилки для 26-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
F90A	63754	Ідентифікатор помилки для 26-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
F90B	63755	Ідентифікатор помилки для 26-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F90C	63756	Ідентифікатор помилки для 26-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
F90D	63757	Ідентифікатор помилки для 26-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
F90E	63758	Ідентифікатор помилки для 26-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
F90F	63759	Ідентифікатор помилки для 26-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
F9F1	63985	Ідентифікатор помилки для 26-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
F9F2	63986	Ідентифікатор помилки для 26-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
F9F7	63991	Ідентифікатор помилки для 26-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
F9F8	63992	Ідентифікатор помилки для 26-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
F9F9	63993	Ідентифікатор помилки для 26-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
F9FA	63994	Ідентифікатор помилки для 26-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
F9FB	63995	Ідентифікатор помилки для 26-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
F9FC	63996	Ідентифікатор помилки для 26-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
F9FD	63997	Ідентифікатор помилки для 26-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FA01	64001	Ідентифікатор помилки для 27-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FA02	64002	Ідентифікатор помилки для 27-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FA04	64004	Ідентифікатор помилки для 27-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FA07	64007	Ідентифікатор помилки для 27-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FA08	64008	Ідентифікатор помилки для 27-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
FA09	64009	Ідентифікатор помилки для 27-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
FA0A	64010	Ідентифікатор помилки для 27-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
FA0B	64011	Ідентифікатор помилки для 27-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FA0C	64012	Ідентифікатор помилки для 27-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
FA0D	64013	Ідентифікатор помилки для 27-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
FA0E	64014	Ідентифікатор помилки для 27-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
FA0F	64015	Ідентифікатор помилки для 27-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
FAF1	64241	Ідентифікатор помилки для 27-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FAF2	64242	Ідентифікатор помилки для 27-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FAF7	64247	Ідентифікатор помилки для 27-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FAF8	64248	Ідентифікатор помилки для 27-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FAF9	64249	Ідентифікатор помилки для 27-го правого модуля, 16#18F9: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FAFA	64250	Ідентифікатор помилки для 27-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FAFB	64251	Ідентифікатор помилки для 27-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FAFC	64252	Ідентифікатор помилки для 27-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FAFD	64253	Ідентифікатор помилки для 27-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FB01	64257	Ідентифікатор помилки для 28-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FB02	64258	Ідентифікатор помилки для 28-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FB04	64260	Ідентифікатор помилки для 28-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FB07	64263	Ідентифікатор помилки для 28-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FB08	64264	Ідентифікатор помилки для 28-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
FB09	64265	Ідентифікатор помилки для 28-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
FB0A	64266	Ідентифікатор помилки для 28-го правого модуля, 16#180A: сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
FB0B	64267	Ідентифікатор помилки для 28-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FB0C	64268	Ідентифікатор помилки для 28-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5
FB0D	64269	Ідентифікатор помилки для 28-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6
FB0E	64270	Ідентифікатор помилки для 28-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7
FB0F	64271	Ідентифікатор помилки для 28-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8
FBF1	64497	Ідентифікатор помилки для 28-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FBF2	64498	Ідентифікатор помилки для 28-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FBF7	64503	Ідентифікатор помилки для 28-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FBF8	64504	Ідентифікатор помилки для 28-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FBF9	64505	Ідентифікатор помилки для 28-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FBFA	64506	Ідентифікатор помилки для 28-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FBFB	64507	Ідентифікатор помилки для 28-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FBFC	64508	Ідентифікатор помилки для 28-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FBFD	64509	Ідентифікатор помилки для 28-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FC01	64513	Ідентифікатор помилки для 29-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FC02	64514	Ідентифікатор помилки для 29-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FC04	64516	Ідентифікатор помилки для 29-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FC07	64519	Ідентифікатор помилки для 29-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FC08	64520	Ідентифікатор помилки для 29-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
FC09	64521	Ідентифікатор помилки для 29-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
FC0A	64522	Ідентифікатор помилки для 29-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 3.
FC0B	64523	Ідентифікатор помилки для 29-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FC0C	64524	Ідентифікатор помилки для 29-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
FC0D	64525	Ідентифікатор помилки для 29-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
FC0E	64526	Ідентифікатор помилки для 29-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
FC0F	64527	Ідентифікатор помилки для 29-го правого модуля, 16#180F: сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
FCF1	64753	Ідентифікатор помилки для 29-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FCF2	64754	Ідентифікатор помилки для 29-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FCF7	64759	Ідентифікатор помилки для 29-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FCF8	64760	Ідентифікатор помилки для 29-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FCF9	64761	Ідентифікатор помилки для 29-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FCFA	64762	Ідентифікатор помилки для 29-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FCFB	64763	Ідентифікатор помилки для 29-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FCFC	64764	Ідентифікатор помилки для 29-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FCFD	64765	Ідентифікатор помилки для 29-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FD01	64769	Ідентифікатор помилки для 30-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FD02	64770	Ідентифікатор помилки для 30-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FD04	64772	Ідентифікатор помилки для 30-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FD07	64775	Ідентифікатор помилки для 30-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FD08	64776	Ідентифікатор помилки для 30-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
FD09	64777	Ідентифікатор помилки для 30-го правого модуля, 16#1809: Сигнал (температура), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 2.
FD0A	64778	Ідентифікатор помилки для 30-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
FD0B	64779	Ідентифікатор помилки для 30-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FD0C	64780	Ідентифікатор помилки для 30-го правого модуля, 16#180C: Сигнал (температура), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
FD0D	64781	Ідентифікатор помилки для 30-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
FD0E	64782	Ідентифікатор помилки для 30-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 7.
FD0F	64783	Ідентифікатор помилки для 30-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
FDF1	65009	Ідентифікатор помилки для 30-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FDF2	65010	Ідентифікатор помилки для 30-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FDF7	65015	Ідентифікатор помилки для 30-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FDF8	65016	Ідентифікатор помилки для 30-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FDF9	65017	Ідентифікатор помилки для 30-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FDFA	65018	Ідентифікатор помилки для 30-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FDFB	65019	Ідентифікатор помилки для 30-го правого модуля, 16#18FB: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN є неправильною.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FDFC	65020	Ідентифікатор помилки для 30-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FDFD	65021	Ідентифікатор помилки для 30-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FE01	65025	Ідентифікатор помилки для 31-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FE02	65026	Ідентифікатор помилки для 31-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FE04	65028	Ідентифікатор помилки для 31-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FE07	65031	Ідентифікатор помилки для 31-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FE08	65032	Ідентифікатор помилки для 31-го правого модуля, 16#1808: Сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 1.
FE09	65033	Ідентифікатор помилки для 31-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
FE0A	65034	Ідентифікатор помилки для 31-го правого модуля, 16#180A: Сигнал (температури), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
FE0B	65035	Ідентифікатор помилки для 31-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FE0C	65036	Ідентифікатор помилки для 31-го правого модуля, 16#180C: Сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 5.
FE0D	65037	Ідентифікатор помилки для 31-го правого модуля, 16#180D: сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 6.
FE0E	65038	Ідентифікатор помилки для 31-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
FE0F	65039	Ідентифікатор помилки для 31-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
FEF1	65265	Ідентифікатор помилки для 31-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FEF2	65266	Ідентифікатор помилки для 31-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FEF7	65271	Ідентифікатор помилки для 31-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FEF8	65272	Ідентифікатор помилки для 31-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FEF9	65273	Ідентифікатор помилки для 31-го правого модуля, 16#18F9: Вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FEFA	65274	Ідентифікатор помилки для 31-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FEFB	65275	Ідентифікатор помилки для 31-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FEFC	65276	Ідентифікатор помилки для 31-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FEFD	65277	Ідентифікатор помилки для 31-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.
FF01	65281	Ідентифікатор помилки для 32-го правого модуля, 16#1801: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FF02	65282	Ідентифікатор помилки для 32-го правого модуля, 16#1802: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FF04	65284	Ідентифікатор помилки для 32-го правого модуля, 16#1804: внутрішня помилка. Заводське калібрування є ненормальним.	Зверніться на завод.
FF07	65287	Ідентифікатор помилки для 32-го правого модуля, 16#1807: помилка температури CJC	Поверніть модуль на завод для ремонту.
FF08	65288	Ідентифікатор помилки для 32-го правого модуля, 16#1808: сигнал (температури), отриманий каналом 1, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 1.
FF09	65289	Ідентифікатор помилки для 32-го правого модуля, 16#1809: Сигнал (температури), отриманий каналом 2, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 2.
FF0A	65290	Ідентифікатор помилки для 32-го правого модуля, 16#180A: Сигнал (температура), отриманий каналом 3, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 3.
FF0B	65291	Ідентифікатор помилки для 32-го правого модуля, 16#180B: Сигнал (температури), отриманий каналом 4, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 4.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FF0C	65292	Ідентифікатор помилки для 32-го правого модуля, 16#180C: сигнал (температури), отриманий каналом 5, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 5.
FF0D	65293	Ідентифікатор помилки для 32-го правого модуля, 16#180D: Сигнал (температури), отриманий каналом 6, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 6.
FF0E	65294	Ідентифікатор помилки для 32-го правого модуля, 16#180E: Сигнал (температури), отриманий каналом 7, перевищує діапазон вхідних даних, які може отримати обладнання.	Перевірте сигнал, отриманий каналом 7.
FF0F	65295	Ідентифікатор помилки для 32-го правого модуля, 16#180F: Сигнал (температури), отриманий каналом 8, перевищує діапазон вхідних даних, які може отримати апаратне забезпечення.	Перевірте сигнал, отриманий каналом 8.
FFF1	65521	Ідентифікатор помилки для 32-го правого модуля, 16#18F1: Збій зовнішнього джерела живлення	Переконайтеся, що зовнішнє джерело живлення 24 В постійного струму для модуля працює нормально.
FFF2	65522	Ідентифікатор помилки для 32-го правого модуля, 16#18F2: апаратна помилка модуля	Поверніть модуль на завод для ремонту.
FFF7	65527	Ідентифікатор помилки для 32-го правого модуля, 16#18F7: Помилка на платі драйвера	Поверніть модуль на завод для ремонту.
FFF8	65528	Ідентифікатор помилки для 32-го правого модуля, 16#18F8: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 1, і його підключення.
FFF9	65529	Ідентифікатор помилки для 32-го правого модуля, 16#18F9: вага, виміряна CH1, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 1.
FFFA	65530	Ідентифікатор помилки для 32-го правого модуля, 16#18FA: CH1 налаштовано неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 1.

Код системної помилки		Пояснення	Виправлення
Шістнадцятковий	Десятковий		
FFFB	65531	Ідентифікатор помилки для 32-го правого модуля, 16#18FB: вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти, або напруга SEN неправильна.	Перевірте сигнал, отриманий каналом 2, і його підключення.
FFFC	65532	Ідентифікатор помилки для 32-го правого модуля, 16#18FC: Вага, виміряна CH2, перевищує максимальну вагу, яку можна виміряти.	Перевірте параметри відповідних значень ваги для каналу 2.
FFFD	65533	Ідентифікатор помилки для 32-го правого модуля, 16#18FD: канал CH2 налаштований неправильно.	Перевірте налаштоване значення ваги та кроки налаштування для каналу 2.

Додаток A Зв'язок Modbus

Зміст

A.1 Message Format in ASCII Mode A-2
A.2 Message Format in RTU Mode A-5
A.3 Modbus Function Codes Supported A-7
A.4 Modbus Exception Response Code Supported A-7
A.5 Introduction to Modbus Function Codes A-8
A.6 Table of Registers and Corresponding Modbus addresses A-15

A.1 Формат повідомлення в режимі ASCII

- Структура даних зв'язку

Назва поля	компоненти	Пояснення
Початковий символ	STX	Початковий символ «:», відповідний код ASCII: 16#3A
Адреса зв'язку	ADR 1	Адреса зв'язку складається з двох кодів ASCII.
	ADR 0	
Код функції	CMD 1	Код функції складається з двох кодів ASCII.
	CMD 0	
Дані	ДАНІ (0)	Вміст даних складається з 2n кодів ASCII, n ≤ 205.
	ДАНІ (1)	
	…….	
	ДАНІ (n-1)	
Перевірка LRC	LRC CHK 1	Перевірка LRC складається з двох кодів ASCII.
	LRC CHK 0	
Кінцевий символ	END1	Кінцевий символ складається з двох кодів ASCII. END1 = CR (16#0D), END0 = LF (16#0A)
	END0	

Відповідне співвідношення між шістнадцятковим символом і кодом ASCII:

Шістнадцятковий символ	«0»	«1»	«2»	«3»	«4»	«5»	«6»	«7»
код ASCII	16#30	16#31	16#32	16#33	16#34	16#35	16#36	16#37
Шістнадцятковий символ	«8»	«9»	«A»	«B»	«C»	«D»	«E»	«F»
код ASCII	16#38	16#39	16#41	16#42	16#43	16#44	16#45	16#46

- ADR (комунікаційна адреса)

  Діапазон допустимих адрес зв’язку: 0 ~ 254.

  Адреса зв’язку: 0 означає, що широкомовне повідомлення надсилається всім підлеглим, а підлеглі, які отримали повідомлення, не відповідатимуть. Якщо адреса зв’язку не дорівнює 0, підлеглі відповідатимуть головному після нормального отримання повідомлення. Наприклад, ASCII-коди для адреси зв’язку 16 позначені нижче.

  Десяткове число 16 дорівнює шістнадцятковому 10. (ADR 1, ADR 0) = '10', '1'=31H, '0' = 30H

- Код функції та дані

  Формат даних визначається кодами функцій. Наприклад, для читання двох неперервних адресних даних із шістнадцятковим 16#0000 як початковою адресою в контролері руху. Адреса зв’язку контролера руху дорівнює 1, 16#0000 – це адреса Modbus %MW0 у контролері.

  Пояснення даних показано нижче:

  ПК → контролер руху

  3A 30 31 30 33 30 30 30 30 30 30 30 32 46 41 0D 0A

  Контролер руху → ПК

  3A 30 31 30 33 30 34 30 30 30 31 30 30 30 32 46 35 0D 0A

■ Повідомлення запиту:

Назва поля	Польовий характер	Код ASCII, що відповідає символу поля
Початковий символ	":"	3A
Адреса зв'язку: 01	«0»	30
	«1»	31
Код функції: 03	«0»	30
	«3»	33
Стартова адреса: 16#0000	«0»	30
	«0»	30
	«0»	30
	«0»	30
Кількість даних (підраховується словами): 2	«0»	30
	«0»	30
	«0»	30
	«2»	32
Контрольний код LRC: 16#FA	«F»	46
	«A»	41
Кінцевий символ 1	CR	0D
Кінцевий символ 0	LF	0A

■ Відповідне повідомлення:

Назва поля	Польовий характер	Код ASCII, що відповідає символу поля
Початковий символ	":"	3A
Адреса зв'язку: 01	«0»	30
	«1»	31
Код функції: 03	«0»	30
	«3»	33
Номер даних (рахується по байтам):	«0»	30
	«4»	34
Прочитайте вміст адреси 16#1000	«0»	30
	«0»	30
	«0»	30
	«1»	31
Прочитати вміст адреси 16#1001	«0»	30
	«0»	30
	«0»	30
	«2»	32

Назва поля	Польовий характер	Код ASCII, що відповідає символу поля
Контрольний код LRC: 16#F5	«F»	46
	«5»	35
Кінцевий символ 1	CR	0D
Кінцевий символ 0	LF	0A

- Чек LRC (контрольна сума)

  Контрольний код LRC – це значення, яке спочатку отримує зворотні значення кожного біта результату операції додавання даних з ідентифікатора зв'язку до останнього вмісту даних (Hex.), а потім додає 1 до остаточного зворотного значення.

  Наприклад, значення коду перевірки LRC: 16#FA. Спосіб розрахунку значення контрольного коду LRC: 16#01+ 16#03 + 16#00 + 16#00 + 16#00 + 16#02 = 16#06, результат 16#FA отримуємо шляхом отримання зворотних значень кожен біт 16#06, а потім додавання 1 до кінцевого зворотного значення.

A

Назва поля	Польовий характер	Код ASCII, що відповідає символу поля
Початковий символ	":"	3A
Адреса зв'язку: 01	«0»	30
	«1»	31
Код функції: 03	«0»	30
	«3»	33
Початкова адреса даних: 16#0000	«0»	30
	«0»	30
	«0»	30
	«0»	30
Номер даних (Порахувати словом):2	«0»	30
	«0»	30
	«0»	30
	«2»	32
Контрольний код LRC: 16#FA	«F»	46
	«A»	41
Кінцевий символ 1: CR	CR	0D
Кінцевий символ 0: LF	LF	0A

A.2 Формат повідомлення в режимі RTU

■ Структура даних зв'язку

старт	Немає вхідних даних більше 10 мс
Адреса зв'язку	Адреса веденого пристрою: 8-бітна двійкова адреса
Код функції	Код функції: 8-бітна двійкова адреса
Дані (n-1)	Вміст даних n × 8 біт двійкових даних, n<=202
…….	
Дані 0	
Молодший байт перевірки CRC	Контрольна сума CRC
Старший байт перевірки CRC	
Кінець	Контрольна сума CRC складається з двох 8-бітних двійкових даних

■ Адреса зв'язку

Діапазон дійсної адреси зв’язку становить 0 ~ 254. Адреса зв’язку 0 вказує на трансляцію повідомлення всім підлеглим, а підлеглі, які отримали широкомовне повідомлення, не відповідають. Якщо комунікаційна адреса не дорівнює 0, підлеглі відповідатимуть головному як зазвичай. Наприклад, для зв’язку з підлеглим із комунікаційною адресою 16 адреса підлеглого встановлюється як 16#10, оскільки десяткове 16 дорівнює шістнадцятковому 10.

■ Код функції та дані

Формат даних визначається кодами функцій.

Наприклад, щоб прочитати дані двох послідовних адрес із 16#0000 як початкову адресу в контролері руху, адреса дорівнює 1, 16#0000 є адресою Modbus %MW0 у контролері.

Дані в кабелі зв’язку та пояснення до них наведені нижче:

ПК→контролер руху: “01 03 00 00 00 02 C4 0B”

Контролер руху→ПК: “01 03 04 00 01 02 00 2A 32”

➢ Повідомлення запиту:

Назва поля	характер
старт	Немає вхідних даних більше 10 мс
Адреса зв'язку	01
Код функції	03
Старший байт адреси Modbus	00
Молодший байт адреси Modbus	00
Прочитати старший байт номера даних	00
Прочитати молодший байт номера даних	02
Молодший байт контрольної суми CRC	C4
Старший байт контрольної суми CRC	0B
Кінець	Немає вхідних даних більше 10 мс

- Відповідне повідомлення:

Назва поля	характер
старт	Немає вхідних даних більше 10 мс
Адреса зв'язку	01
Код функції	03
Кількість прочитаних даних （ Рахується в байтах ）	04
Читання старшого байта вмісту даних	00
Читання молодшого байта вмісту даних	01
Читання старшого байта вмісту даних	00
Читання молодшого байта вмісту даних	02
Молодший байт контрольної суми CRC	2A
Старший байт контрольної суми CRC	32
Кінець	Немає вхідних даних більше 10 мс

A

■ Перевірка CRC (контрольна сума)

Перевірка CRC починається з «Адреси зв’язку» до останнього «Вмісту даних». Метод розрахунку наведено нижче.

Крок 1. Завантажте 16-розрядний шістнадцятковий регістр (CRC-реєстр) зі значенням вмісту FFFF.

Крок 2: Виконайте операцію XOR між 8-бітовими даними першого байту в команді та 8-бітовими даними молодшого байта в регістрі CRC, а потім збережіть результат операції в регістрі CRC.

Крок 3: перемістіть значення вмісту регістра CRC на один біт праворуч і заповніть 0 у старшому біті.

Крок 4: Перевірте значення молодшого біта в регістрі CRC. Якщо значення дорівнює 0, повторіть дію кроку 3; якщо 1, виконати операцію XOR між вмістом у регістрі CRC і шістнадцятковим. A001, а потім зберегти результат у регістрі CRC.

Крок 5: повторюйте кроки 3 і 4, доки вміст у регістрі CRC не переміститься на 8 біт вправо. На цьому обробка першого байта командного повідомлення закінчена.

Крок 6: повторюйте дії з кроку 2 по крок 5 для наступного байта в повідомленні команди, доки не завершиться обробка всіх байтів. Останнім вмістом у реєстрі CRC є контрольне значення CRC. Коли передається контрольне значення CRC у повідомленні команди, старший і молодший байти в обчисленому контрольному значенні CRC повинні обмінюватися один з одним, тобто молодший байт передається першим.

Приклад розрахунку контрольного значення CRC мовою C

```
unsigned char* data ← // Покажчик вмісту командного повідомлення
unsigned char length ← // Довжина вмісту командного повідомлення
unsigned int crc_chk （ дані без знаку*, довжина символу без знаку ）
{
int j;
unsigned int reg_crc=0Xffff;
while （ довжина-- ）
{
reg_crc ^= *дані++;
for （ j=0;j<8;j++ ）
{
If （ reg_crc & 0x01 ） reg_crc= （ reg_crc>>1 ） ^ 0Xa001; /* LSB （ b0 ） =1 */ else
reg_crc=reg_crc >>1;
}
```

```
}
повернення reg_crc;          // остаточне значення, яке повертається до реєстру CRC
}
```

A.3 Підтримуються коди функцій Modbus

- Функціональні коди, які підтримуються контролером руху, перераховані в наступній таблиці, якщо порт COM2 має модуль керування рухом.

Код функції	Пояснення	Наявний реєстр
16#01	Читання вихідних розрядних значень регістра; за один раз можна прочитати щонайбільше 256 біт	%QX
16#02	Читання значень бітового регістра; за один раз можна прочитати щонайбільше 256 біт	%IX,%QX
16#03	Читання одного або кількох слів у регістрі; за один раз можна прочитати щонайбільше 100 слів.	%MW,%QW,%IW
16#05	Записати одне значення регістра одного біта.	%QX
16#06	Запишіть одне значення регістра слова.	%MW,%QW
16#0F	Записати багатобітове значення регістра; за раз можна записати щонайбільше 256 біт.	%QX
16#10	Записати значення регістра кількох слів; за раз можна записати щонайбільше 100 слів.	%MW,%QW

A.4 Код відповіді Modbus Exception Підтримується

- Коди відповіді винятків, які підтримує контролер руху, наведено в таблиці нижче.

Код відповіді винятку	Пояснення
16#01	Незаконні коди команд: коди команд у командному повідомленні, яке отримує ПЛК, недійсні.
16#02	Незаконна адреса реєстрації: адреса в отриманому повідомленні команди недійсна.
16#03	Неправильне значення регістру: дані в командному повідомленні, отриманому ПЛК, недійсні.
16#07	◆ Помилка перевірки суми ✔ Перевірте, чи правильна контрольна сума. ◆ Незаконне командне повідомлення ✔ Занадто коротке командне повідомлення ✔ Довжина командного повідомлення перевищує допустимий діапазон.

A.5 Вступ до функціональних кодів Modbus

- Функціональний код 03 зчитує значення регістра одного або кількох слів
  - Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Прочитайте початкову адресу регістрів слів у контролері руху	Старший байт
Байт3		Молодший байт
Байт4	Прочитайте кількість адрес регістрів слів у контролері руху (Порахувати словом)	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

  - Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Прочитайте кількість адрес регістрів слів у контролері руху (Рахунок по байтам)	Один байт
Байт3	Вміст адреси регістру слів у контролері руху	Старший байт
Байт4		Молодший байт
…	Вміст адреси регістру слів у контролері руху	Старший байт
…		Молодший байт
Байт n	Вміст адреси регістру слів у контролері руху	Старший байт
Байт n+1		Молодший байт
Байт n+2	Молодший байт контрольної суми CRC	Молодший байт
Байт n+3	Старший байт контрольної суми CRC	Старший байт

  - Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ код функції	Один байт
Байт2	Код відповіді винятку	Один байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

■ **приклад**

Щоб прочитати вміст адреси 16#0000 і 16#0001 у контролері руху за допомогою коду функції 03.

16#0000 і 16#0001 є адресами Modbus для %MW0 і %MW1 у контролері руху відповідно.

Припустимо, що значення %MW0 дорівнює 16#0001, а %MW1 дорівнює 16#0002: Повідомлення запиту: 01 03 00 00 00 02 C4 0B

Відповідь: 01 03 04 00 01 00 02 2A 32

- Код функції 06 записує одне значення регістра слова
  - ■ Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Адреса регістру контролера, куди потрібно записати значення	Старший байт
Байт3		Молодший байт
Байт4	Написане значення	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Адреса регістру слів контролера, куди потрібно записати значення	Старший байт
Байт3		Молодший байт
Байт4	Написане значення	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ код функції	Один байт
Байт2	Код відповіді винятку	Один байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

■ **приклад**

Напишіть 16#0100 на адресу 16#0000 у контролері руху за допомогою коду функції 06.

Повідомлення запиту: 01 06 00 00 01 00 88 5A

Відповідь: 01 06 00 00 01 00 88 5A

- Код функції 16#10 записує кілька значень регістра слів
  - Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса регістра слова контролера, куди слід записати значення	Старший байт
Байт3		Молодший байт
Байт4	Кількість адрес регістрів слів контролера, куди потрібно записати значення. (Порахувати по слову)	Старший байт
Байт5		Молодший байт
Байт6	Кількість адрес регістрів слів контролера, куди потрібно записати значення. (Рахунок по байтам)	Один байт
Байт7	Значення адреси, записане в регістр слів контролера	Старший байт
Байт8		Молодший байт
…	Значення адреси, записане в регістр слів контролера	Старший байт
…		Молодший байт
Байт n	Значення адреси, записане в регістр слів контролера	Старший байт
Байт n+1		Молодший байт
Байт n+2	Молодший байт контрольної суми CRC	Молодший байт
Байт n+3	Старший байт контрольної суми CRC	Старший байт

A

  - Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса регістра слова контролера, куди слід записати значення	Старший байт
Байт3		Молодший байт
Байт4	Кількість регістрів слів контролера, куди потрібно записати значення. (Порахувати словом)	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

  - Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ код функції	Один байт
Байт2	Код відповіді винятку	Один байт

Байт NO.	Ім'я	Байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

■ **приклад**

Запишіть 16#0100 і 16#0200 на адреси 16#0000 і 16#0001 у контролер відповідно за допомогою коду функції 16#10. 16#0000 і 16#0001 є адресами Modbus %MW0 і %MW1 у контролері відповідно.

Повідомлення запиту: 01 10 00 00 00 02 04 01 00 02 00 F3 33

Відповідь: 01 10 00 00 00 02 41 C8

● Функціональний код 16#01 зчитує кілька вихідних значень бітового регістра

■ Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса бітових регістрів контролера для читання	Старший байт
Байт3		Молодший байт
Байт4	Кількість бітових регістрів контролера для читання	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Зчитування кількості байтів бітових регістрів.	Один байт
Байт3	Прочитайте значення стану бітового регістра.	Один байт
…	Прочитайте значення стану бітового регістра.	Один байт
Байт n	Прочитайте значення стану бітового регістра.	Один байт
Байт n+1	Молодший байт контрольної суми CRC	Молодший байт
Байт n+2	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ код функції	Один байт
Байт2	Відповідне повідомлення про виняток	Один байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

Примітка:

Значення Байта 2 у повідомленні відповіді визначається значеннями Байта 4 і Байта 5 у повідомленні запиту. Наприклад, кількість прочитаних бітових регістрів у повідомленні запиту дорівнює A. Ділення A на 8 дає B. Якщо частка є цілим числом, кількість байтів бітових регістрів у повідомленні відповіді дорівнює B. В іншому випадку кількість байтів буде B + 1.
Подробиці див. у прикладі нижче.

■ **приклад**

Прочитайте значення стану %QX2.0~%QX3.4 у контролері руху за допомогою коду функції 01. Адреса %QX2.0 — 16#A010. Припустимо, що значення %QX2.0~%QX2.7 дорівнює 1000 0001 і %QX3.0~%QX3.4 дорівнює 1 0001.

Повідомлення запиту: 01 01 A0 10 00 0D DE 0A
Відповідь: 01 01 02 81 11 19 A0

● Код функції 16#02 зчитує значення кількох бітових регістрів

■ Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса бітових регістрів контролера, де читати стан	Старший байт
Байт3		Молодший байт
Байт4	Прочитайте кількість бітових регістрів.	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Зчитування кількості байтів бітових регістрів.	Один байт
Байт3	Прочитайте значення стану бітового регістра.	Один байт
…	Прочитайте значення стану бітового регістра.	Один байт
Байт n	Прочитайте значення стану бітового регістра.	Один байт
Байт n+1	Молодший байт контрольної суми CRC	Молодший байт
Байт n+2	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ Код функції	Один байт
Байт2	Код відповіді винятку	Один байт

Байт NO.	Ім'я	Байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

Примітка:

Значення Байта 2 у повідомленні відповіді визначається значеннями Байта 4 і Байта 5 у повідомленні запиту. Наприклад, кількість прочитаних бітових регістрів у повідомленні запиту дорівнює A. Ділення A на 8 дає B. Якщо частка є цілим числом, кількість байтів бітових регістрів у повідомленні відповіді дорівнює B. В іншому випадку кількість байтів буде бути B+ 1.

Подробиці див. у прикладі нижче.

■ **приклад**

Прочитайте значення стану %QX2.0~%QX3.4 у контролері руху за допомогою коду функції 02. Адреса %QX2.0 — 16#A010. Припустимо, %QX2.0~%QX2.7=1000 0001, %QX3.0~%QX3.4=1 0001.

Повідомлення запиту: 01 02 A0 10 00 0D 9A 0A

Відповідь: 01 02 02 81 11 19 E4

● Функціональний код 16#05 записує одне бітове значення регістра

■ Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Адреса Modbus бітового регістра	Старший байт
Байт3		Молодший байт
Байт4	Значення, записане в бітовому регістрі	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Адреса Modbus бітового регістра	Старший байт
Байт3		Молодший байт
Байт4	Значення, записане в бітовому регістрі	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ Код функції	Один байт
Байт2	Код відповіді винятку	Один байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

Примітка: Записане значення 16#0000 для бітового регістра в повідомленні запиту або відповіді вказує, що значення FALSE записане в бітовий регістр; записане значення 16#FF00 для бітового регістра вказує, що значення TRUE записане в бітовий регістр.

A

■ **приклад**

Значення %QX0.0 у контролері руху встановлено на TRUE, а адреса %QX0.0 встановлена на 16#A000 через код функції 05.

Повідомлення запиту: 01 05 A0 00 FF 00 AE 3A

Повідомлення відповіді: 01 05 A0 00 FF 00 AE 3A

● Код функції 16#0F записує кілька значень бітового регістра

■ Структура даних повідомлення запиту:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса бітового регістра, куди записувати значення	Старший байт
Байт3		Молодший байт
Байт4	Кількість бітових регістрів, куди записувати значення	Старший байт
Байт5		Молодший байт
Байт6	Кількість байтів бітових регістрів, куди записувати значення	Один байт
Байт7	Значення, записане в бітовий регістр	Один байт
…	Значення, записане в бітовий регістр	Один байт
Байт n	Значення, записане в бітовий регістр	Один байт
Байт n+1	Молодший байт контрольної суми CRC	Молодший байт
Байт n+2	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення-відповіді:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	Код функції	Один байт
Байт2	Початкова адреса бітових регістрів, куди записувати значення	Старший байт
Байт3		Молодший байт
Байт4	Кількість бітових регістрів, куди записувати значення	Старший байт
Байт5		Молодший байт
Байт6	Молодший байт контрольної суми CRC	Молодший байт
Байт7	Старший байт контрольної суми CRC	Старший байт

■ Структура даних повідомлення про виняток:

Байт NO.	Ім'я	Байт
Байт0	ID Modbus	Один байт
Байт1	16#80+ Код функції	Один байт
Байт2	Код відповіді винятку	Старший байт
Байт3	Молодший байт контрольної суми CRC	Молодший байт
Байт4	Старший байт контрольної суми CRC	Старший байт

Примітка. Кількість байтів даних у повідомленні запиту залежить від кількості бітових регістрів у повідомленні запиту.

■ **приклад**

Значення %QX0.0~%QX0.7 встановлено на 1000 0001, а адреса %QX0.0 — 16#A000 через код функції 0F у контролері руху.

Повідомлення запиту: 01 0F A0 00 00 08 01 81 26 55

Відповідь: 01 0F A0 00 00 08 76 0D

A.6 Таблиця регістрів і відповідних адрес Modbus

● Номери реєстрів у модулі керування рухом та відповідні адреси вказані нижче:

Зареєструвати ім'я	Реєстраційний номер	Пояснення	Адреса (шістнадцяткова)	Атрибут
I	%IX0.0~%IX127.7	Бітові регістри	6000 ~ 63FF	Лише для читання
Q	%QX0.0~%QX127.7		A000 ~ A3FF	Читати писати
I	%IW0~%IW63	Реєстри слів	8000 ~ 803F	Лише для читання
Q	%QW0~%QW63		A000 ~ A03F	Читати писати
M	%MW0~%MW32767		0000 ~ 7FFF	Читати писати

ПАМ'ЯТКА

B

Додаток B Зв'язок Modbus TCP

Зміст

B.1 Modbus TCP Message Structure B-2

B.2 Modbus Function Codes Supported in Modbus TCP B-2

B.3 Exception Response Code in Modbus TCP B-3

B.4 Modbus Function Codes in Modbus TCP B-3

B.5 Registers in PLC and Corresponding Modbus Addresses B-12

B.1 Структура повідомлення Modbus TCP

- **Структура повідомлення Modbus TCP**

Байт NO.	Ім'я		Пояснення
Байт0	Ідентифікатор транзакції	Старший байт	0
Байт1		Молодший байт	
Байт2	Ідентифікатор протоколу	Старший байт	0
Байт3		Молодший байт	
Байт4	Довжина даних Modbus	Старший байт	Кількість байтів адреси Modbus і даних після неї.
Байт5		Молодший байт	
Байт6	ID Modbus	Один байт	0 ~ 16#FF
Байт7	Код функції	Один байт	
Байт8	Адреса реєстрації в контролері	Старший байт	0~16#FFFF
Байт9		Молодший байт	
Байт10	Дані Modbus	Старший байт	Кількість байтів даних Modbus визначається кодом функції.

B

B.2 Коди функцій Modbus, що підтримуються в Modbus TCP

- **Коди функцій Modbus, які підтримує контролер руху**

Код функції	функція	зареєструватися
16#02	Зчитування значення бітового регістра; максимум 256 біт даних можна прочитати одночасно.	%IX і %QX
16#03	Читання одного або кількох значень регістра слів; максимум 100 слів даних можна прочитати одночасно.	%IW, %QW і %MW
16#05	Записати одне значення регістра одного біта.	% QX
16#06	Запишіть одне значення регістра слова.	%QW і %MW
16#0F	Записати кілька значень бітового регістра; максимум 256 біт даних можна записати одночасно.	% QX
16#10	Записати кілька значень регістра слів; одночасно можна записати максимум 100 слів даних.	%QW і %MW

B.3 Код відповіді винятку в Modbus TCP

- **Коди відповіді винятків Modbus, які підтримує контролер руху, наведено в таблиці нижче.**

Код відповіді винятку	Індикація
16#01	Непідтримуваний код функції
16#02	Непідтримувана адреса Modbus
16#03	Довжина даних перевищує діапазон

B.4 Коди функцій Modbus у Modbus TCP

- **Код функції: 03 для читання одного або кількох значень регістра слів**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Молодший байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса регістрів слів для читання	Старший байт
Байт9		Молодший байт
Байт10	Кількість регістрів слів (підраховується за словами)	Старший байт
Байт11		Молодший байт

  - **Структура даних повідомлення відповіді:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт

Байт NO.	Ім'я	Байт
Байт7	Код функції	Один байт
Байт8	Кількість прочитаних регістрів слів. (Рахунок по байтам)	Один байт
Байт9	Значення вмісту в реєстрі слів	Старший байт
Байт10		Молодший байт
…	Значення вмісту в реєстрі слів	Старший байт
Байт n		Молодший байт

- **Структура даних повідомлення про виняток:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

- **приклад**

  Щоб прочитати значення вмісту за адресами 16#0000 і 16#0001 в контролері руху за допомогою коду функції 03. 16#0000 і 16#0001 є адресою Modbus для %MW0 і %MW1 всередині контролера руху відповідно. Припустимо, що значення %MW0 дорівнює 16#0100, а значення %MW1 дорівнює 16#0200.

  Повідомлення запиту: 00 00 00 00 00 06 01 03 00 00 00 02

  Відповідь: 00 00 00 00 00 07 01 03 04 01 00 02 00

- **Код функції: 06 для запису одного значення регістра слова**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт

Байт NO.	Ім'я	Байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Адреса регістра слів, куди потрібно записати значення	Старший байт
Байт9		Молодший байт
Байт10	Значення, записане в реєстрі слів	Старший байт
Байт11		Молодший байт

- **Структура даних повідомлення відповіді:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Адреса регістру слів, куди потрібно записати значення	Старший байт
Байт9		Молодший байт
Байт10	Значення, записане в реєстрі слів	Старший байт
Байт11		Молодший байт

- **Структура даних повідомлення про виняток:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

- **приклад:**

  Щоб записати значення 16#0100 на адресу 16#0000 у контролері руху за допомогою коду функції 06

  Повідомлення запиту: 00 00 00 00 00 06 01 06 00 00 01 00

  Відповідь: 00 00 00 00 00 06 01 06 00 00 01 00

- **Код функції: 16#10 для запису значень регістру кількох слів**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса регістрів слів, куди записувати значення	Старший байт
Байт9		Молодший байт
Байт10	Кількість регістрів слів, куди записувати значення (підраховано за словами)	Старший байт
Байт11		Молодший байт
Байт12	Кількість регістрів слів, куди записувати значення (враховується в байтах)	Один байт
Байт13	Значення, записане в реєстрі слів	Старший байт
Байт14		Молодший байт
…	Значення, записане в реєстрі слів	Старший байт
Байт n		Молодший байт

  - **Структура даних повідомлення відповіді:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт

Байт NO.	Ім'я	Байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса регістрів слів, куди записувати значення	Старший байт
Байт9		Молодший байт
Байт10	Кількість регістрів слів, куди записувати значення. (Порахувати словом)	Старший байт
Байт11		Молодший байт

- **Структура даних повідомлення про виняток:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

Примітка:

Кількість байтів даних у повідомленні відповіді залежить від кількості прочитаних адрес регістра в контролері руху в повідомленні запиту. Отже, значення n у байті n у повідомленні відповіді можна обчислити шляхом зчитування кількості адрес реєстру в контролері руху.

- **приклад**

  Записати 16#0100 і 16#0200 на адреси 16#0000 і 16#0001 в контролері руху за допомогою коду функції 06.

  16#0000 і 16#0001 є адресами Modbus для %MW0 і %MW1 у контролері руху відповідно.

  Повідомлення запиту: 00 00 00 00 00 0B 01 10 00 00 00 02 04 01 00 02 00

  Відповідь: 00 00 00 00 00 06 01 10 00 00 00 02

- **Код функції: 16#02 для читання значень кількох бітових регістрів**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса бітових регістрів читання	Старший байт
Байт9		Молодший байт
Байт10	Кількість прочитаних бітових регістрів	Старший байт
Байт11		Молодший байт

  - **Структура даних повідомлення відповіді:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Скільки байтів для бітових регістрів читання	Один байт
Байт9	Значення стану бітового регістра, який зчитується	Один байт
…	Значення стану бітового регістра, який зчитується	Один байт
Байт n	Значення стану бітового регістра, який зчитується	Один байт

- **Структура даних повідомлення про виняток:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

- **приклад**

Щоб прочитати значення стану %QX2.0~%QX3.4 у контролері руху за допомогою коду функції 02. 16#A010 — адреса %QX2.0. Припустимо, що %QX2.0~%QX2.7=1000 0001 і %QX3.0~%QX3.4=10001.

Повідомлення запиту: 00 00 00 00 00 06 01 02 A0 10 00 0D

Відповідь: 00 00 00 00 00 06 01 02 02 81 11

- **Код функції: 16#05 для запису одного бітового значення регістра**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Адреса Modbus бітового регістра	Старший байт
Байт9		Молодший байт
Байт10	Значення, записане в бітовому регістрі	Старший байт
Байт11		Молодший байт

■ **Структура даних повідомлення відповіді:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Адреса Modbus бітового регістра	Старший байт
Байт9		Молодший байт
Байт10	Значення, записане в бітовому регістрі	Старший байт
Байт11		Молодший байт

■ **Структура даних повідомлення про виняток:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

Примітка. Записане значення 16#0000 означає, що 0 записується в бітовий регістр, а 16#FF00 означає, що 1 записується в бітовий регістр.

■ **приклад**

Встановіть значення %QX0.0 у контролері руху на 1 за допомогою коду функції 05; адреса %QX0.0 — 16#A000.

Повідомлення запиту: 00 00 00 00 00 06 01 05 A0 00 FF 00

Повідомлення відповіді: 00 00 00 00 00 06 01 05 A0 00 FF 00

- **Код функції: 16#0F для запису кількох значень бітового регістра.**
  - **Структура даних запиту повідомлення:**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса бітового регістра, куди записувати значення	Старший байт
Байт9		Молодший байт
Байт10	Кількість бітових регістрів, куди записувати значення	Старший байт
Байт11		Молодший байт
Байт12	Скільки байтів займають бітові регістри, куди записувати значення	Один байт
Байт13	Значення, записане в бітовому регістрі	Один байт
Байт n	Значення, записане в бітовому регістрі	Один байт

  - **Структура даних повідомлення відповіді**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Один байт
Байт5	ID Modbus	Старший байт
Байт6		Молодший байт
Байт7	Код функції	Один байт
Байт8	Початкова адреса бітових регістрів, де читати статус	Старший байт
Байт9		Молодший байт
Байт10	Кількість бітових регістрів, куди записувати значення	Старший байт
Байт 11		Молодший байт

- **Структура даних повідомлення про виняток**

Байт NO.	Ім'я	Байт
Байт0	Ідентифікатор транзакції	Старший байт
Байт1		Молодший байт
Байт2	Ідентифікатор протоколу	Старший байт
Байт3		Молодший байт
Байт4	Довжина даних Modbus	Старший байт
Байт5		Молодший байт
Байт6	ID Modbus	Один байт
Байт7	16#80+ код функції	Один байт
Байт8	Код відповіді винятку	Один байт

- **приклад**

Встановіть %QX0.0~%QX0.7=1000 0001 за допомогою коду функції 0F і встановіть адресу %QX0.0 на 16#A000 у контролері руху.
Повідомлення запиту: 00 00 00 00 00 0A 01 0F A0 00 00 08 01 81
Повідомлення відповіді: 00 00 00 00 00 06 01 0F A0 00 00 08

B.5 Реєстри в ПЛК і відповідні адреси Modbus

Зареєструвати ім'я	реєстраційний номер	Пояснення	Адреса (шістнадцяткова)	Атрибут
I	%IX0.0~%IX127.7	Бітовий регістр	6000 ~ 63FF	Лише для читання
Q	%QX0.0~%QX127.7		A000 ~ A3FF	Читати писати
I	%IW0~%IW63	Реєстр слів	8000 ~ 803F	Лише для читання
Q	%QW0~%QW63		A000 ~ A03F	Читати писати
M	%MW0~%MW32767		0000 ~ 7FFF	Читати писати

Додаток C Протокол CANopen

Зміст

C.1 Node States C-4

C.2 Network Management (NMT) C-6

C.3 PDO (Process Data Object) C-6

C.4 SDO (Service Data Object) C-8

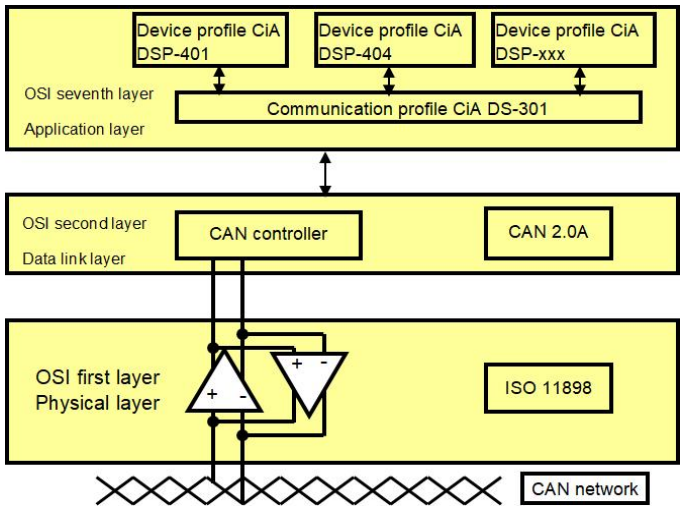
- **Про протокол CANopen**

  Польова шина CAN (мережі контролера) визначає лише фізичний рівень і рівень каналу передачі даних. (Див. стандарт ISO11898.) Він не визначає прикладний рівень. У практичному проекті фізичний рівень і рівень каналу даних реалізуються апаратним забезпеченням. Сама польова шина CAN не завершена. Йому потрібен вищий протокол, щоб визначити використання 11/29-бітного ідентифікатора та 8-байтових даних.

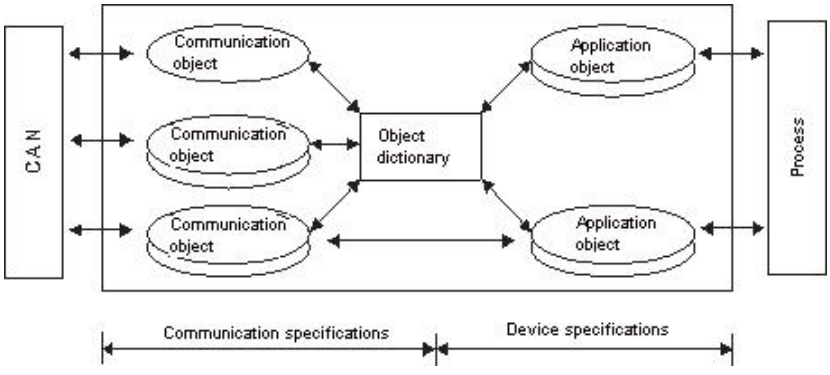
  Протокол CANopen є найкращим протоколом на основі CAN. Це один із протоколів, визначених і підтримуваних CiA (CAN-in-Automation). Він розроблений на основі протоколу CAL (протокол прикладного рівня CAN), використовуючи підмножину протоколів зв'язку та обслуговування CAL.

  Протокол CANopen охоплює прикладний рівень і профіль зв'язку (CiA DS301). Він також охоплює структуру для програмованих регістрів (CiA 302), рекомендації щодо кабелів і з'єднувачів (CiA 303-1), а також одиниць CI та представлення префіксів (CiA 303-2).

  У моделі OSI зв'язок між стандартом CAN і протоколом CANopen такий.



- **Об'єктний словник**
  - CANopen використовує об'єктний спосіб визначення стандартного пристрою. Кожен пристрій представлено набором об'єктів, і його можна відвідати в мережі. Модель пристрою CANopen наведена нижче. Як показано на малюнку нижче, словник об'єктів є інтерфейсом між комунікаційною програмою та вищою прикладною програмою.
  - Основною концепцією CANopen є словник об'єктів пристрою (OD). Це впорядкований набір об'єктів. Кожен об'єкт приймає 16-бітний індекс для адресації. Щоб дозволити відвідування окремого елемента в структурі даних, він також визначає 8-бітовий субіндекс. Кожен вузол у мережі CANopen має словник об'єктів. Словник об'єктів містить параметри, які описують пристрій і поведінку мережі. Словник об'єктів вузла описаний в електронному паспорті.



- **Комунікаційний об’єкт CANopen**

  Протокол зв’язку CANopen містить PDO, SDO, NMT та інші попередньо визначені об’єкти зв’язку CANopen.

  Зверніться до розділу C.3 для ознайомлення з PDO.

  Зверніться до розділу C.4 для ознайомлення з SDO.

  Зверніться до розділу C.2 для ознайомлення з NMT.

- **Інші попередньо визначені комунікаційні об’єкти CANopen (SYNC і EMCY)**

  - **Об’єкт SYNC (синхронний об’єкт)**
    Синхронний об’єкт — це повідомлення, яке періодично транслюється головним вузлом у мережі CANopen. Цей об'єкт використовується для реалізації тактового сигналу мережі. Кожен пристрій вирішує, чи використовувати подію та здійснювати синхронний зв’язок з іншими мережевими пристроями відповідно до своєї конфігурації. Наприклад, при управлінні приводним пристроєм пристрої не діють відразу після отримання команди, надісланої майстром. Вони діють, поки не отримають синхронне повідомлення. Таким чином багато пристроїв можуть працювати синхронно.

    Формат повідомлення SYNC:

COB-ID
80 (шістнадцятковий)

  - **Аварійний об'єкт**
    Екстрений об’єкт використовується пристроєм CANopen для вказівки на внутрішню помилку. Коли в пристрої виникає екстрена помилка, пристрій надсилає екстрене повідомлення (включаючи код екстреної помилки), і пристрій переходить у стан помилки. Після усунення помилки пристрій надсилає екстрене повідомлення, код екстреної помилки 0, і пристрій переходить у нормальний стан.
    Формат екстреного повідомлення:

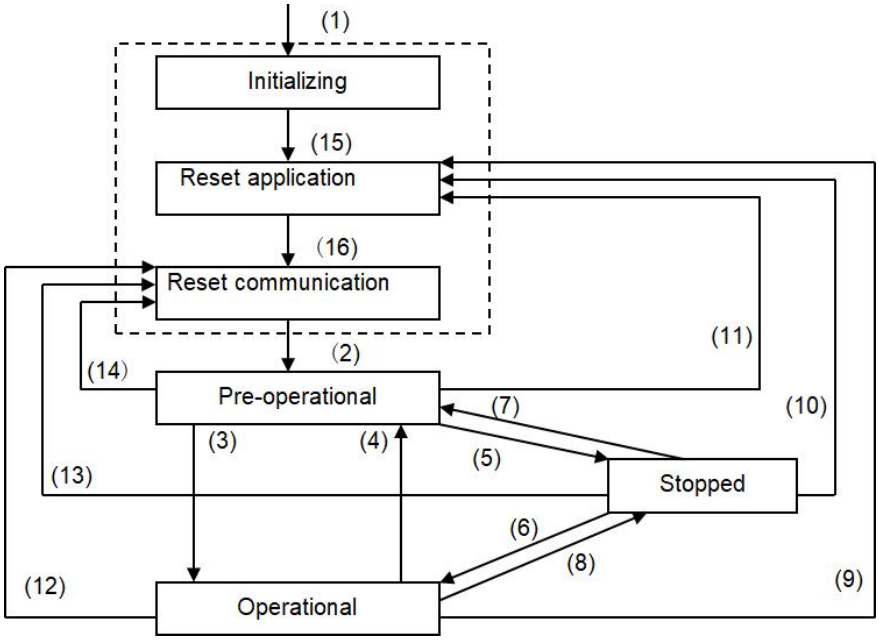
COB-ID	**Байт 0**	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	**Байт 6**	**Байт 7**
80 (шістнадцятковий) +Node-ID	Код екстреної помилки		Реєстр помилок	Заводський код помилки				
	LSB	MSB						

Примітка. Значення в регістрі помилок зіставляється з індексом 1001 (шістнадцятковий) у словнику об’єктів. Якщо значення дорівнює 0, помилка не виникає. Якщо значення дорівнює 1, виникає загальна помилка. Якщо значення H'80, у пристрої сталася внутрішня помилка.

C.1 Стан вузла

- **Сервіси керування модулем**

  Головний вузол у мережі CANopen керує підлеглим, надсилаючи команду. Підлеглий виконує команду після отримання команди, і йому не потрібно відповідати. Усі вузли CANopen мають внутрішні стани NMT. Підлеглий вузол має чотири стани: ініціалізація, передопераційний, робочий і зупинений стан.
  Нижче наведено стан пристрою.



C

(1) Після подачі живлення пристрій автоматично переходить у стан ініціалізації.

(2)Після завершення ініціалізації пристрій автоматично переходить у передопераційний стан.

(3)(3)(6) Віддалений вузол запущено.

(4)(7) Пристрій переходить у передопераційний стан.

(5)(8) Віддалений вузол зупинено.

(9)(10)(11) Прикладний рівень скинуто.

(12)(13)(14) Зв’язок скинуто.

(15) Після завершення ініціалізації пристрій автоматично переходить у стан «скидання програми».

(16)Після завершення стану «скидання програми» пристрій автоматично переходить у стан «скидання зв’язку».

Зв'язок між об'єктом зв'язку та станом показаний нижче. Обслуговування об'єкта зв'язку може виконуватися тільки в належному стані. Наприклад, SDO може бути виконано тільки в робочому стані та в передопераційному стані.

	Ініціалізація	**Передопераційний**	**Оперативний**	**Зупинився**
PDO			X	
SDO		X	X	
SYNC		X	X	
Позначка часу		X	X	
EMCY		X	X	
Завантаження	X			
NMT		X	X	X

Формат контрольного повідомлення для стану вузла:

COB-ID	Байт 0	Байт 1
0	Специфікатор команди (CS)	Підлегла адреса (0: широкомовна)

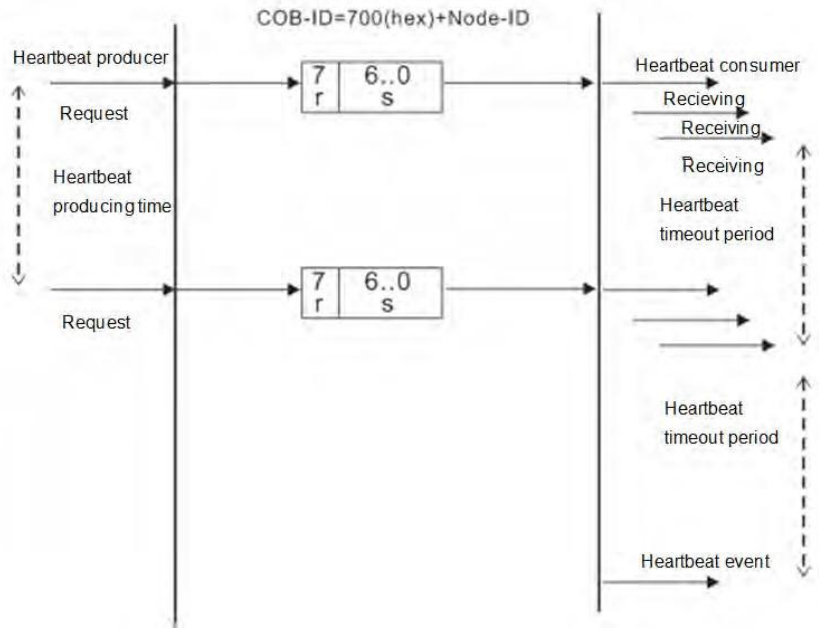
Специфікатори команд наведено нижче.

Специфікатор команди (шістнадцятковий)	функція
01	Запустіть віддалений вузол
02	Зупиніть віддалений вузол
80	Перейти в передексплуатаційний стан
81	Скинути прикладний рівень
82	Скинути зв'язок

- **Послуги контролю помилок**

Служба контролю помилок використовується для виявлення відключення вузла в мережі. Служби контролю помилок можна класифікувати на два типи: Heartbeat і Node Guarding. ПЛК підтримує лише Heartbeat. Наприклад, головний може виявити відключення підлеглого лише після того, як підлеглий увімкне службу Heartbeat.

Принцип Heartbeat проілюстрований наступним чином. Виробник Heartbeat передає повідомлення Heartbeat відповідно до встановленого часу створення Heartbeat. Один або кілька споживачів Heartbeat виявляють повідомлення, передане виробником Heartbeat. Якщо споживач не отримує повідомлення, передане виробником протягом періоду очікування, згенерована подія пульсу вказує на те, що зв’язок CANopen є ненормальним.



- **Послуги завантаження**

Після того, як підлеглий пристрій перейде в передопераційний стан, він передасть повідомлення Boot-up, яке вказує на завершення ініціалізації.

C.2 Керування мережею (NMT)

Керування мережею CANopen відповідає режиму Master/Slave. Може існувати лише один головний NMT, а інші вузли вважаються підлеглими в мережі CANopen. NMT містить три типи послуг: служби керування модулями, служби контролю помилок і служби завантаження. Додаткову інформацію див. у розділі C.1 посібника.

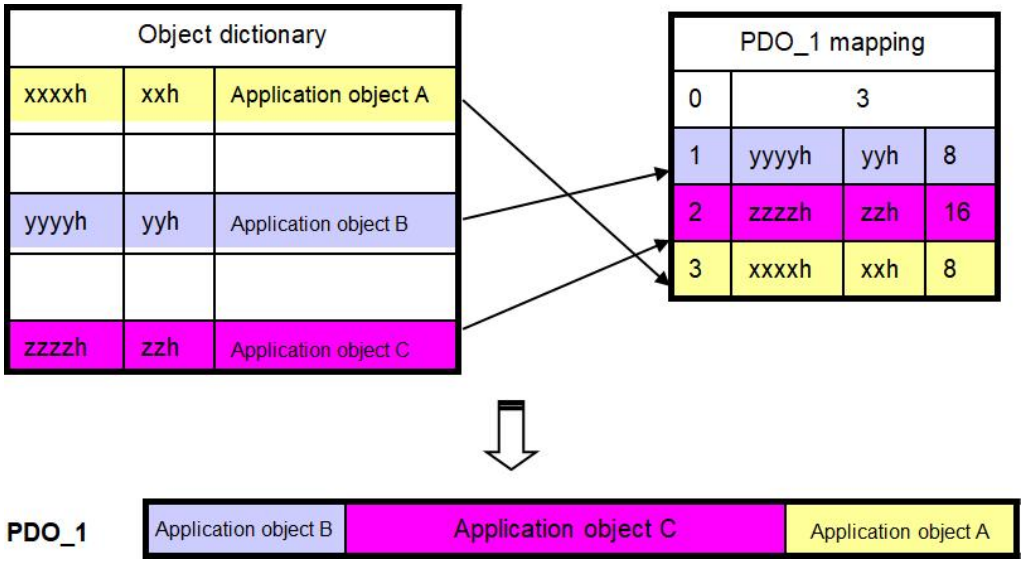
C.3 PDO (об’єкт даних процесу)

- **PDO**
  - PDO забезпечує канал прямого відвідування для об’єкта програми пристрою, використовується для передачі даних у реальному часі та має високий пріоритет. Кожен байт у списку даних повідомлень PDO CAN використовується для передачі даних. Швидкість використання повідомлення висока.
  - PDO описується за допомогою режиму «виробник/споживач». Дані передаються від одного виробника до одного або багатьох споживачів. Дані, які можна передати, обмежені від 1 до 8 байт. Після того, як дані передаються виробником, споживачеві не потрібно відповідати на дані. Кожен вузол у мережі виявляє інформацію про дані, передану вузлом передачі, і вирішує, чи обробляти отримані дані.
  - Для кожного PDO існує два типи служб PDO: TxPDO та RxPDO. PDO, надісланий виробником, називається PDO (TxPDO), надісланий пристроєм виробника. А PDO, який отримує споживач, називається PDO (RxPDO), який отримує пристрій споживача.
  - Кожен PDO описується двома об’єктами в словнику об’єктів: параметрами зв’язку PDO та параметрами відображення PDO.
    Параметри зв'язку PDO:
    Включіть COB-ID, який буде використовуватися PDO, тип передачі, час заборони та цикл лічильника.
    Параметри зіставлення PDO:
    Містить список об’єктів у словнику об’єктів. Ці об’єкти відображаються в PDO, включаючи довжину даних (у бітах). Щоб пояснити зміст PDO, виробник і споживач повинні розуміти відображення.
  - Режими передачі PDO: синхронний і асинхронний Синхронний: синхронний періодичний і синхронний неперіодичний
    Асинхронний: PDO передається, коли дані змінюються, або після запуску події.
    - PDO підтримує такі режими передачі.

Тип			Передача PDO		
	Періодичні	Неперіодичні	Синхронний	Асинхронний	PTP
0		X	X		
1 – 240	X		X		
254				X	
255				X	

Режим 0: інформація PDO передається лише тоді, коли дані PDO змінюються та надходить синхронний сигнал.
Режими 1~240: один фрагмент інформації PDO передається кожні 1~240 синхронних сигналів.
Режим 254: Передача тригера події визначається виробником. Для контролера руху визначення таке ж, як і режим 255.
Режим 255: PDO передається, коли дані змінюються, або після запуску події.

- Усі дані в PDO мають бути зіставлені зі словника об’єктів. Нижче наведено приклад зіставлення PDO.



- Формат даних для RxPDO і TxPDO такий.

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
Ідентифікатор об'єкта	Дані							

C.4 SDO (об’єкт службових даних)

- **SDO**
  - SDO використовується для побудови зв’язку клієнт/сервер між двома пристроями CANopen. Клієнтський пристрій може читати дані з об’єктного словника серверного пристрою та записувати дані в об’єктний словник серверного пристрою. Режим доступу SDO — режим «клієнт/сервер». Режимом, до якого здійснюється доступ, є сервер SDO. Кожен пристрій CANopen має принаймні один об’єкт службових даних, який забезпечує канал доступу до словника об’єктів пристрою. SDO може читати всі об’єкти в словнику об’єктів і записувати всі об’єкти в словник об’єктів.
  - Повідомлення SDO містить інформацію про індекс та інформацію про субіндекс, які можна використовувати для розміщення об’єктів у словнику об’єктів, а складена структура даних може легко пропускати доступ SDO. Тригерний метод SDO відноситься до типу командної відповіді. Іншими словами, сервер SDO повинен відповісти після того, як клієнт SDO надішле запит на читання/запис. Клієнт і сервер можуть зупинити передачу SDO. Повідомлення запиту та повідомлення відповіді можна відрізнити відповідно до їхніх різних COB-ID.
  - SDO може передавати дані будь-якої довжини. Якщо довжина даних перевищує 4 байти, дані повинні передаватися сегментами. Останній сегмент даних містить позначку кінця. Структури запитуваного повідомлення SDO та повідомлення відповіді такі. Формати повідомлення запиту та повідомлення відповіді:
    - Формат повідомлення запиту

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
600 (шістнадцятковий) +Node-ID	Код запиту	Індекс об'єкта		Індекс об'єкта	Запитувані дані			
		LSB	MSB		біт 7-0	біт 15-8	біт 23-16	біт 31-24

    - Визначення коду запиту в повідомленні запиту:

Код запиту (шістнадцятковий)	опис
23	Запис 4-байтових даних
2B	Запис 2-байтових даних
2F	Запис 1-байтових даних
40	Читання даних
80	Зупинка поточної функції SDO

    - Формат повідомлення-відповіді

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
580 (шістнадцятковий) +Node-ID	Код відповіді	Індекс об'єкта		Підіндекс об'єкта	Дані відповіді			
		LSB	MSB		біт 7-0	біт 15-8	біт 23-16	біт 31-24

    - Визначення коду відповіді у повідомленні відповіді:

Код відповіді (шістнадцятковий)	опис
43	Читання 4-байтових даних
4B	Читання 2-байтових даних
4F	Читання 1-байтових даних
60	Запис 1/2/4-байтних даних
80	Зупинка функції SDO

D

Додаток D Пояснення режимів наведення

Зміст

D.1 Explanation of Homing Modes D-2

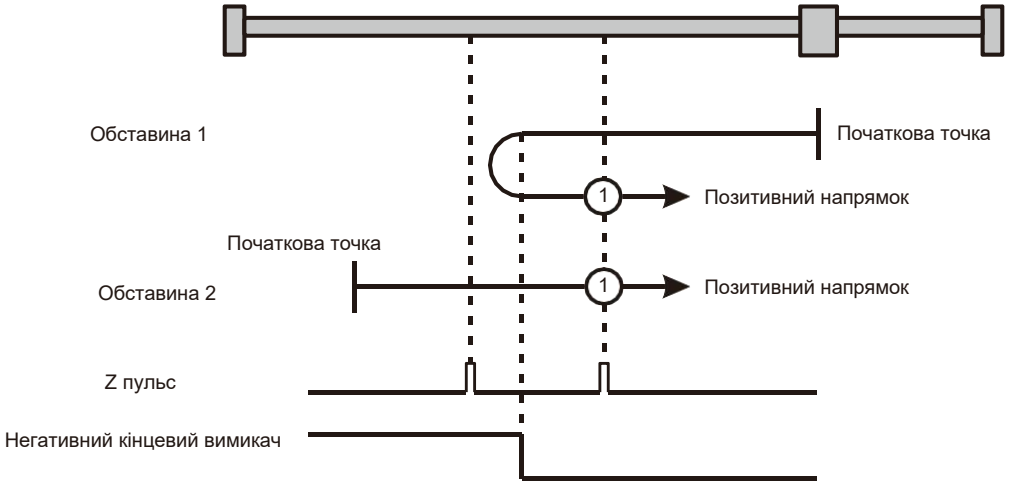
D.1 Пояснення режимів наведення

контролер руху забезпечує багато режимів самонаведення, з яких користувач може вибрати відповідний відповідно до польових умов і технічних вимог.

- Режим 1 Налаштування, яке залежить від негативного кінцевого вимикача та Z-імпульсу.

  Обставина 1 : інструкція MC_Home виконується, коли негативний кінцевий вимикач вимкнено, а вісь рухається у негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли вісь виявляє, що негативний кінцевий вимикач увімкнено. Коли зустрічається перший Z-імпульс, це вихідне положення, коли негативний кінцевий вимикач вимкнено.

  Обставина 2 : інструкція MC_Home виконується, коли негативний кінцевий вимикач увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Коли зустрічається перший Z-імпульс, це вихідне положення, коли негативний кінцевий вимикач вимкнено.



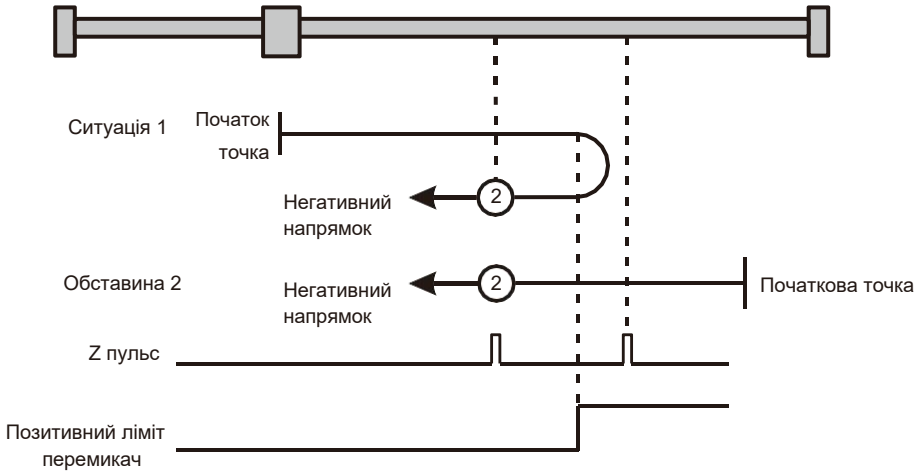
Повернення до початкової точки залежно від негативного кінцевого вимикача та Z-імпульсу (① : режим 1)

D

➢ Режим 2 Повернення до початкової точки, яке залежить від позитивного кінцевого вимикача та Z-імпульсу

Обставина 1 : інструкція MC_Home виконується, коли позитивний кінцевий вимикач вимкнено, а вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли вісь виявляє, що позитивний кінцевий вимикач увімкнено. Там, де зустрічається перший Z-імпульс, це початкове положення, тоді як позитивний кінцевий вимикач ВИМКНЕНО.

Обставина 2 : інструкція MC_Home виконується, коли позитивний кінцевий вимикач увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Там, де зустрічається перший Z-імпульс, це початкове положення, тоді як позитивний кінцевий вимикач ВИМКНЕНО.



Повернення до початкового положення в залежності від позитивного кінцевого вимикача та імпульсу Z (② : режим 2)

Режим 3 і режим 4 Наведення, яке залежить від перемикача вдома та Z-імпульсу

- Режим 3

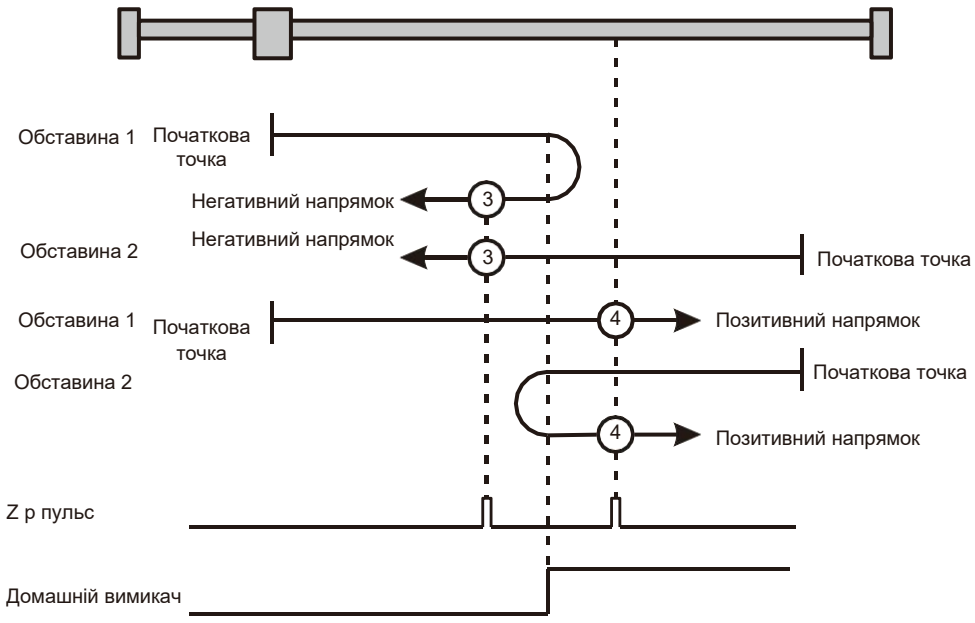
Обставина 1 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Коли вісь виявляє, що вихідний перемикач увімкнено, напрямок руху змінюється, і вісь рухається зі швидкістю другої фази. Перший Z-імпульс зустрічається у вихідному положенні, коли вихідний перемикач у ВИМК.

Обставина 2 : коли перемикач вихідного сигналу ввімкнено, інструкція MC_Home виконується, і вісь безпосередньо рухається у негативному напрямку зі швидкістю другої фази. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.

- Режим 4

Обставина 1 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Вісь рухається зі швидкістю другої фази, коли вісь стикається з тим, що вихідний перемикач увімкнено. Місце, де зустрічається перший Z-імпульс, є початковою позицією.

Обставина 2 : коли перемикач вихідного сигналу ввімкнено, виконується інструкція MC_Home, і вісь рухається у негативному напрямку зі швидкістю другої фази. Коли вісь стикається з тим, що вихідний перемикач вимкнено, напрямок руху змінюється, і вісь рухається зі швидкістю другої фази. Місце, де зустрічається перший Z-імпульс, є початковою позицією.



Налаштування в залежності від домашнього перемикача та Z-імпульсу (③ : режим 3; ④ : режим 4)

Режим 5 і режим 6 Наведення, яке залежить від вихідного перемикача та Z-імпульсу

- Режим 5

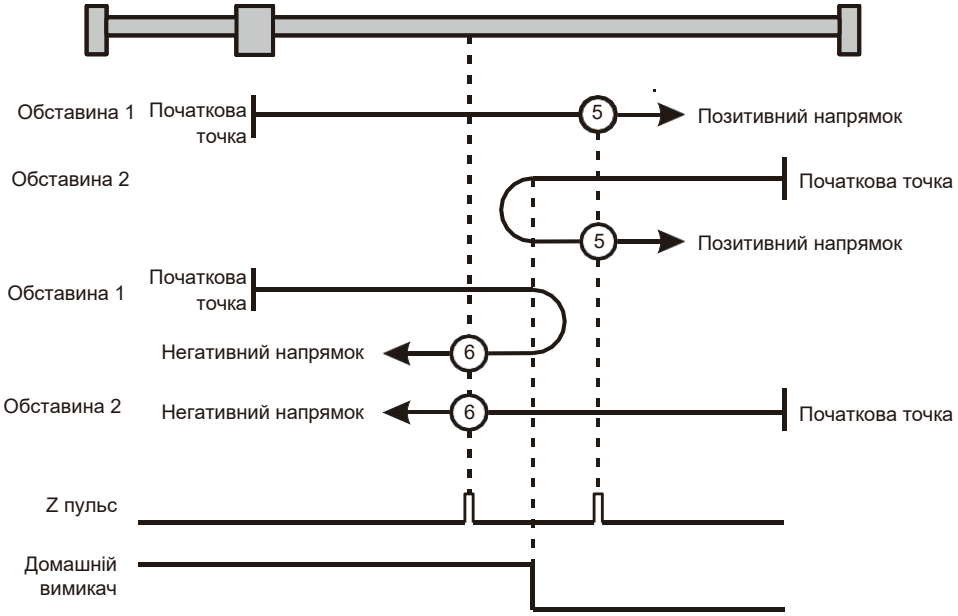
Обставина 1 : коли перемикач вихідного сигналу ввімкнено, виконується інструкція MC_Home, і вісь рухається в позитивному напрямку зі швидкістю другої фази. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.

Обставина 2 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в негативному напрямку зі швидкістю першої фази. Коли головний перемикач увімкнено, напрямок руху змінюється, і вісь рухається зі швидкістю другої фази. Перший Z-імпульс зустрічається у вихідному положенні, коли вихідний перемикач у ВИМК.

- Режим 6

Обставина 1 : коли перемикач вихідного сигналу ввімкнено, виконується інструкція MC_Home, і вісь рухається в позитивному напрямку зі швидкістю другої фази. Коли головний перемикач вимкнено, напрямок руху змінюється, і вісь рухається зі швидкістю другої фази. Місце, де зустрічається перший Z-імпульс, є початковою позицією.

Обставина 2 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в негативному напрямку зі швидкістю першої фази. Коли вихідний перемикач увімкнено, вісь рухається зі швидкістю другої фази, і місце, де зустрічається перший Z-імпульс, є початковим положенням.



Налаштування в залежності від домашнього перемикача та Z-імпульсу
(⑤ : режим 5, ⑥ : режим 6)

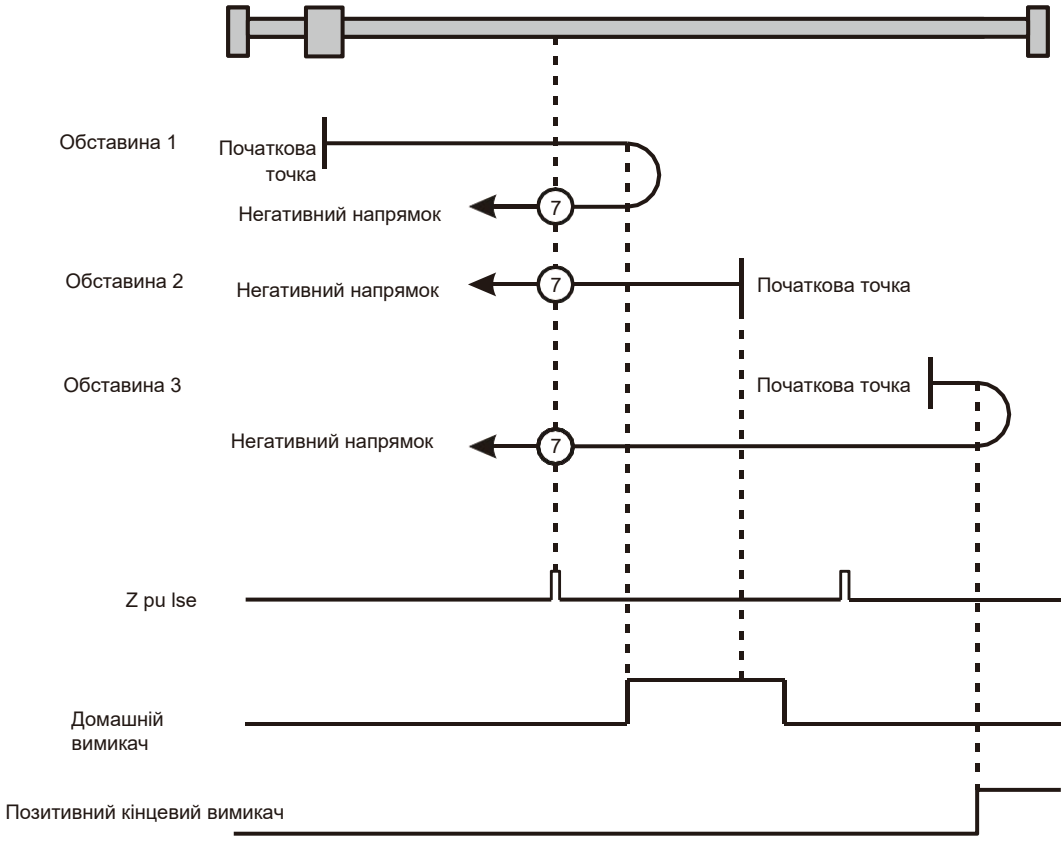
Режим 7~ режим 10 Налаштування, яке залежить від вихідного вимикача, позитивного кінцевого вимикача та Z-імпульсу

- Режим 7

Обставина 1 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач початкового положення увімкнено. Перший Z-імпульс зустрічається у вихідному положенні, коли вихідний перемикач у ВИМК.

Обставина 2 : коли перемикач вихідного сигналу ввімкнено, виконується інструкція MC_Home, і вісь рухається у негативному напрямку зі швидкістю другої фази. Перший Z-імпульс зустрічається у вихідному положенні, коли вихідний перемикач у ВИМК.

Обставина 3 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Вісь починає рухатися зі швидкістю другої фази, коли вихідний перемикач увімкнено. Перший Z-імпульс зустрічається у вихідному положенні, коли вихідний перемикач у ВИМК.



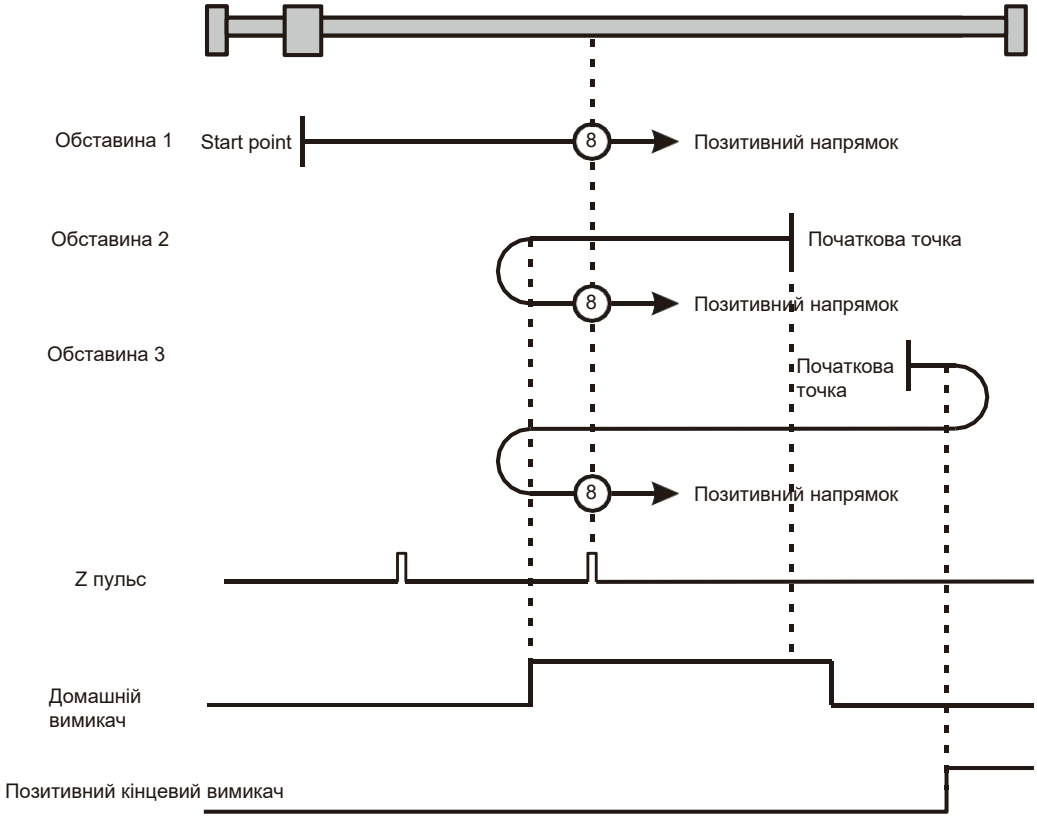
Повернення до початкового положення в залежності від вихідного вимикача, позитивного кінцевого вимикача та Z-імпульсу (⑦ : режим 7)

➢ Режим 8

Обставина 1 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено, а перший Z-імпульс зустрічається у вихідному положенні.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 3 : коли перемикач вихідного сигналу вимкнено, інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Вісь все ще рухається зі швидкістю першої фази, коли вихідний перемикач увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли головний перемикач ВИМКНЕНО. Вісь рухається зі швидкістю другої фази, і місце, де зустрічається перший Z-імпульс, є вихідним положенням, коли вихідний перемикач увімкнено.



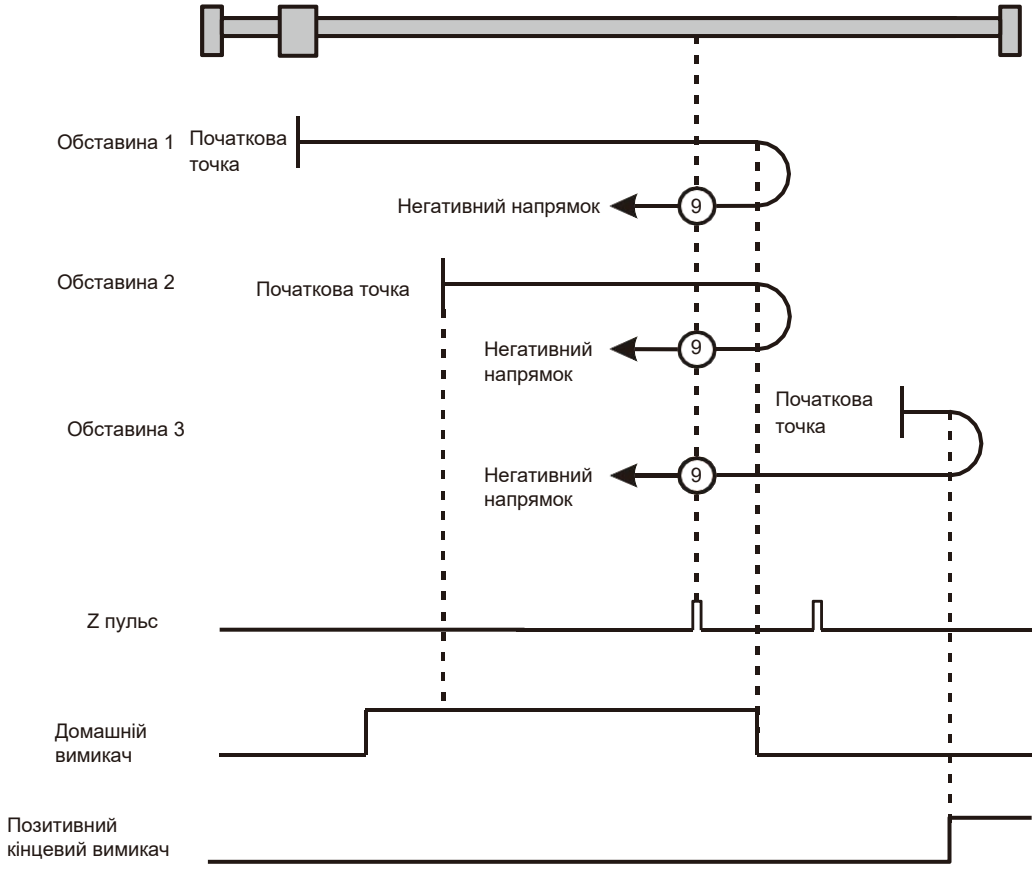
Перехід до початкового положення залежно від вихідного вимикача, позитивного кінцевого вимикача та Z-імпульсу (⑧ : режим 8)

➢ Режим 9

Обставина 1 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази, коли вихідний перемикач вимкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 2 : коли перемикач вихідного сигналу ввімкнено, виконується інструкція MC_Home, і вісь рухається в позитивному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази, коли перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Вісь рухається зі швидкістю другої фази, і місце, де зустрічається перший Z-імпульс, є вихідним положенням, коли вихідний перемикач увімкнено.



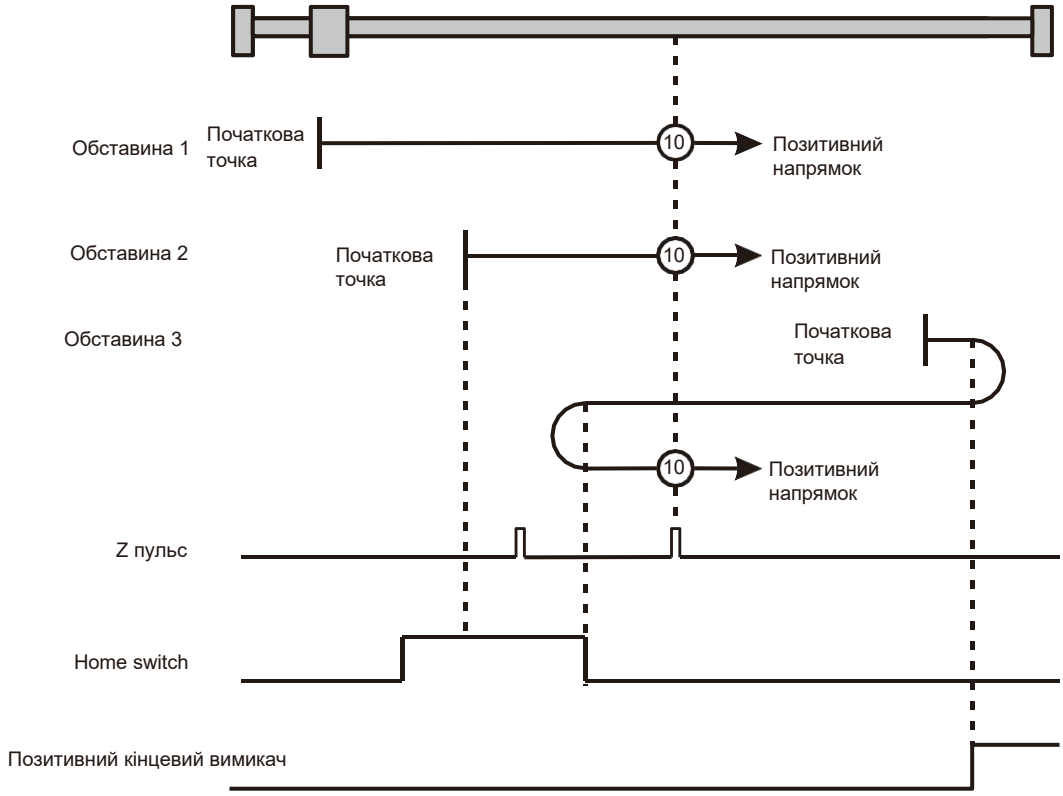
Повернення до початкового положення в залежності від вихідного вимикача, позитивного кінцевого вимикача та Z-імпульсу (⑨ : режим 9)

➢ Режим 10

Обставина 1 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази, коли вихідний перемикач вимкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. І місце, де зустрічається перший Z-імпульс, є початковим положенням, коли перемикач вихідного сигналу вимкнено.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу увімкнено. І місце, де зустрічається перший Z-імпульс, є початковим положенням, коли перемикач вихідного сигналу вимкнено.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази, коли перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач увІМКНЕНО. Напрямок руху знову змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.



Повернення до початкового положення в залежності від вихідного вимикача,

позитивного кінцевого вимикача та Z-імпульсу (⑩ : режим 10) Режим 11~, режим 14

Повернення до початкового положення, яке залежить від вихідного вимикача, негативного

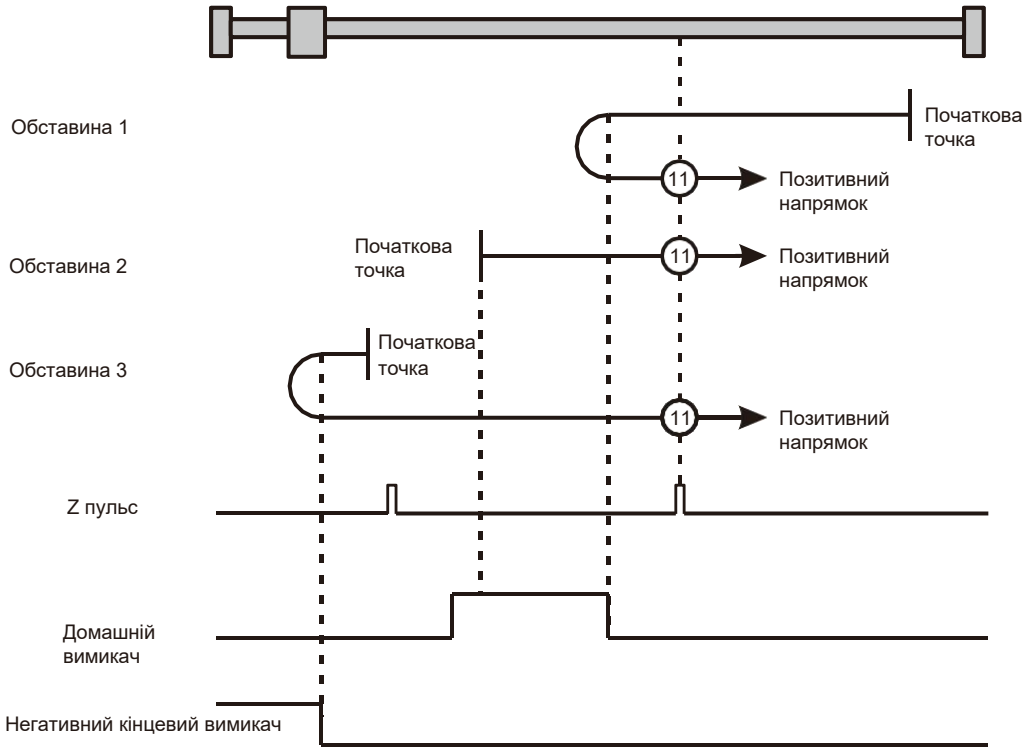
кінцевого вимикача та Z-імпульсу

➢ Режим 11

Обставина 1 : інструкція MC_Home виконується, і вісь рухається в негативному напрямку на швидкості першої фази, коли перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач початкового положення увімкнено. І місце, де зустрічається перший Z-імпульс, є початковим положенням, коли перемикач вихідного сигналу вимкнено.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. І місце, де зустрічається перший Z-імпульс, є початковим положенням, коли перемикач вихідного сигналу вимкнено.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку на швидкості першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.



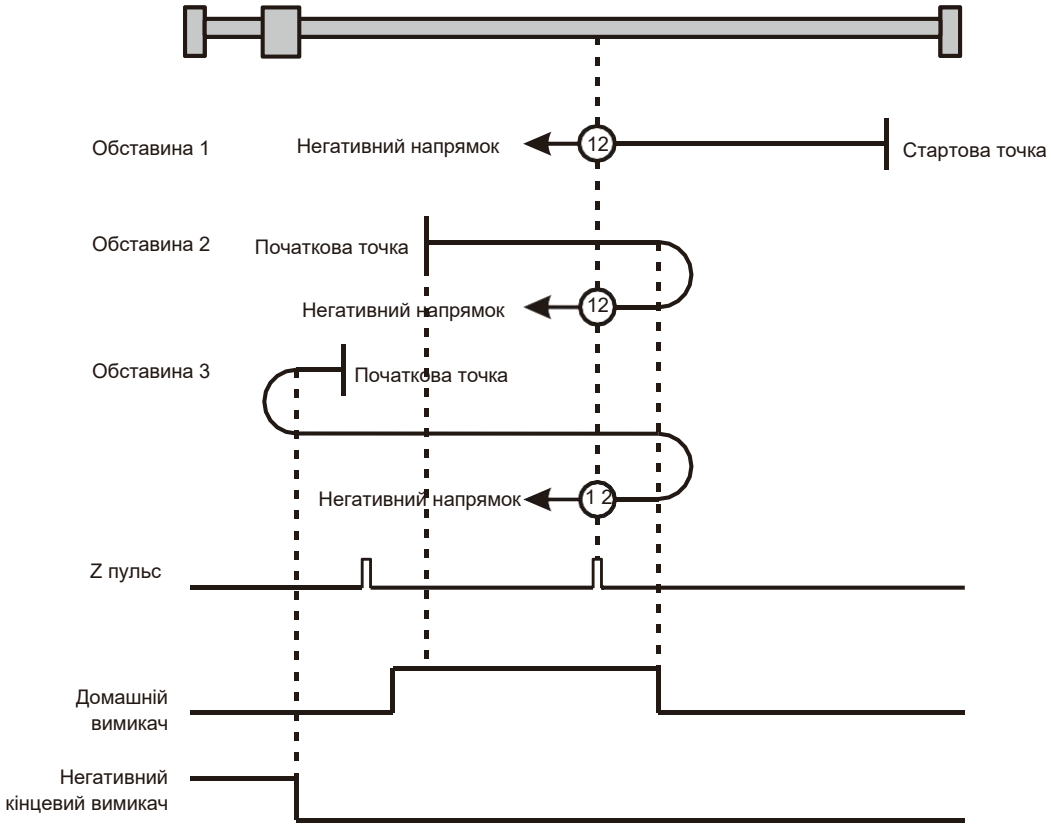
Повернення до початкового положення залежно від вихідного вимикача, негативного кінцевого вимикача та Z-імпульсу (◯11 : режим 11)

➢ Режим 12

Обставина 1 : інструкція MC_Home виконується, і вісь рухається в негативному напрямку на швидкості першої фази, коли перемикач початкового положення вимкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, поки перемикач початкового положення вимкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку на швидкості першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Вісь все ще рухається зі швидкістю першої фази, коли вихідний перемикач увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, поки перемикач початкового положення вимкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.



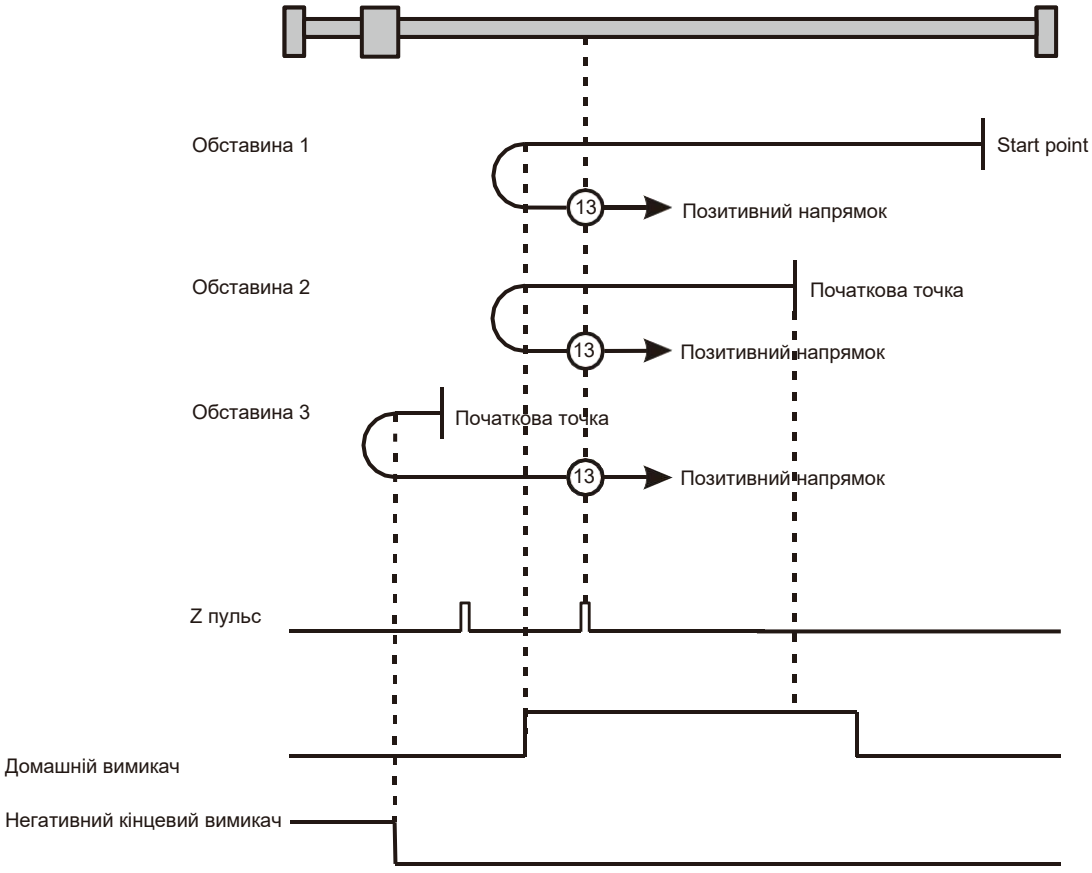
Повернення до початкового положення залежно від вихідного вимикача, негативного кінцевого вимикача та Z-імпульсу (◯ 12 : режим 12)

➢ Режим 13

Обставина 1 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю першої фази, тоді як перемикач початкового положення вимкнено. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, у той час як головний перемикач ВИМКНЕНО. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю другої фази, поки перемикач початкового положення увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, поки перемикач початкового положення вимкнено. І місце, де зустрічається перший Z-імпульс, є вихідною позицією.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку на швидкості першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Вісь рухається зі швидкістю другої фази, і місце, де зустрічається перший Z-імпульс, є вихідним положенням, коли вихідний перемикач увімкнено.



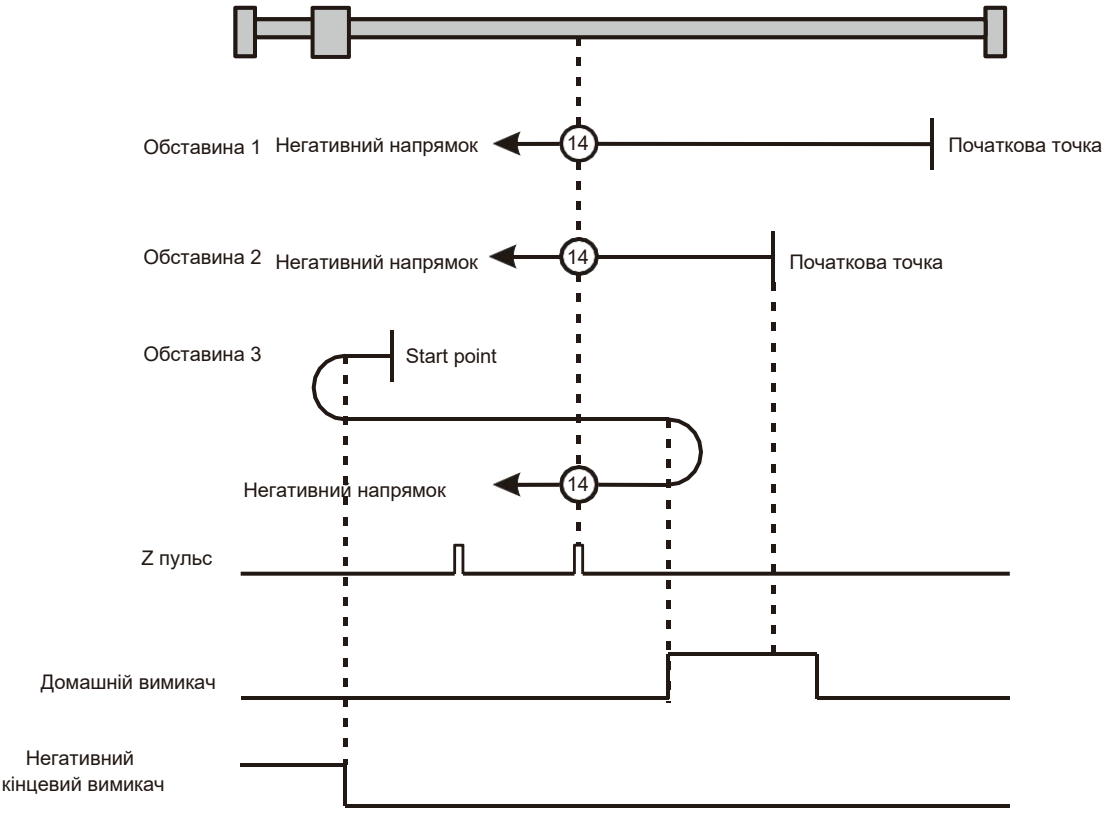
Перехід до початкового положення залежно від вихідного вимикача, негативного кінцевого вимикача та Z-імпульсу (◯ 13 : режим 13)

➢ Режим 14

Обставина 1 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю першої фази, тоді як перемикач початкового положення вимкнено. Вісь рухається зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. І місце, де зустрічається перший Z-імпульс, є початковим положенням, коли перемикач вихідного сигналу вимкнено.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю другої фази, поки перемикач початкового положення увімкнено. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.

Обставина 3 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Напрямок руху знову змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. Там, де зустрічається перший Z-імпульс, це початкове положення, коли перемикач вихідного сигналу вимкнено.



Повернення до початкового положення в залежності від вихідного вимикача, негативного кінцевого вимикача та Z-імпульсу (◯ 14 : режим 14) Режим 15 і режим 16 зарезервовано для майбутнього розвитку.

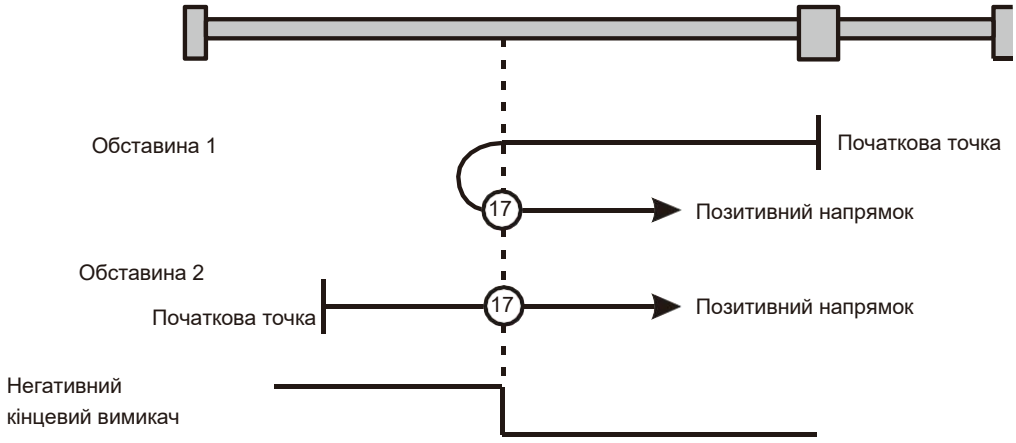
Режим 17~режим 30 Наведення, яке не має нічого спільного з Z-імпульсом

У режимі 17~режимі 30, який відповідно схожий на режим 1~режим 14, згаданий раніше, вісь не має нічого спільного з Z-імпульсом, крім відповідного початкового перемикача та статусу кінцевого вимикача під час повернення у вихідне положення. Режим 17 подібний до режиму 1, режим 18 подібний до режиму 2, режим 19 і режим 20 подібний до режиму 3, режим 21 і режим 22 подібний до режиму 5, режим 23 і режим 24 подібний до режиму 7, режим 25 і режим 26 подібні до режиму 9, режим 27 і режим 28 подібні до режиму 11, а режим 29 і режим 30 подібні до 13.

➢ Режим 17 Повернення до початкової точки, яке залежить від негативного кінцевого вимикача

Обставина 1 : інструкція MC_Home виконується, коли негативний кінцевий вимикач вимкнено, а вісь рухається у негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли вісь виявляє, що негативний кінцевий вимикач увімкнено. Там, де сервопривід знаходиться, коли негативний кінцевий вимикач ВИМКНЕНО, це початкове положення.

Обставина 2 : інструкція MC_Home виконується, коли негативний кінцевий вимикач увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Сервопривід знаходиться у вихідному положенні, коли негативний кінцевий вимикач ВИМКНЕНО.

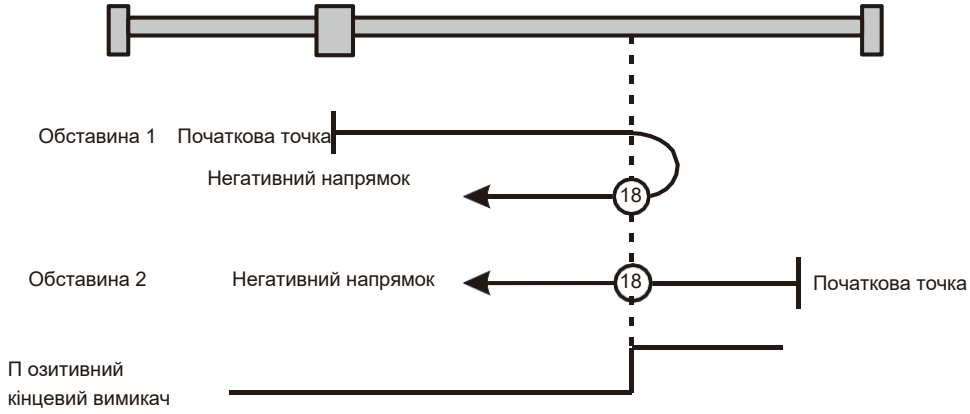


Перехід до початкового положення залежно від негативного кінцевого вимикача (◯ 17 : режим 17)

➢ Режим 18 Повернення до початкової точки, яке залежить від позитивного кінцевого вимикача

Обставина 1 : інструкція MC_Home виконується, коли позитивний кінцевий вимикач вимкнено, а вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли вісь виявляє, що позитивний кінцевий вимикач увімкнено. Сервопривід знаходиться у вихідному положенні, тоді як позитивний кінцевий вимикач ВИМКНЕНО.

Обставина 2 : інструкція MC_Home виконується, коли позитивний кінцевий вимикач увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Сервопривід знаходиться у вихідному положенні, тоді як позитивний кінцевий вимикач ВИМКНЕНО.

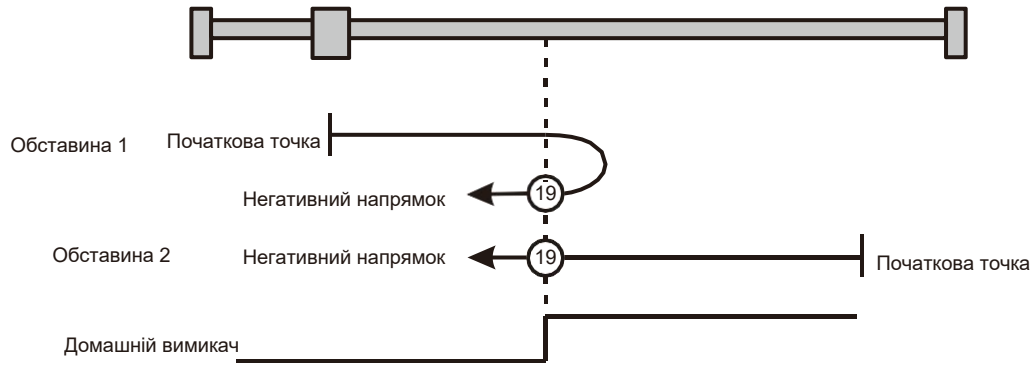


Повернення до початкового положення в залежності від позитивного кінцевого вимикача (◯ 18 : режим 18)

➢ Режим 19

Обставина 1 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, як тільки перемикач початкового положення стає УВІМКНЕНИМ. І де стоїть вісь, це початкове положення в момент, коли перемикач вихідного положення стає ВИМКНЕНИМ.

Обставина 2 : інструкція MC_Home виконується, і вісь напряму рухається у негативному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу увімкнено. І де знаходиться вісь, це вихідне положення в момент, коли перемикач вихідного положення стає ВИМК.

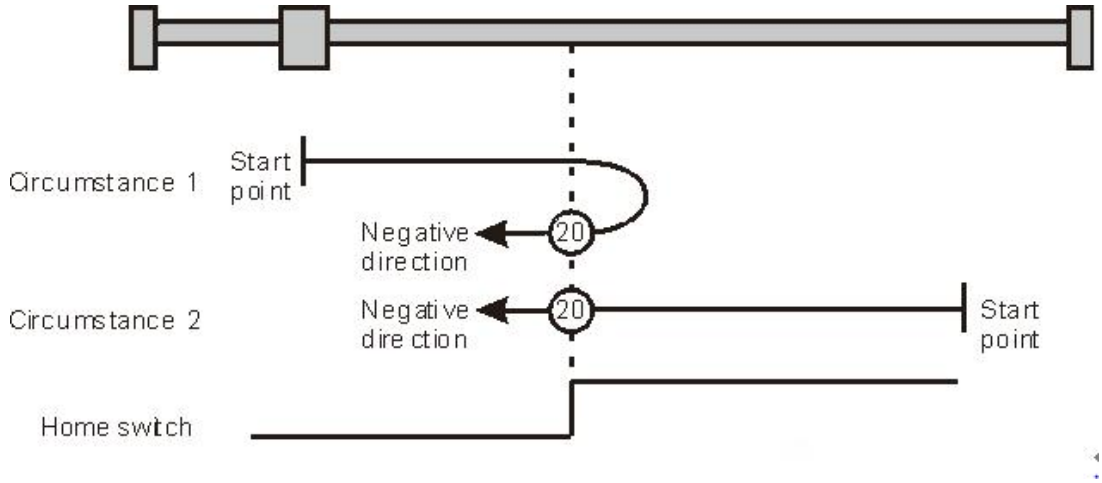


Налаштування в залежності від домашнього перемикача (◯ 19 : режим 19)

➢ Режим 20

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь рухається в позитивному напрямку зі швидкістю першої фази. Там, де знаходиться сервопривід, це початкове положення, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу ввімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь переміщується зі швидкістю другої фази, коли головний перемикач стає ВИМК. Там, де знаходиться сервопривід, це початкове положення, коли перемикач вихідного сигналу увімкнено.

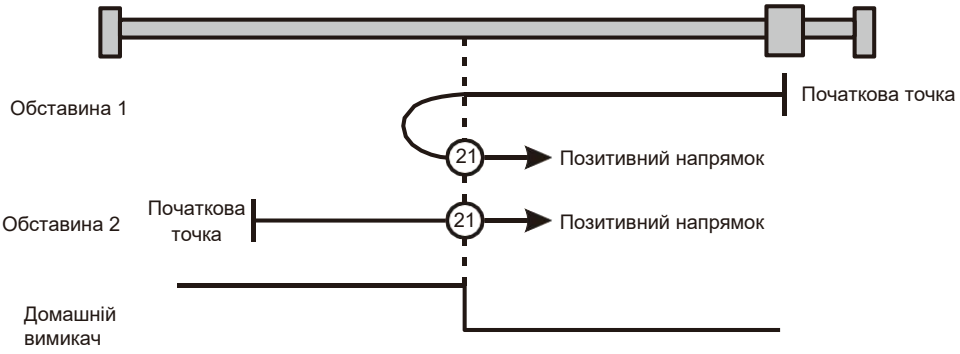


Налаштування в залежності від домашнього перемикача (◯ 20 : режим 20)

➢ Режим 21

Обставина 1 : інструкція MC_Home виконується, і вісь рухається у негативному напрямку зі швидкістю першої фази, тоді як перемикач початкового положення вимкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, як тільки перемикач початкового положення стає УВІМКНЕНИМ. І де стоїть вісь, це початкове положення в момент, коли перемикач вихідного положення стає ВИМКНЕНИМ.

Обставина 2 : інструкція MC_Home виконується, і вісь рухається в позитивному напрямку зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. І де стоїть вісь, це початкове положення в момент, коли перемикач вихідного положення стає ВИМКНЕНИМ.



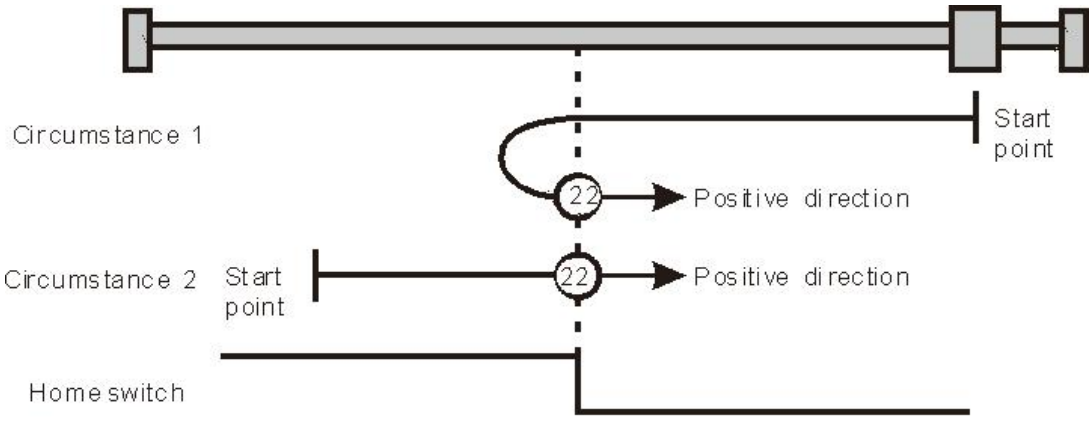
Налаштування в залежності від домашнього перемикача (◯ 21 : режим 21)

➢ Режим 22

Обставина 1 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь переміщується зі швидкістю другої фази, коли перемикач початкового положення стає ВИМКНЕНИМ. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь рухається в негативному напрямку зі швидкістю першої фази. Там, де знаходиться вісь, знаходиться початкове положення, коли перемикач вихідного сигналу стає УВІМК.

D



Homing depending on the home switch (22 : mode 22)

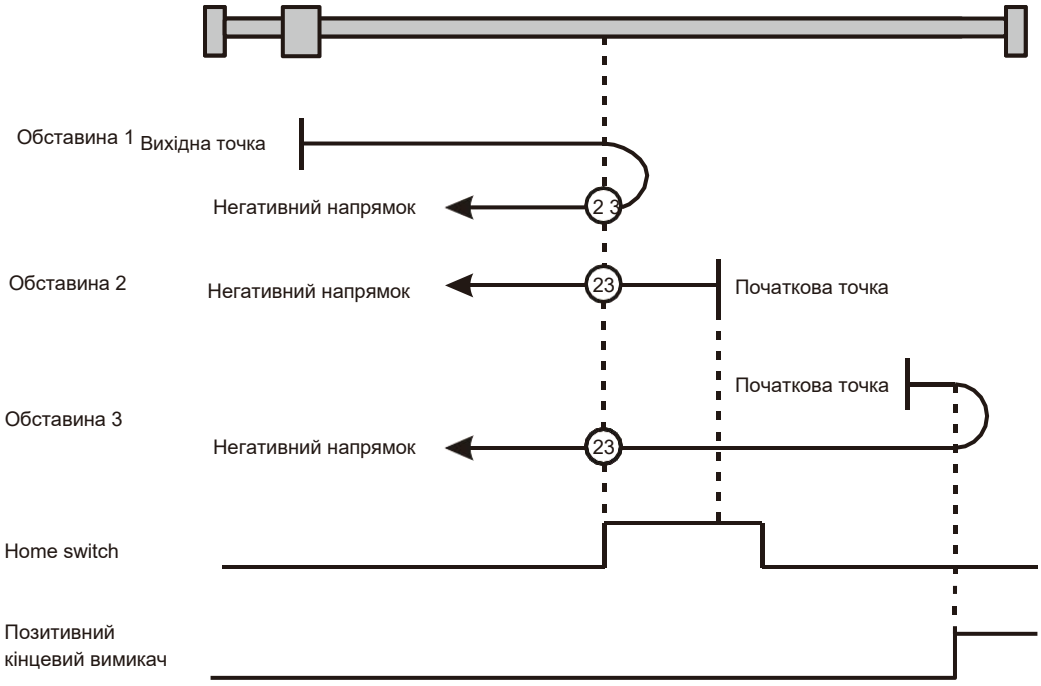
Налаштування в залежності від домашнього перемикача (◯ 22 : режим 22)

➢ Режим 23

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь переміщується зі швидкістю другої фази, коли перемикач початкового положення стає УВІМКНЕНИМ. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. І там, де стоїть вісь, це початкове положення, коли перемикач вихідного положення стає ВИМКНЕНИМ.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач увІМКНЕНО. Коли вихідний перемикач увімкнено, вісь починає рухатися зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.



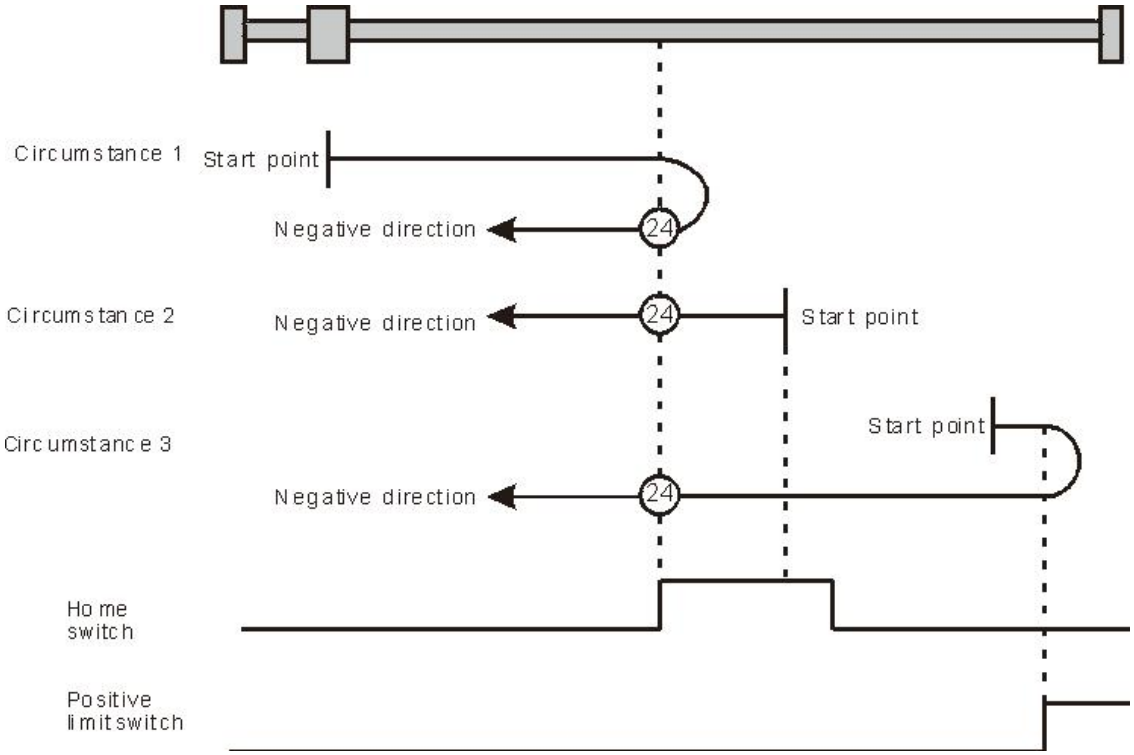
Повернення до початкового положення залежно від вихідного вимикача та позитивного кінцевого вимикача (◯ 23 : режим 23)

➢ Режим 24

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в позитивному напрямку зі швидкістю першої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Коли вихідний перемикач увімкнено, вісь усе ще рухається зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.



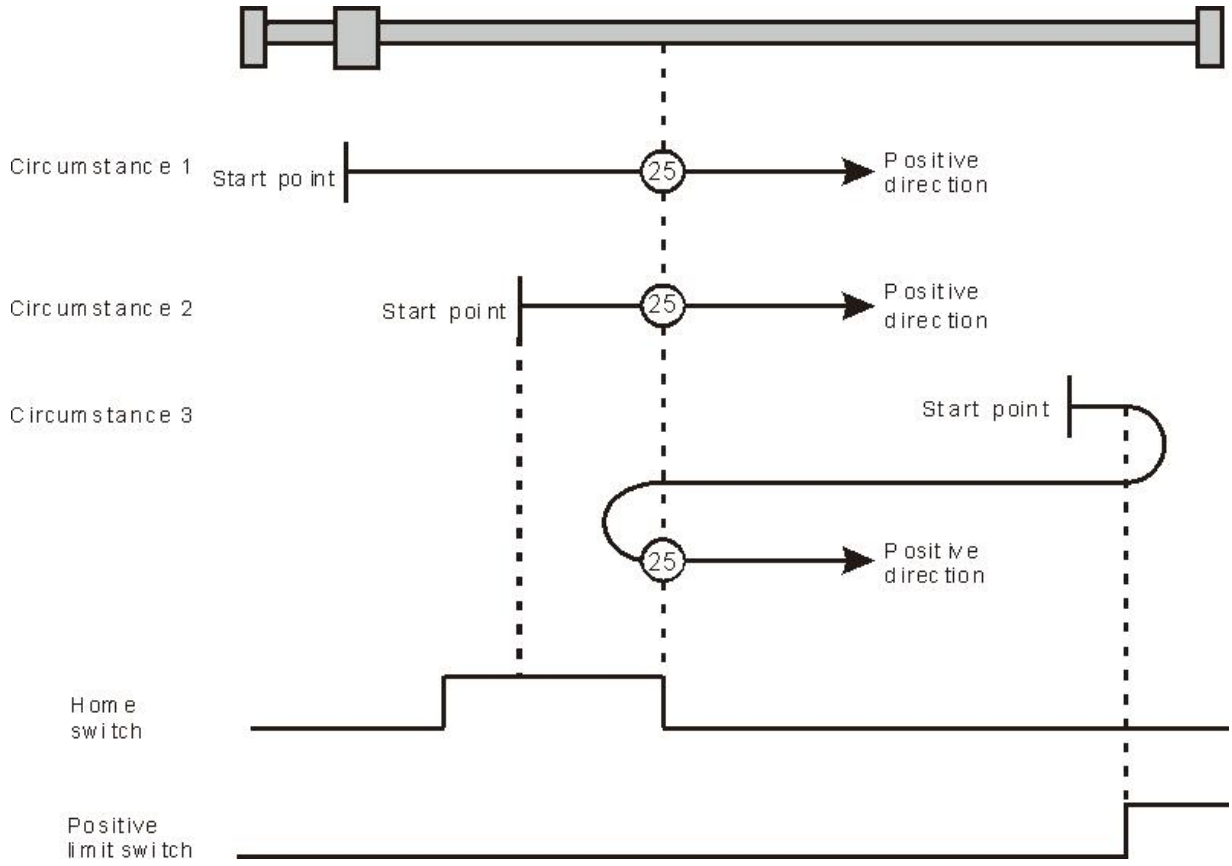
Повернення до початкового положення залежно від вихідного вимикача та позитивного кінцевого вимикача (◯ 24 : режим 24)

➢ Режим 25

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в позитивному напрямку зі швидкістю першої фази. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.



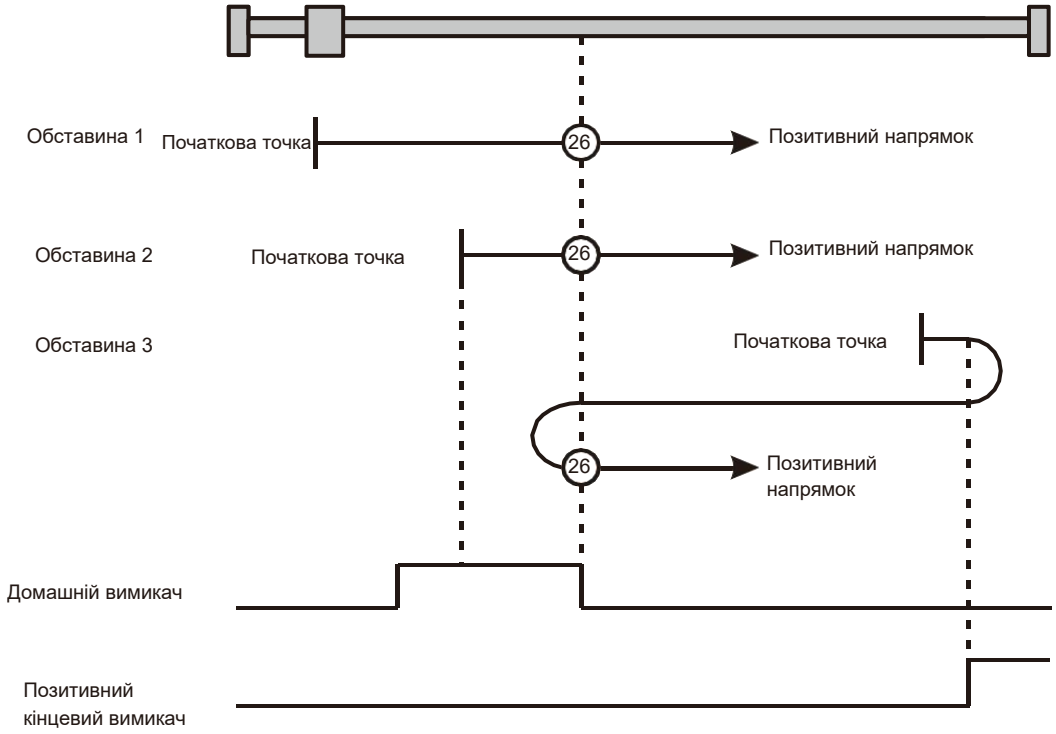
Повернення до початкового положення залежно від вихідного вимикача та позитивного кінцевого вимикача (◯ 25 : режим 25)

➢ Режим 26

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в позитивному напрямку зі швидкістю першої фази. Вісь рухається зі швидкістю другої фази, коли вихідний перемикач увімкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в позитивному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач ВИМКНЕНО, а позитивний кінцевий вимикач уВІМКНЕНО. Напрямок руху знову змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач вихідного сигналу ввімкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.



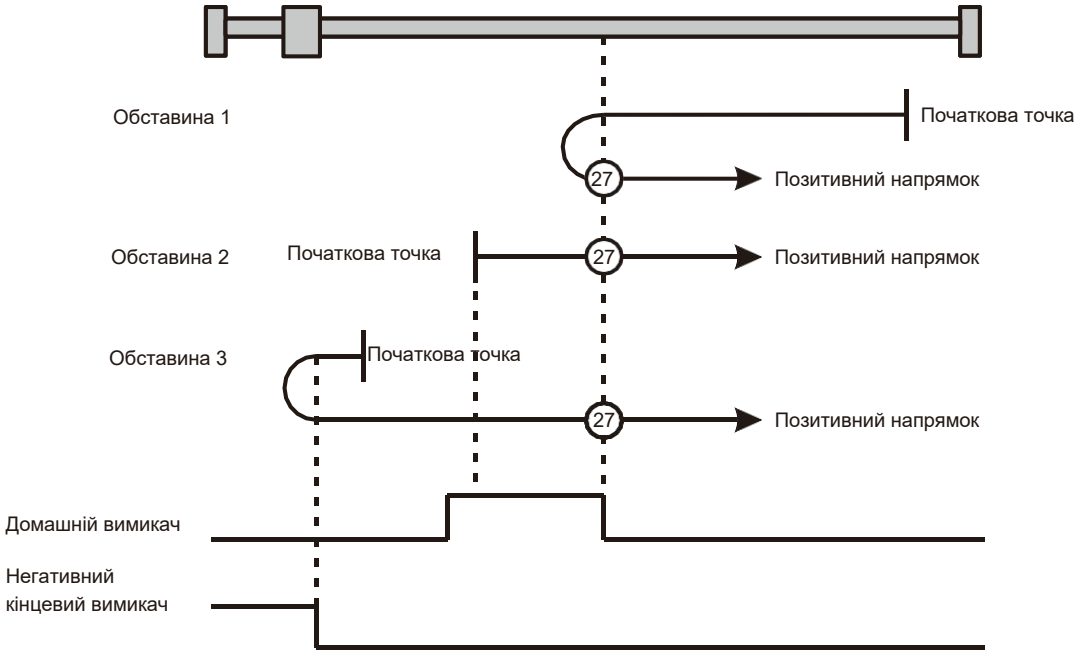
Повернення до початкового положення залежно від вихідного вимикача та позитивного кінцевого вимикача (◯ 26 : режим 26)

➢ Режим 27

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли перемикач початкового положення увімкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Коли вихідний перемикач увімкнено, вісь починає рухатися зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.



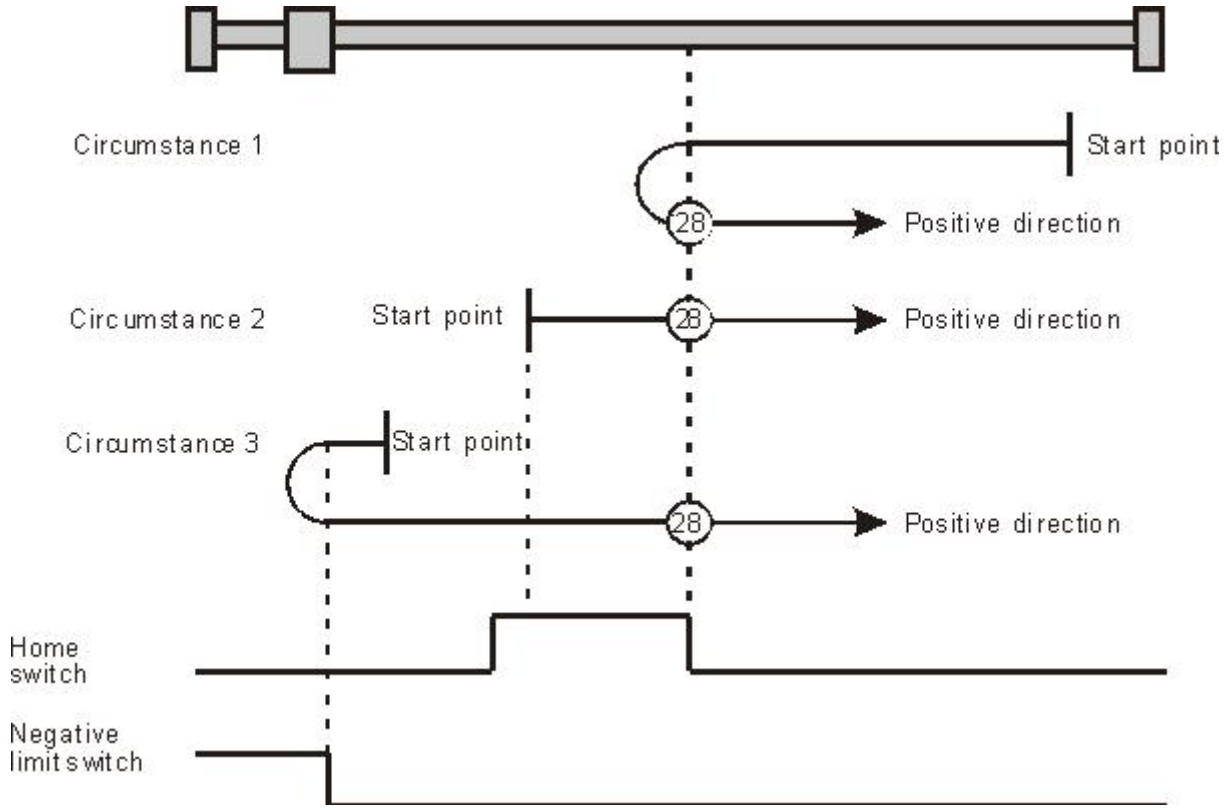
Повернення до початкового положення залежно від вихідного вимикача та негативного кінцевого вимикача (◯ 27: режим 27)

➢ Режим 28

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в негативному напрямку зі швидкістю першої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається в позитивному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Коли вихідний перемикач увімкнено, вісь усе ще рухається зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли головний перемикач ВИМКНЕНО. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.



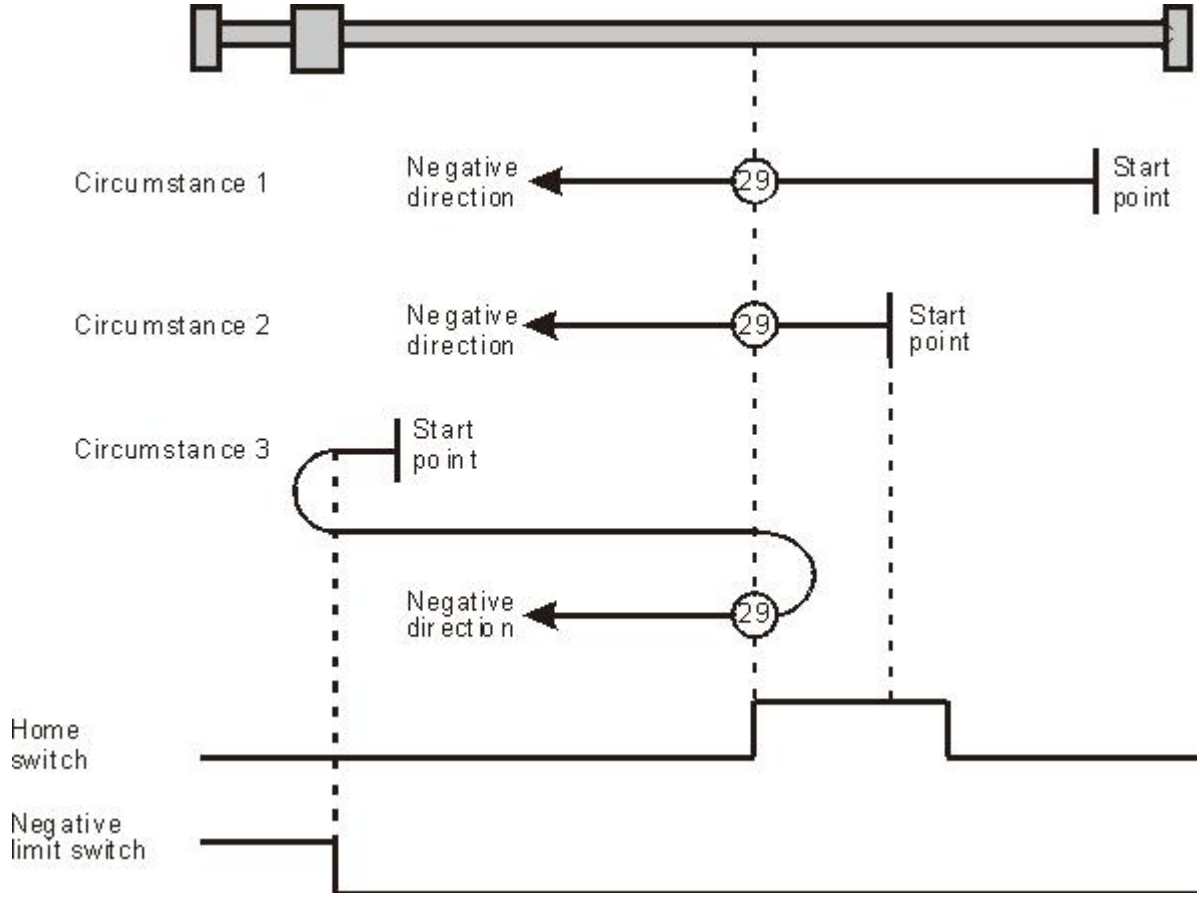
Повернення до початкового положення залежно від вихідного вимикача та негативного кінцевого вимикача (◯ 28 : режим 28)

➢ Режим 29

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в негативному напрямку зі швидкістю першої фази. Коли вихідний перемикач увімкнено, вісь починає рухатися зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю другої фази, коли головний перемикач вимкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного сигналу увімкнено.



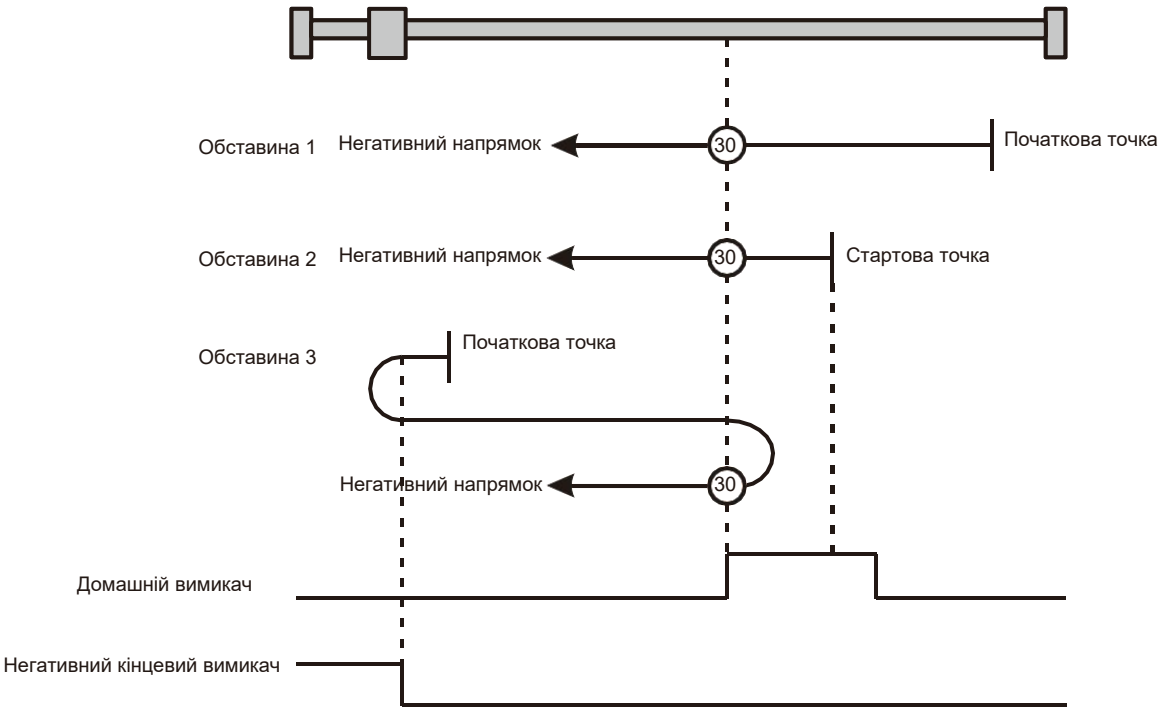
Повернення до початкового положення залежно від вихідного вимикача та негативного кінцевого вимикача (◯ 29 : режим 29)

➢ Режим 30

Обставина 1 : інструкція MC_Home виконується, коли перемикач початкового положення вимкнено, а вісь починає рухатися в негативному напрямку зі швидкістю першої фази. Коли вихідний перемикач увімкнено, вісь починає рухатися зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 2 : інструкція MC_Home виконується, коли перемикач вихідного сигналу увімкнено, а вісь рухається у негативному напрямку зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.

Обставина 3 : Інструкція MC_Home виконується, коли перемикач Home знаходиться у ВИМКНЕНОМУ положенні. Вісь рухається в негативному напрямку зі швидкістю першої фази. Напрямок руху змінюється, і вісь рухається зі швидкістю першої фази, коли вихідний перемикач вимкнено, а негативний кінцевий перемикач увімкнено. Коли вихідний перемикач увімкнено, напрямок руху знову змінюється, і вісь рухається зі швидкістю другої фази. Там, де стоїть вісь, є вихідна позиція, коли перемикач вихідного положення вимкнено.



Повернення до початкового положення залежно від вихідного вимикача та негативного кінцевого вимикача (◯ 30 : режим 30)

Режим 31 і режим 32 Зарезервовано для майбутнього розвитку.

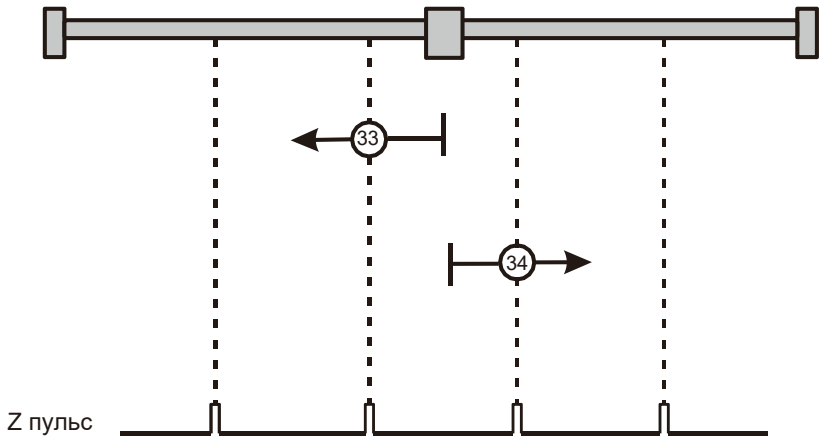
Режим 33 ~ режим 34 Наведення, яке залежить від імпульсу Z

➢ Режим 33

У режимі 33 виконується інструкція MC_Home і вісь рухається зі швидкістю другої фази в негативному напрямку. І місце, де стоїть вісь, є вихідним положенням, коли зустрічається перший імпульс Z.

➢ Режим 34

У режимі 34 виконується інструкція MC_Home і вісь рухається зі швидкістю другої фази в позитивному напрямку. І місце, де стоїть вісь, є вихідним положенням, коли зустрічається перший імпульс Z.



Наведення залежно від імпульсу Z (◯33 : режим 33, ◯34 : режим 34)

Режим 35 Наведення залежить від поточної позиції

У режимі 35 виконується інструкція MC_Home, вісь не рухається, а її поточне положення вважається початковим.

Пам'ятка

Додаток E Список аксесуарів

Зміст

E.1 Accessories for EtherCAT Communication E-2
E.2 Accessories for CANopen Communication..... E-2
E.3 Accessories for DeviceNet Communication E-5

E.1 Аксесуари для зв’язку EtherCAT

- **Кабелі**

малюнок	Модель	Довжина	Діаметр (AWG)
	UC-EMC003-02A	0,3 м.	4#22 ПВХ
	UC-EMC005-02A	0,5 м.	4#22 ПВХ
	UC-EMC010-02A	1,0 м.	4#22 ПВХ
	UC-EMC020-02A	2,0 м.	4#22 ПВХ
	UC-EMC050-02A	5,0 м.	4#22 ПВХ
	UC-EMC100-02A	10,0 м.	4#22 ПВХ
	UC-EMC200-02A	20,0 м.	4#22 ПВХ

E.2 Аксесуари для зв’язку CANopen

- **Кабелі**

малюнок	Модель	Довжина	Діаметр (AWG)
	UC-DN01Z-01A	305M	2#15 · 2#18 SHLD PVC (Товстий кабель)
	UC-DN01Z-02A	305M	2#22 · 2#24 SHLD PVC (Тонкий кабель)
	UC-CMC003-01A	0,3 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC005-01A	0,5 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC010-01A	1,0 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC015-01A	1,5 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC020-01A	2,0 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC030-01A	3,0 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC050-01A	5,0 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC100-01A	10,0 м.	4#26 · 1#24 PVC (Тонкий кабель)
	UC-CMC200-01A	20,0 м.	4#26 · 1#24 PVC (Тонкий кабель)

Примітки:

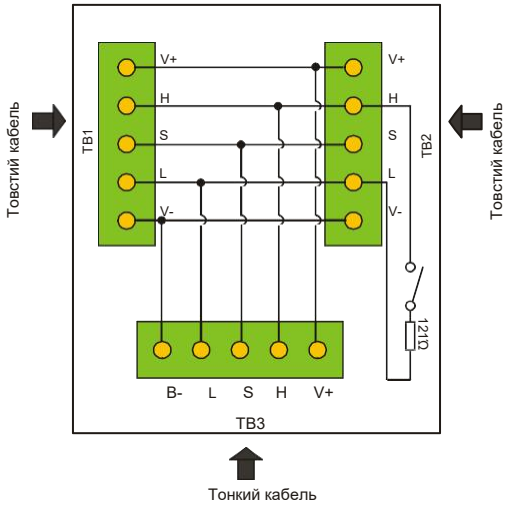
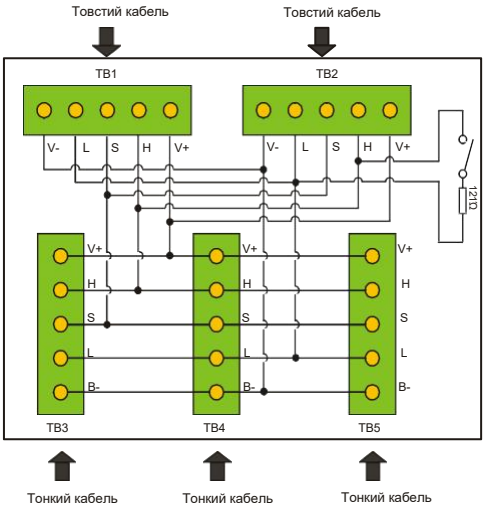
1. Максимальна довжина кабелю для придбання становить 305 M на котушку, а мінімальна довжина – 1 M з метром як одиницею.
2. UC-DN01Z-01A та UC-DN01Z-02A можна використовувати як магістральний кабель, так і кабель відгалуження. Максимальна відстань зв’язку, яку вони підтримують, різна. Максимальна відстань зв’язку, яку підтримують два кабелі при різній швидкості передачі даних CANopen, відображається наступним чином.

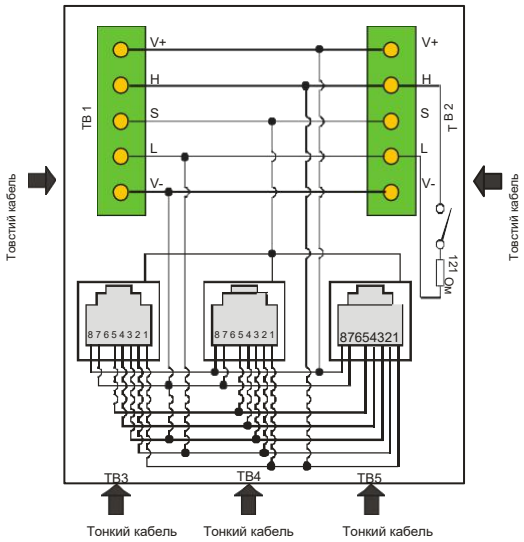
Швидкість передачі CANopen (біт/с)	125 тис	250 тис	500 тис	1M
Макс. відстань зв'язку для UC-DN01Z-01A (м)	500	250	100	40
Макс. відстань зв'язку для UC-DN01Z-02A (м)	100	100	100	40

3. Максимальна відстань зв'язку на швидкості передачі регулюється в протоколі CANopen. В наведеній нижче таблиці показано співвідношення між максимальною відстанню зв'язку та швидкістю передачі.

Швидкість передачі (біт/с)	10 тис	20 тис	50 тис	125 тис	250 тис	500 тис	800 тис	1M
Макс. спілкування відстань (м)	5000	2500	1000	500	250	100	50	40

- **Коробка розподільна**

Модель		Схема фігури
TAP-CN01		
TAP-CN02		

Модель		Схема фігури
TAP-CN03		
Роз'єм	Знімні клеми (5,08 мм)	
Кінцевий резистор	120 Ом	

- **Кінцевий резистор**

  Як пропонується в протоколі CANopen, два кінці кабелю зв'язку CANopen мають підключатися до кінцевого резистора 120 Ом (1/4 Вт) відповідно, щоб узгодити опір сигналу зв'язку та зменшити перешкоди від відбиття сигналу під час нормальної передачі сигналу.

  - Кінцевий резистор, підключений до початку кабелю:

    Кінцевий резистор на розподільній коробці можна використовувати, просто встановивши перемикач кінцевого резистора в положення ON.
  - Кінцевий резистор, підключений до кінцевого кінця кабелю:
    Для підключення до іншого кінця кабелю потрібен кінцевий резистор TAP-TR01.
  - Модель кінцевого резистора: TAP-TR01, значення опору: 120 Ом (1/4 Вт), як показано нижче

TAP-TR01


E.3 Аксесуари для зв’язку DeviceNet

- **Кабель**

малюнок	Модель	Довжина	Діаметр (AWG)
	UC-DN01Z-01A	305M	2#15 , 2#18 SHLD ПВХ (товстий)
	UC-DN01Z-02A	305M	2#22 , 2#24 SHLD PVC (тонкий)

Примітки:

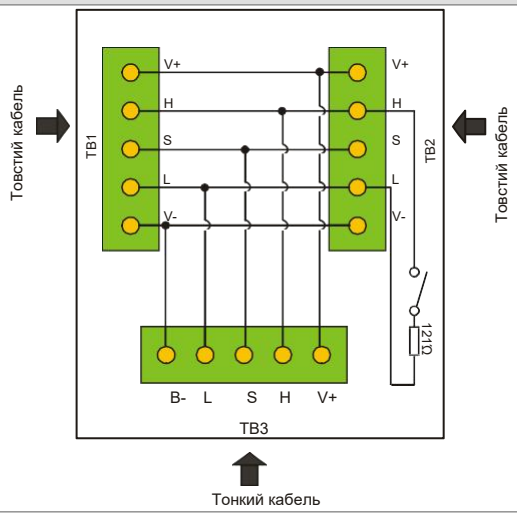
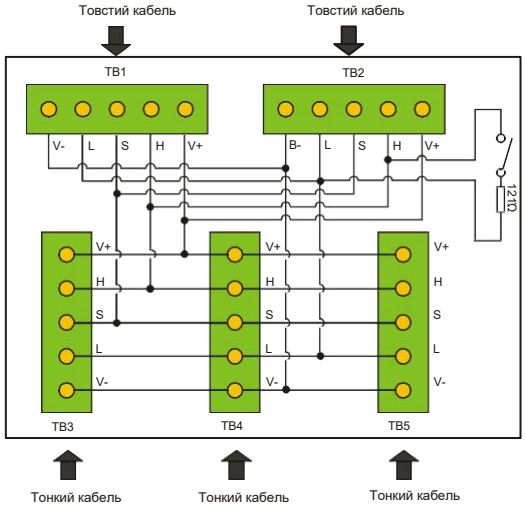
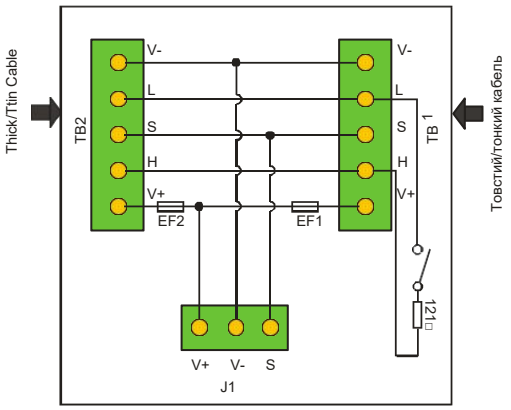
1. Максимальна довжина кабелю для придбання становить 305 M на котушку, а мінімальна довжина – 1 M з метром як одиницею.
2. UC-DN01Z-01A та UC-DN01Z-02A можна використовувати як магістральний кабель, так і кабель відгалуження. Максимальна відстань зв’язку, яку вони підтримують, різна. Максимальна відстань зв’язку, яку підтримують два кабелі при різній швидкості передачі DeviceNet, відображається наступним чином.

Швидкість передачі DeviceNet (біт/с)	125 тис	250 тис	500 тис
Макс. відстань зв'язку для UC- DN01Z-01A (м)	500	250	100
Макс. відстань зв'язку для UC- DN01Z-02A (м)	100	100	100

3. Максимальна відстань зв'язку на швидкості передачі регулюється в протоколі DeviceNet. В наведеній нижче таблиці показано співвідношення між максимальною відстанню зв’язку та швидкістю передачі.

Швидкість передачі (біт/с)	10 тис	20 тис	50 тис	125 тис	250 тис	500 тис
Макс. відстань зв'язку (м)	5000	2500	1000	500	250	100

- **Коробка розподільна**

Модель		Схема фігури
TAP-CN01		
TAP-CN02		
TAP-CP01 (Розподільна коробка)		
Роз'єм	Знімні клеми (5,08 мм)	
Кінцевий резистор	120 Ом	

- **Кінцевий резистор**

  Згідно з вимогами протоколу DeviceNet, два кінці кабелю зв'язку DeviceNet повинні з'єднуватися з кінцевим резистором 120 Ом (1/4 Вт) відповідно.

  1. Кінцевий резистор, підключений до початку кабелю:
     Кінцевий резистор на розподільній коробці можна використовувати, встановивши перемикач кінцевого резистора в положення ON.
  1. Кінцевий резистор, підключений до кінцевого кінця кабелю:
     Для підключення до кінцевого кінця кабелю потрібен кінцевий резистор 120 Ом (1/4 Вт).

Пам'ятка