# PROFINET Mode 13

This chapter provides details for the required parameter settings when the servo communicates with the controller through PROFINET communication.


13.1 Basic configuration ····· 13-3
13.1.1 Hardware configuration ····· 13-3
13.1.2 GSD file import ····· 13-4
13.1.3 Install DriveLib ····· 13-4
13.1.4 Servo firmware version ····· 13-4
13.1.5 Parameter settings of PROFINET mode ····· 13-5
13.2 Communication function ····· 13-7
13.2.1 Specifications ····· 13-7
13.2.2 RT / IRT mode ····· 13-8
13.2.2.1 Real-Time (RT) mode ····· 13-8
13.2.2.2 Isochronous Real-Time (IRT) mode ····· 13-8
13.3 PROFINET application classes ····· 13-9
13.3.1 AC1 ····· 13-9
13.3.2 AC3 ····· 13-9
13.3.3 AC4 ····· 13-9
13.4 Supported telegrams ····· 13-10
13.4.1 Descriptions of PZD (Process Data) ····· 13-12
13.4.2 Control word definition ····· 13-13
STW1 control word 1 (for telegrams 1, 3, 102, 105) ····· 13-13
STW1 control word 1 (for telegram 111) ····· 13-14
POS_STW1 positioning control word 1 (for telegram 111) ····· 13-15
POS_STW2 positioning control word 2 (for telegram 111) ····· 13-15
13.4.3 Status word definition ····· 13-16
ZSW1 status word 1 (for telegrams 1, 3, 102, 105) ····· 13-16
ZSW1 status word 1 (for telegram 111) ····· 13-17
13.5 Using telegrams in PROFINET mode ····· 13-18
13.5.1 Telegram 111 (Profile Position mode) ····· 13-18
Jog / Incremental jogging ····· 13-19
Relative / Absolute positioning ····· 13-20
Positioning as setup ····· 13-21
Homing ····· 13-22

13

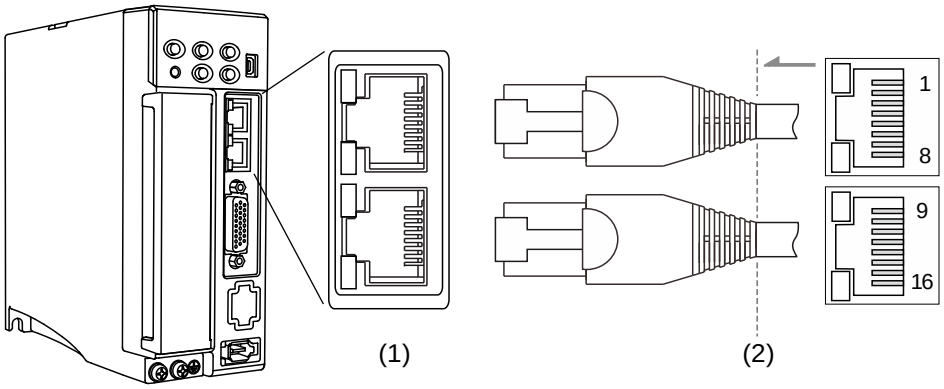
13.5.2 Telegram 1 (Profile Velocity mode) ····· 13-24
Speed control ····· 13-25
13.5.3 Telegrams 3, 102, 105 (Cyclic Synchronous Velocity mode) ····· 13-26
Technology object configuration ····· 13-27
Motion command planning ····· 13-28
13.5.4 Accessing servo parameters ····· 13-29
13.6 SINA function blocks and PNU parameters ····· 13-31
13.6.1 Position control (SINA_POS, FB284) ····· 13-31
13.6.2 Speed control (SINA_SPEED, FB285) ····· 13-34
13.6.3 Acyclic reading / writing (SINA_PARA_S, FB287) ····· 13-36
13.6.4 PNU parameters ····· 13-37
13.7 Troubleshooting ····· 13-50

# 13.1 Basic configuration

## 13.1.1 Hardware configuration



The CN6 connector of the B3A-P models allows you to connect the servo drive to the controller using standard RJ45 connectors and shielded network cables, controlling the position and speed of the motor, as well as accessing or monitoring the servo status with Siemens' PROFINET system.

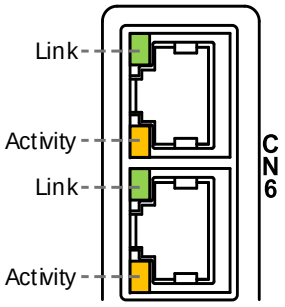



(1) CN6 connector (female); (2) CN6 connector (male)

Pin assignment:

| Pin No. | Signal | Description |
|---|---|---|
| 1, 9 | TX+ | Transmit + |
| 2, 10 | TX- | Transmit - |
| 3, 11 | RX+ | Receive + |
| 4, 12 | - | Reserved |
| 5, 13 | - | Reserved |
| 6, 14 | RX- | Receive - |
| 7, 15 | - | Reserved |
| 8, 16 | - | Reserved |

Description of each indicator for the CN6 connector:




| Name | Color | Status | Description |
|---|---|---|---|
| Link | Green | On | Network is connected. |
| | | Off | No connection or connection error. |
| Activity | Orange | On | Data exchange in progress. |
| | | Off | No data exchange. |

### 13.1.2 GSD file import

The PROFINET motion control fieldbus is an open standard that requires using the GSD (General Station Description) file to configure the functions and related object properties for each slave device. Generally, the GSD file is a standard XML file (GSDML).

**Integration with Siemens SIMATIC S7-1200/1500 controllers**

Import the GSD file of the slave into the Siemens TIA Portal software, so the controller can recognize and control each slave device according to the configuration in the GSD file. Download the GSD file from Delta's website and make sure the file version is V2.41 or above.

Follow these steps to import the GSD file:

1. Open the Siemens TIA Portal software, and then click **Project view** in the lower left of the software screen.
2. In the editing page, go to the toolbar and click **Options** > **Manage general station description files (GSD)**.
3. In the Manage general station description files window, click "**...**" to select the source path, and then the following area shows the GSD file(s) in that source path. Select the check box and then click **Install** to import the GSD file.

After being imported into TIA Portal, the GSD file is stored in the following path:
C:\Users\user_name\Automaiton\project_name\AdditionalFiles\GSD

Note: refer to the controller manufacturer's instruction manuals for the actual storage path.

### 13.1.3 Install DriveLib

Install the DriveLib library in the Siemens TIA Portal software for the controller to access the servo data using the library function blocks. Download the DriveLib file from Siemens' website.

### 13.1.4 Servo firmware version

Make sure the servo firmware version is v3109 sub8387 or above. You can check the firmware version and subversion with the servo parameters P0.000 and P5.000. To update the servo firmware successfully, use the Firmware Update Tool of the version V3.0.0.33.

### 13.1.5 Parameter settings of PROFINET mode

Follow these steps to connect the controller and the servo drive:

1. Set the servo to PROFINET mode by setting P1.001.YX to 0C.
2. Set P3.012.Z to 1 to enable the non-volatile setting for the parameters.

   Note: setting P3.012.Z to 0 is not available in the PROFINET mode.



| Function | P3.012 = 0x0100 (Z = 1) | |
|---|---|---|
| | Servo parameter | Default |
| Motor stop mode | P1.032 | 0x0000 |
| S-curve acceleration constant | P1.034 | 200 |
| Zero speed range | P1.038 | 100 (0.1 rpm) |
| E-Gear ratio - numerator N1 | P1.044 | 16777216 |
| E-Gear ratio - denominator M | P1.045 | 100000 |
| Speed reached (DO.SP_OK) range | P1.047 | 10 (1 rpm) |
| Accumulated time to reach desired speed | P1.049 | 0 |
| Maximum speed limit | P1.055 | Depending on the motor (1 rpm) |
| Excessive deviation warning condition of Position command | P2.035 | 50331648 (pulse) |
| Positive software limit | P5.008 | 2147483647 (PUU) |
| Negative software limit | P5.009 | -2147483648 (PUU) |

Note:

1. The setting of P6.001 (Origin definition) is volatile, so you need to define the origin when executing homing.
2. Unit conversion: PUU = pulse x $\frac{P1.044}{P1.045}$

13

Set the following parameters based on the application requirements.

| P2.121 | Special bit register 6 | | Address: 02E0H 02E1H |
|---|---|---|---|
| Default: | 0x00000000 | Control mode: | All |
| Unit: | - | Setting range: | 0x00000000 - 0x000001FF |
| Format: | HEX | Data size: | 32-bit |

Settings:

| Bit | Function | Description |
|---|---|---|
| Bit 1 | [PROFINET] Behavior after homing in communication mode | 0: after homing, execute absolute positioning to the position with the offset distance set in PNU11.<br>1: decelerate to a stop after homing. |
| Bit 2 | [PROFINET] Definition of the settings for Origin definition (P6.001) and Home offset (PNU11) in communication mode | 0: origin definition (P6.001) = - (setting of PNU11)<br>1: origin definition (P6.001) = PNU11 |
| Bit 3 | [PROFINET] Unit of Homing speeds (PNU12, PNU13) in communication mode | 0: 0.1 rpm<br>1: 1 rpm |
| Bit 5 | [PROFINET] Unit selection for Homing speeds (PNU12, PNU13), Homing acceleration (PNU14), Profile acceleration (PNU15), and Profile deceleration (PNU16) in communication mode | 0: the unit of PNU12 and PNU13 is determined by the setting of P2.121 [Bit 3]; the unit of PNU14, PNU15, and PNU16 is ms (0 - 3000 rpm)<br>When P2.121 [Bit 3] = 0, the unit of PNU12 and PNU13 is 0.1 rpm.<br>When P2.121 [Bit 3] = 1, the unit of PNU12 and PNU13 is 1 rpm.<br>1: the unit of PNU12 and PNU13 is PUU/sec; the unit of PNU14, PNU15, and PNU16 is PUU/sec$^2$. |

| P3.011 | CANopen / DMCNET / PROFINET options | | Address: 0316H 0317H |
|---|---|---|---|
| Default: | 0x0000 | Control mode: | CANopen / DMCNET / PROFINET |
| Unit: | - | Setting range: | Shown as follows |
| Format: | HEX | Data size: | 16-bit |

Settings:




| | | | |
|---|---|---|---|
| X | Store parameters in EEPROM or not | Z | Reserved |
| Y | Reserved | U | Reserved |

- X: store parameters in EEPROM or not

  0: not to store parameters in EEPROM.

  1: when writing parameters with packets through cyclic synchronous communication, store parameters in EEPROM.

  Note: if you set X to 1 and continuously write parameters with packets through cyclic synchronous communication, it shortens the lifetime of the EEPROM.

# 13.2 Communication function

## 13.2.1 Specifications

| | |
|---|---|
| Physical layer | 100BASE-TX |
| Communication interface | RJ45 x 2 |
| Network protocol | Serial connection |
| Baud rate | 2 x 100 Mbps (full duplex) |
| Transmission distance | When multiple servo drives are connected, the maximum distance between each drive is 50 m (164.04 ft). |
| Transmission cable | CAT5e STP cables with metal connectors |
| Number of controllable slaves | Maximum 65,535 slaves; the actual number is determined by the controller. |
| Data frame size | Maximum 1,440 bytes |
| RT / IRT mode | IRT mode (synchronous): 1 ms<br>RT mode (asynchronous): 1 ms |
| Communication service | Cyclic I/O data transmission<br>Acyclic I/O data transmission |
| Application layer | Meets the definitions in the PROFIdrive application profile, supporting the application classes AC1, AC3, and AC4. |
| Supported topology | ■ Line topology<br>■ Star topology<br>■ Ring topology<br>■ Tree topology<br>■ Hybrid topology |

### 13.2.2 RT / IRT mode

PROFINET communication supports two types of real-time data transmission: Real-Time (RT) mode and Isochronous Real-Time (IRT) mode.

#### 13.2.2.1 Real-Time (RT) mode

The master and slave(s) run asynchronously in the RT mode. The slave clock runs independently of the master clock. That is, the clocks are not synchronized. The command and feedback between the master and slave(s) are transmitted sequentially rather than synchronously.

#### 13.2.2.2 Isochronous Real-Time (IRT) mode

There is precise time synchronization between the master and slave(s) in the IRT mode. The master executes the control program and sends packets at a fixed time cyclically according to the synchronization clock, transmitting the command to and receiving the feedback from the slave(s). The slave(s) receives and updates the data at a fixed time according to the synchronization clock.

Note: in IRT mode, the Topology view in the Siemens TIA Portal software must be configured according to the actual wiring.

## 13.3 PROFINET application classes

Among the PROFINET application profiles, PROFIdrive is applicable to motion control. PROFIdrive is the standard profile for drive control on the data exchange between the controller and the servo drive. The PROFIdrive profile defines 6 application classes, AC1 to AC6, and the B3A-P model is currently applicable to AC1, AC3, and AC4.



### 13.3.1 AC1

In AC1, the servo drive is in Profile Velocity mode, where the motion planning is done using telegram 1 of the controller. The controller specifies the speed command and sets the acceleration / deceleration conditions, and then the trajectory generator in the servo drive plans the motion path according to these conditions. This mode is suitable for applications that do not require high responsiveness.

### 13.3.2 AC3

In AC3, the servo drive is in Profile Position mode, where the motion planning is done using telegram 111 of the controller. After receiving the position command from the controller, the servo drive controls the servo motor to reach the target position.
In Profile Position mode, the controller only informs the servo drive of the target position, speed command, and acceleration / deceleration settings at the beginning. The motion planning from command triggering to the arrival of the target position is performed by the trajectory generator in the servo drive. This mode is suitable for applications of single-axis positioning.

### 13.3.3 AC4

In AC4, the servo drive is in the Cyclic Synchronous Velocity mode, where the motion planning is done using telegram 3, 102, or 105 of the controller. The controller transmits the motion command to the servo drive cyclically. This mode is suitable for applications of multi-axis synchronization, motion path planning, and interpolation control.

## 13.4 Supported telegrams



1. Directly change the telegram from the controller. Setting the servo parameters additionally is not necessary.
2. If you **change** the telegram when power is supplied to the servo, cycle power to the servo drive to have the change take effect.
3. This servo drive supports the following standard telegrams. Make sure the controller supports these telegrams before using them.

| Telegram | RT / IRT mode | Maximum number of PZD | |
|---|---|---|---|
| | | Receive word | Send word |
| Telegram 1 | RT | 2 | 2 |
| Telegram 3 | IRT / RT | 4 | 9 |
| Telegram 102 | IRT / RT | 6 | 10 |
| Telegram 105 | IRT | 10 | 10 |
| Telegram 111 | RT | 12 | 12 |

Note:

1. PZD is the process data for cyclic data transmission; one PZD is one word (16-bit).
2. Telegrams 3 and 102 currently do not support RT mode.

Telegrams used for speed control:

| Telegram | 1 | | 3 | | 102 | | 105 | |
|---|---|---|---|---|---|---|---|---|
| Application Class | AC1 | | AC4 | | AC4 | | AC4 | |
| PZD1 | STW1 | ZSW1 | STW1 | ZSW1 | STW1 | ZSW1 | STW1 | ZSW1 |
| PZD2 | NSOLL_A | NIST_A | NSOLL_B | NIST_B | NSOLL_B | NIST_B | NSOLL_B | NIST_B |
| PZD3 | Receive telegram from PROFINET | Send telegram to PROFINET | | | | | | |
| PZD4 | | | STW2 | ZSW2 | STW2 | ZSW2 | STW2 | ZSW2 |
| PZD5 | | | G1_STW | G1_ZSW | MOMRED | MELDW | MOMRED | MELDW |
| PZD6 | | | - | G1_XIST1 | G1_STW | G1_ZSW | G1_STW | G1_ZSW |
| PZD7 | | | | | - | G1_XIST1 | XERR | G1_XIST1 |
| PZD8 | | | | G1_XIST2 | | | | |
| PZD9 | | | | | | G1_XIST2 | KPC | G1_XIST2 |
| PZD10 | | | | - | | | | |

Telegrams used for position control:

| Telegram | 111 | |
|---|---|---|
| Application Class | AC3 | |
| PZD1 | STW1 | ZSW1 |
| PZD2 | POS_STW1 | POS_ZSW1 |
| PZD3 | POS_STW2 | POS_ZSW2 |
| PZD4 | STW2 | ZSW2 |
| PZD5 | OVERRIDE | MELDW |
| PZD6 | MDI_TARPOS | XIST_A |
| PZD7 | | |
| PZD8 | MDI_VELOCITY | NIST_B |
| PZD9 | | |
| PZD10 | MDI_ACC | FAULT_CODE |
| PZD11 | MDI_DEC | WARN_CODE |
| PZD12 | User | User |

Note: the PZDs in gray are not yet supported.

13

### 13.4.1 Descriptions of PZD (Process Data)

The following describes the PZDs of telegrams 1, 3, 102, 105, and 111.

| PZD | Description | Received word / Send word | Data type |
|---|---|---|---|
| STW1 | Control word 1 | Receive word | U16 |
| STW2 | Control word 2 | Receive word | U16 |
| NSOLL_A | Speed setpoint A (16-bit) | Receive word | I16 |
| NSOLL_B | Speed setpoint B (32-bit) | Receive word | I32 |
| MDI_TARPOS | MDI position | Receive word | I32 |
| MDI_VELOCITY | MDI velocity | Receive word | I32 |
| OVERRIDE | Position velocity override | Receive word | I16 |
| MDI_ACC | MDI acceleration override | Receive word | I16 |
| MDI_DEC | MDI deceleration override | Receive word | I16 |
| POS_STW1 | Positioning control word 1 | Receive word | U16 |
| POS_STW2 | Positioning control word 2 | Receive word | U16 |
| G1_XIST2 | Actual position 2 | Send word | U32 |
| ZSW1 | Status word 1 | Send word | U16 |
| ZSW2 | Status word 2 | Send word | U16 |
| NIST_A | Speed actual value A (16-bit) | Send word | I16 |
| NIST_B | Speed actual value B (32-bit) | Send word | I32 |
| XIST_A | Position actual value A | Send word | I32 |
| FAULT_CODE | Alarm code | Send word | U16 |
| WARN_CODE | Warning code | Send word | U16 |

## 13.4.2 Control word definition

### STW1 control word 1 (for telegrams 1, 3, 102, 105)

Important: STW1.10 must be set to 1 first for the servo to allow requests from the controller.



| Signal | Function | Description |
|---|---|---|
| STW1.0 | Servo On / Off | 0: Servo Off<br>↑: Servo On |
| STW1.1 | Free-run stop<br>(AL013 triggering; OFF2) | 0: trigger servo alarm AL013<br>1: when STW1.2 is also set to 1, the servo can be switched to On |
| STW1.2 | PFQS<br>(AL35F triggering; OFF3) | 0: trigger servo alarm AL35F<br>1: when STW1.1 is also set to 1, the servo can be switched to On |
| STW1.3 | Enable / disable operation | 0: disable operation<br>1: enable operation |
| STW1.4 | Quick stop | 0: servo is in the Quick stop state<br>1: disable the function |
| STW1.5 | Halt | 0: halt the command; once the function is disabled, the operation continues until the command is complete<br>1: disable the function |
| STW1.6 | Trigger | 0: disable the function<br>↑: trigger the command |
| STW1.7 | Fault reset | 0: disable the function<br>↑: reset the servo alarm |
| STW1.8 to 1.9 | Reserved | - |
| STW1.10 | Servo allows requests from controller | 0: servo does not allow requests from the controller<br>1: servo allows requests from the controller |
| STW1.11 to 1.15 | Reserved | - |

13

### STW1 control word 1 (for telegram 111)

Important: STW1.10 must be set to 1 first for the servo to allow requests from the controller.

| Signal | Function | Description |
|---|---|---|
| STW1.0 | Servo On / Off | 0: Servo Off<br>↑: Servo On |
| STW1.1 | Free-run stop<br>(AL013 triggering; OFF2) | 0: trigger servo alarm AL013<br>1: when STW1.2 is also set to 1, the servo can be switched to On |
| STW1.2 | PFQS<br>(AL35F triggering; OFF3) | 0: trigger servo alarm AL35F<br>1: when STW1.1 is also set to 1, the servo can be switched to On |
| STW1.3 | Enable / disable operation | 0: disable operation<br>1: enable operation |
| STW1.4 | Reject | 0: reject the command; the command must be issued again<br>1: disable the function |
| STW1.5 | Halt | 0: halt the command; once the function is disabled, the operation continues until the command is complete<br>1: disable the function |
| STW1.6 | Trigger | 0: disable the function<br>↑: trigger the command |
| STW1.7 | Fault reset | 0: disable the function<br>↑: reset the servo alarm |
| STW1.8 to 1.9 | Jog | 00: disable the function<br>01: jog in forward direction (Jog1)<br>10: jog in reverse direction (Jog2)<br>11: disable the function |
| STW1.10 | Servo allows requests from controller | 0: servo does not allow requests from the controller<br>1: servo allows requests from the controller |
| STW1.11 | Homing | 0: disable the function<br>1: start homing |
| STW1.12 to 1.15 | Reserved | - |

## POS_STW1 positioning control word 1 (for telegram 111)

| Signal | Function | Description | |
|---|---|---|---|
| POS_STW1.0 to 1.7 | Reserved | - | |
| POS_STW1.8 | MDI positioning mode | 0: relative positioning<br>1: absolute positioning | |
| POS_STW1.9 | MDI direction selection | When ModePos = 2 (absolute positioning)*1:<br>01: positioning command in forward direction<br>10: positioning command in reverse direction | When ModePos = 3 (positioning as setup)*2:<br>00: stop operation<br>01: forward operation<br>10: reverse operation<br>11: stop operation |
| POS_STW1.10 | | | |
| POS_STW1.11 to 1.13 | Reserved | - | |
| POS_STW1.14 | Working mode | 0: signal positioning<br>1: signal setting-up | |
| POS_STW1.15 | MDI selection | 0: disable the MDI function<br>1: enable the MDI function | |

Note:

1. When STW1.8, STW1.9, STW1.11, POS_STW1.8, and POS_STW1.14 are all set to 0, ModePos = 2.
2. When STW1.8, STW1.9, and STW1.11 are set to 0, and POS_STW1.8 and POS_STW1.14 are set to 1, ModePos = 3.
3. The MDI (Manual Data Input) positioning function enables the controller to inform the servo of the target position, speed, and acceleration / deceleration for the servo to perform the calculation.

## POS_STW2 positioning control word 2 (for telegram 111)

| Signal | Function | Description |
|---|---|---|
| POS_STW2.0 | Reserved | - |
| POS_STW2.1 | Set current feedback position as origin | 0: disable the function<br>1: set the current feedback position as the origin |
| POS_STW2.2 to 2.4 | Reserved | - |
| POS_STW2.5 | Jog mode setting | 0: incremental jogging (ModePos = 8)<br>1: jog (ModePos = 7) |
| POS_STW2.6 to 2.13 | Reserved | - |
| POS_STW2.14 | Software limit switch (ConfigEPos [Bit 2]) | 0: disable the function<br>1: enable the software limit switch |
| POS_STW2.15 | Hardware limit switch (ConfigEPos [Bit 3]) | 0: disable the function<br>1: enable the hardware limit switch |

13

### 13.4.3 Status word definition

#### ZSW1 status word 1 (for telegrams 1, 3, 102, 105)

| Signal | Function | Description |
|---|---|---|
| ZSW1.0 | Switched On | 0: not ready for Servo On<br>1: ready for Servo On |
| ZSW1.1 | Ready for operation | 0: operation disabled<br>1: operation enabled |
| ZSW1.2 | Operation enabled | 0: Servo Off<br>1: Servo On |
| ZSW1.3 | Fault present | 0: no servo alarm (ALM)<br>1: servo alarm (ALM) occurs |
| ZSW1.4 | Free-run stop<br>(AL013 triggering; OFF2) | 0: servo alarm AL013 triggered<br>1: servo alarm AL013 cleared |
| ZSW1.5 | PFQS<br>(AL35F triggering; OFF3) | 0: servo alarm AL35F triggered<br>1: servo alarm AL35F cleared |
| ZSW1.6 | Switching on inhibited | 0: operation allowed<br>1: operation inhibited |
| ZSW1.7 | Warning present | 0: servo warning triggered<br>1: servo warning cleared |
| ZSW1.8 | Speed deviation within tolerance | 0: speed deviation exceeded tolerance<br>1: speed deviation within tolerance |
| ZSW1.9 | Control requested | 0: controller request not allowed<br>1: controller request allowed |
| ZSW1.10 to 1.15 | Reserved | - |

### ZSW1 status word 1 (for telegram 111)

| Signal | Function | Description |
|---|---|---|
| ZSW1.0 | Switched On | 0: not ready for Servo On<br>1: ready for Servo On |
| ZSW1.1 | Ready for operation | 0: operation disabled<br>1: operation enabled |
| ZSW1.2 | Operation enabled | 0: Servo Off<br>1: Servo On |
| ZSW1.3 | Fault present | 0: no servo alarm (ALM)<br>1: servo alarm (ALM) occurs |
| ZSW1.4 | Free-run stop<br>(AL013 triggering; OFF2) | 0: servo alarm AL013 triggered<br>1: servo alarm AL013 cleared |
| ZSW1.5 | PFQS<br>(AL35F triggering; OFF3) | 0: servo alarm AL35F triggered<br>1: servo alarm AL35F cleared |
| ZSW1.6 | Switching on inhibited | 0: operation allowed<br>1: operation inhibited |
| ZSW1.7 | Warning present | 0: servo warning triggered<br>1: servo warning cleared |
| ZSW1.8 | Speed deviation within tolerance | 0: speed deviation exceeded tolerance<br>1: speed deviation within tolerance |
| ZSW1.9 | Control requested | 0: controller request not allowed<br>1: controller request allowed |
| ZSW1.10 | Target reached | 0: target not reached<br>1: target reached |
| ZSW1.11 | Homing complete | 0: homing not complete<br>1: homing complete |
| ZSW1.12 to 1.15 | Reserved | - |

## 13.5 Using telegrams in PROFINET mode

This section describes the use of telegrams in PROFINET mode. The content includes basic operation settings and related object descriptions.

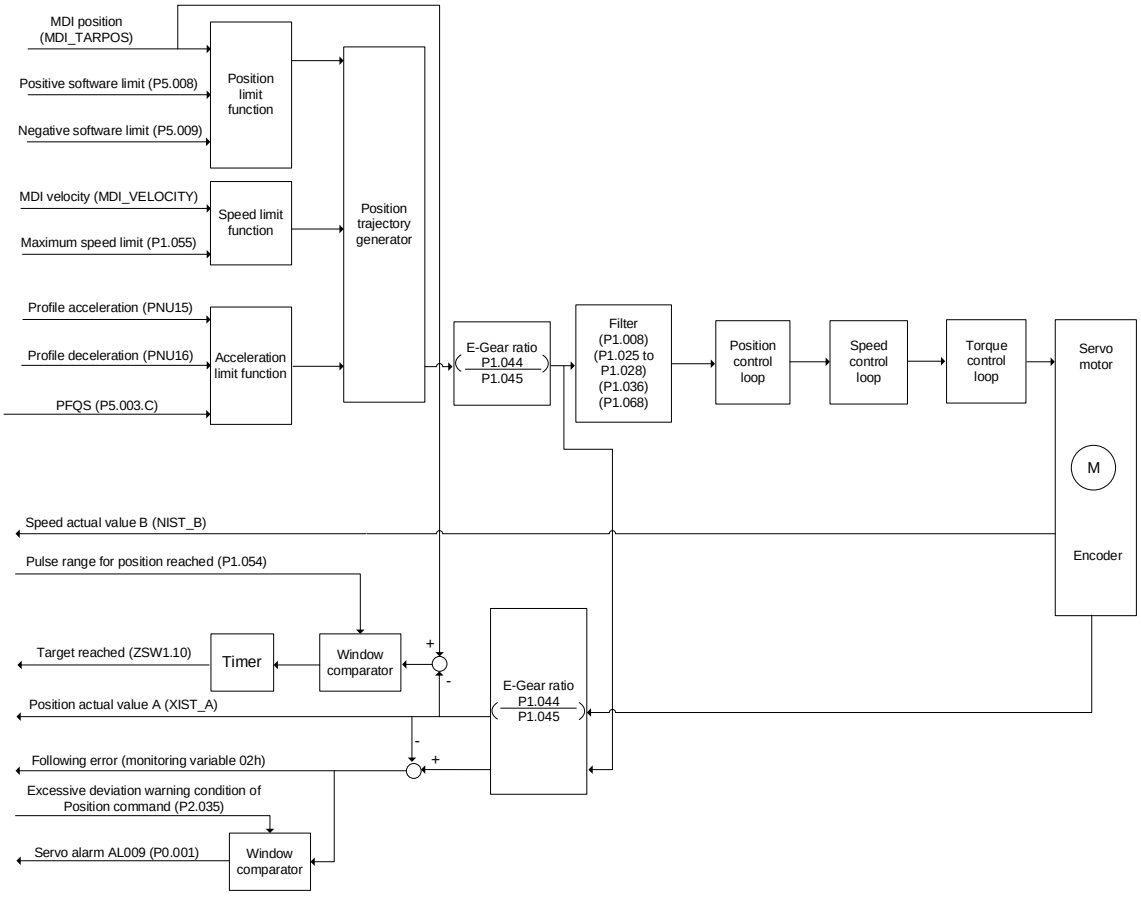
13

### 13.5.1 Telegram 111 (Profile Position mode)

After receiving the position command from the controller, the servo drive controls the servo motor to reach the target position.

When telegram 111 is used, the controller only informs the servo drive of the target position, speed command, and acceleration / deceleration settings at the beginning. The motion planning from command triggering to the arrival of the target position is performed by the trajectory generator in the servo drive.

The following figure shows the Profile Position mode architecture of the servo drive when telegram 111 is used:




Using the function blocks in the DriveLib of TIA Portal can achieve the basic positioning control (SINA_POS, FB284) and access the PNU and servo parameters (SINA_PARA_S, FB287). The following are the examples of using these function blocks.

Note: refer to Section 13.6 for the descriptions of function block pins.

### Jog / Incremental jogging

Operating condition: for FB284, the CancelTraversing and IntermediateStop pins must be set to 1 (disable the functions).

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Select the jog mode | FB284 | ModePos | 7 | 7: jog; continuous triggering (in units of 5000 PUU/s)<br>8: incremental jogging; rising-edge triggering (in units of 1000 PUU) |
| 2. Set the servo to On | FB284 | EnableAxis | 1 | 0: Servo Off<br>1: Servo On |
| 3. Execute jog operation | | | | Jog1 and Jog2 cannot be set to 1 simultaneously. |
| ◆ Execute jog in forward direction | FB284 | Jog1 | 1 | 0: disable the function<br>1: jog in forward direction |
| ◆ Execute jog in reverse direction | FB284 | Jog2 | 1 | 0: disable the function<br>1: jog in reverse direction |
| 4. Set jog speed override | FB284 | OverV | - | Setting range: 1 - 199% |

Note: changing the speed and position commands for the jog operation is not yet supported.

### Read the servo data:

| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the motor feedback position at present | FB284 | ActPosition | - | Unit: PUU |
| 2. Read the motor feedback speed at present | FB284 | ActVelocity | - | Unit: 16#40000000h indicates 100% of the motor rated speed<br>For example, when the read value is 16#60000000h, the feedback speed is the rated speed*150%. |

13

### Relative / Absolute positioning

Operating condition: for FB284, the CancelTraversing and IntermediateStop pins must be set to 1 (disable the functions), and the Jog1 and Jog2 pins must be set to 0 (disable the function).

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Select the positioning function | FB284 | ModePos | 2 | 1: relative positioning<br>2: absolute positioning |
| 2. Set the target position | FB284 | Position | - | Unit: PUU |
| 3. Set the speed command | FB284 | Velocity | - | Unit: 1000 PUU/min |
| 4. Set the Profile acceleration (PNU15) | | | | |
| 4.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 4.2 Select PNU parameter | FB287 | Parameter | 15 | 15: Profile acceleration |
| 4.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for acceleration from 0 rpm to 3000 rpm |
| 4.4 Start | FB287 | Start | ⤒ | 0: disable the function<br>⤒: start accessing |
| 5. Set the Profile deceleration (PNU16) | | | | |
| 5.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 5.2 Select PNU parameter | FB287 | Parameter | 16 | 16: Profile deceleration |
| 5.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for deceleration from 3000 rpm to 0 rpm |
| 5.4 Start | FB287 | Start | ⤒ | 0: disable the function<br>⤒: start accessing |
| 6. Set the servo to On | FB284 | EnableAxis | 1 | 0: Servo Off<br>1: Servo On |
| 7. Execute the command | FB284 | ExecuteMode | ⤒ | 0: disable the function<br>⤒: execute commands of signal positioning and signal setting-up |

Note: PNU15 (Profile acceleration) and PNU16 (Profile deceleration time) are volatile parameters.

### Read the servo data:

| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the motor feedback position at present | FB284 | ActPosition | - | Unit: PUU |
| 2. Read the motor feedback speed at present | FB284 | ActVelocity | - | Unit: 16#40000000h indicates 100% of the motor rated speed<br>For example, when the read value is 16#60000000h, the feedback speed is the rated speed*150%. |

### Command to take immediate effect

When FB284 - ModePos = 2 (absolute positioning), you can have the command take effect immediately or not with the setting of FB284 - ConfigEPos [Bit 8].

### Positioning as setup

Operating condition: for FB284, the CancelTraversing and IntermediateStop pins must be set to 1 (disable the functions), and the Jog1 and Jog2 pins must be set to 0 (disable the function).

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Select the mode of positioning as setup | FB284 | ModePos | 3 | 3: positioning as setup |
| 2. Select the rotation direction | | | | Positive and Negative cannot be set to 1 simultaneously. |
| ◆ Set to operate in positive direction | FB284 | Positive | 0 | 0: disable the function<br>1: operate in positive direction |
| ◆ Set to operate in negative direction | FB284 | Negative | 1 | 0: disable the function<br>1: operate in negative direction |
| 3. Set the speed command | FB284 | Velocity | - | Unit: 1000 PUU/min |
| 4. Set the Profile acceleration (PNU15) | | | | |
| 4.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 4.2 Select PNU parameter | FB287 | Parameter | 15 | 15: Profile acceleration |
| 4.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for acceleration from 0 rpm to 3000 rpm |
| 4.4 Start | FB287 | Start | ⊓↑ | 0: disable the function<br>⊓↑: start accessing |
| 5. Set the Profile deceleration (PNU16) | | | | |
| 5.1 Enable the writing function for PNU parameters | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 5.2 Select PNU parameter | FB287 | Parameter | 16 | 16: Profile deceleration |
| 5.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for deceleration from 3000 rpm to 0 rpm |
| 5.4 Start | FB287 | Start | ⊓↑ | 0: disable the function<br>⊓↑: start accessing |
| 6. Set the servo to On | FB284 | EnableAxis | 1 | 0: Servo Off<br>1: Servo On |
| 7. Execute the command | FB284 | ExecuteMode | ⊓↑ | 0: disable the function<br>⊓↑: execute commands of signal positioning and signal setting-up |

Note: PNU15 (Profile acceleration) and PNU16 (Profile deceleration) are volatile parameters.

### Read the servo data:

| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the motor feedback position at present | FB284 | ActPosition | - | Unit: PUU |
| 2. Read the motor feedback speed at present | FB284 | ActVelocity | - | Unit: 16#40000000h indicates 100% of the motor rated speed<br>For example, when the read value is 16#60000000h, the feedback speed is the rated speed*150%. |

#### Command to take immediate effect

When FB284 - ModePos = 2 (absolute positioning), you can have the command take effect immediately or not with the setting of FB284 - ConfigEPos [Bit 8].

## Homing

Operating condition: for FB284, the CancelTraversing and IntermediateStop pins must be set to 1 (disable the functions), and the Jog1 and Jog2 pins must be set to 0 (disable the function).

| Step | FB | Pin | Setting value | Description |
|---|---|---|---|---|
| 1. Select the homing mode | FB284 | ModePos | 4 | 4: homing mode<br>5: regard the current feedback position as the origin<br>Note: directly go to Step 8 when setting ModePos to 5. |
| 2. Set the Home offset (PNU11) | | | | |
| 2.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 2.2 Select PNU parameter | FB287 | Parameter | 11 | 11: Home offset |
| 2.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: PUU |
| 2.4 Start | FB287 | Start | ↑ | 0: disable the function<br>↑: start accessing |
| 3. Set the Homing method (PNU10) | | | | |
| 3.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 3.2 Select PNU parameter | FB287 | Parameter | 10 | 10: Homing method |
| 3.3 Set the value to write | FB287 | ValueWrite2 | - | Refer to the description of PNU10 |
| 3.4 Start | FB287 | Start | ↑ | 0: disable the function<br>↑: start accessing |
| 4. Set Homing speeds 1 (PNU12) | | | | |
| 4.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 4.2 Select PNU parameter | FB287 | Parameter | 12 | 12: Homing speeds 1 |
| 4.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: 0.1 rpm |
| 4.4 Start | FB287 | Start | ↑ | 0: disable the function<br>↑: start accessing |
| 5. Set Homing speeds 2 (PNU13) | | | | |
| 5.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 5.2 Select PNU parameter | FB287 | Parameter | 13 | 13: Homing speeds 2 |
| 5.3 Set the value to write | FB287 | ValueWrite2 | | Unit: 0.1 rpm |
| 5.4 Start | FB287 | Start | ↑ | 0: disable the function<br>↑: start accessing |
| 6. Set Homing acceleration (PNU14) | | | | |
| 6.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 6.2 Select PNU parameter | FB287 | Parameter | 14 | 14: Homing acceleration |
| 6.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for acceleration from 0 rpm to 3000 rpm and deceleration from 3000 rpm to 0 rpm |
| 6.4 Start | FB287 | Start | ↑ | 0: disable the function<br>↑: start accessing |
| 7. Set the servo to On | FB284 | EnableAxis | 1 | 0: Servo Off<br>1: Servo On |
| 8. Execute the command | FB284 | ExecuteMode | ↑ | 0: disable the function<br>↑: command execution |

Note: PNU10 (Homing method), PNU11 (Home offset), PNU12 (Homing speeds 1), PNU13 (Homing speeds 2), and PNU14 (Homing acceleration) are volatile parameters.

**Read the servo data:**

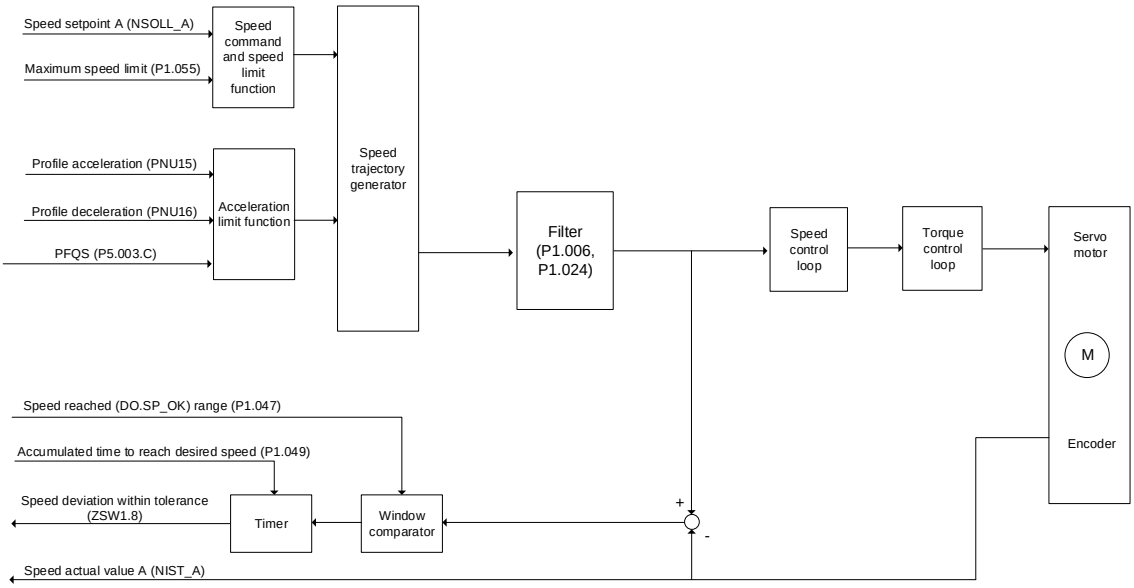
| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the motor feedback position at present | FB284 | ActPosition | - | Unit: PUU |
| 2. Read the motor feedback speed at present | FB284 | ActVelocity | - | Unit: 16#40000000h indicates 100% of the motor rated speed<br>For example, when the read value is 16#60000000h, the feedback speed is the rated speed*150%. |

## 13.5.2 Telegram 1 (Profile Velocity mode)

13

When telegram 1 is used, the controller specifies the speed command and sets the acceleration / deceleration conditions, and then the trajectory generator in the servo drive plans the motion path according to these conditions.

The following figure shows the Profile Velocity mode architecture of the servo drive when telegram 1 is used:




Using the function blocks in the DriveLib of TIA Portal can achieve the basic speed control (SINA_SPEED, FB285) and access the PNU and servo parameters (SINA_PARA_S, FB287). The following is an example of using these function blocks.

Note: refer to Section 13.6 for the descriptions of function block pins.

## Speed control

Operating condition: for FB285, the ConfigAxis pin must be set to 16#003F (default).

| Step | FB | Pin | Setting value | Description |
|---|---|---|---|---|
| 1. Set the rated speed | FB285 | RefSpeed | - | Set the speed in units of rpm according to the motor specification. |
| 2. Set the speed setpoint | FB285 | SpeedSp | 100 | Unit: rpm |
| 3. Set the Profile acceleration (PNU15) | | | | |
| 3.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 3.2 Select PNU parameter | FB287 | Parameter | 15 | 15: Profile acceleration |
| 3.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for acceleration from 0 rpm to 3000 rpm |
| 3.4 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 4. Set the Profile deceleration (PNU16) | | | | |
| 4.1 Enable the writing function for the PNU parameter | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 4.2 Select PNU parameter | FB287 | Parameter | 16 | 16: Profile deceleration |
| 4.3 Set the value to write | FB287 | ValueWrite2 | - | Unit: ms; the time for deceleration from 3000 rpm to 0 rpm |
| 4.4 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 5. Set the servo to On | FB285 | EnableAxis | 1 | 0: Servo Off<br>1: Servo On |

Note: PNU15 (Profile acceleration) and PNU16 (Profile deceleration) are volatile parameters.

### Read the servo data:

| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the motor feedback speed at present | FB285 | ActVelocity | - | Unit: rpm |

### Command to take effect

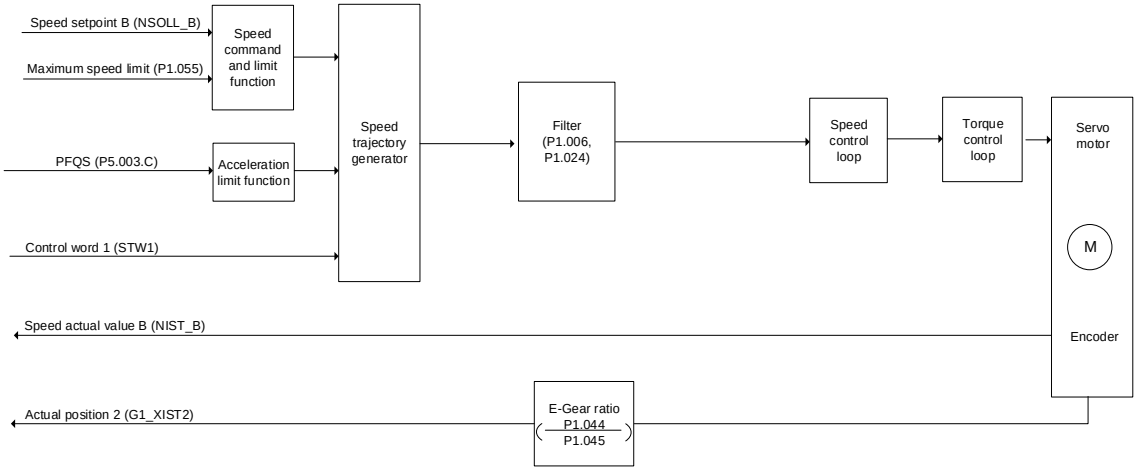
When telegram 1 is used, set FB285 - ConfigAxis [Bit 5] to have the command take effect or not.

13

### 13.5.3 Telegrams 3, 102, 105 (Cyclic Synchronous Velocity mode)

When telegram 3 or 102 is used, the controller performs calculation for position control, and the servo drive is in charge of speed control. In this mode, when the controller transmits the PZD cyclically, it transmits the Speed setpoint B (NSOLL_B) and Control word 1 (STW1) data to the servo at the same time. Currently, telegram 102 does not support the torque limit setting, and telegram 105 does not support the torque limit setting and dynamic servo control (DSC).

The following figure shows the Cyclic Synchronous Velocity mode architecture of the servo drive when telegram 3, 102, or 105 is used:




In TIA Portal, you can set the Technology objects and MC function blocks for telegrams 3, 102, and 105 to perform motion control.

Note:
1. Only supports the Motion Control technology objects of V4.0.
2. Telegrams 3, 102, and 105 currently only support incremental encoders.
3. When data is exchanged with the drive, the function of **Automatically apply drive values at runtime** is not available.
4. When data is exchanged with the encoder, the function of **Automatically apply encoder values at runtime** is not available. The **Increments per revolution** must be set to 2048 and the **Bits in incr. actual value (Gx_XIST1)** must be set to 13.

### Technology object configuration

**Example:**

1. Add new object → Add a new axis
2. Set the axis: Configuration → Hardware interface

   (1) Drive

   - Drive type: PROFIdrive
   - Data connection: Drive
   - Drive: [Drive device name].PROFIdrive Modul AC1,3,4

   (2) Encoder

   - Data connection: Encoder
   - Encoder: [Drive device name].PROFIdrive Modul AC1,3,4
   - Encoder type: Incremental

   Note: in the Technology object configuration, the Encoder type currently only supports “Incremental”.

   (3) Data exchange with the drive

   - Drive telegram: select telegram 3, 102, or 105
   - Reference speed: corresponds to the rated speed of the rotary motor
   - Maximum speed: corresponds to the maximum speed of the rotary motor

   Note: the functions of **Automatically apply drive values at runtime** and **Torque data** are currently unavailable.

   (4) Data exchange with encoder

   - Encoder telegram: select telegram 3, 102, or 105
   - Measuring system: Rotary
   - Increments per revolution: 2048
   - Bits in incr. actual value (Gx_XIST1): 13

   Note: the function of **Automatically apply encoder values at runtime** is currently unavailable.

13

### Motion command planning

You can add MC function blocks as required. The following briefly introduces the MC function blocks for planning the motion commands. Refer to the controller manual for detailed descriptions.

- MC_Power: enable or disable the technology object.
- MC_Reset: acknowledge the alarm and restart the technology object.
- MC_Home: execute homing for the technology object.
- MC_Halt: pause the axis.
- MC_MoveAbsolute: perform absolute positioning of the axis.
- MC_MoveRelative: perform relative positioning of the axis.
- MC_MoveVelocity: move the axis at a constant velocity.
- MC_MoveJog: move the axis with jog operation.
- MC_MoveSuperimposed: start a relative positioning command which is superimposed on the current positioning command.

**Example of using the absolute positioning command:**

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Enable or disable the technology object | | | | |
| 1.1 Set the axis | MC_Power | Axis | - | Axis name of the technology object |
| 1.2 Set the servo to On | MC_Power | Enable | 1 | 0: Servo Off<br>1: Servo On |
| 2. Execute homing for the technology object | | | | |
| 2.1 Set the axis | MC_Home | Axis | - | Axis name of the technology object |
| 2.2 Operating mode | MC_Home | Mode | - | Set the homing method according to the controller |
| 2.3 Start | MC_Home | Execute |  | 0: disable<br>: trigger the command |
| 3. Perform absolute positioning of the axis | | | - | |
| 3.1 Set the axis | MC_Move Absolute | Axis | - | Axis name of the technology object |
| 3.2 Velocity setting | MC_Move Absolute | Velocity | - | Unit: the unit set by the controller/sec |
| 3.3 Absolute target position | MC_Move Absolute | Position | - | Unit: the unit set by the controller |
| 3.4 Start | MC_Move Absolute | Execute |  | 0: disable<br>: trigger the command |

### 13.5.4 Accessing servo parameters

When using telegram 1, 3, 102, 105, or 111, the controller can access the servo parameters by using the function block SINA_PARA_S (FB287).



**Operation steps:**

Access the servo parameters with PNU20 - PNU22 by following these steps.

Example 1: set the servo parameter P1.044 to 16777216

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Enable the writing function for parameters | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 2. Set the servo parameter group | | | | |
| 2.1 Select P parameter group | FB287 | Parameter | 20 | 20: P parameter group |
| 2.2 Set the value to write | FB287 | ValueWrite2 | 1 | Servo parameter group number |
| 2.3 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 3. Set the servo parameter number | | | | |
| 3.1 Select P parameter number | FB287 | Parameter | 21 | 21: P parameter number |
| 3.2 Set the value to write | FB287 | ValueWrite2 | 44 | Servo parameter number |
| 3.3 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 4. Set the servo parameter value | | | | |
| 4.1 Select P parameter value | FB287 | Parameter | 22 | 22: P parameter value |
| 4.2 Set the value to write | FB287 | ValueWrite2 | 16777216 | Servo parameter value<br>Note: the input value should be in decimal. If the parameter value is in hex, convert it into a decimal value. |
| 4.3 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |

Example 2: read the servo parameter P0.001

| Step | FB | Pin | Input value | Description |
|---|---|---|---|---|
| 1. Enable the writing function for parameters | FB287 | ReadWrite | 1 | 0: enable the reading function<br>1: enable the writing function |
| 2. Set the servo parameter group | | | | |
| 2.1 Select P parameter group | FB287 | Parameter | 20 | 20: P parameter group |
| 2.2 Set the value to write | FB287 | ValueWrite2 | 0 | Servo parameter group number |
| 2.3 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 3. Set the servo parameter number | | | | |
| 3.1 Select P parameter number | FB287 | Parameter | 21 | 21: P parameter number |
| 3.2 Set the value to write | FB287 | ValueWrite2 | 1 | Servo parameter number |
| 3.3 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |
| 4. Enable the reading function for parameters | FB287 | ReadWrite | 0 | 0: enable the reading function<br>1: enable the writing function |
| 5. Set the servo parameter value | | | | |
| 5.1 Select P parameter value | FB287 | Parameter | 22 | 22: P parameter value |
| 5.2 Start | FB287 | Start | ⊓ | 0: disable the function<br>⊓: start accessing |

**Read the servo data:**

| Step | FB | Pin | Read value | Description |
|---|---|---|---|---|
| 1. Read the servo parameter value | FB284 | ValueRead2 | - | The read value is displayed in decimal. If the parameter value is in hex, it is converted into a decimal value. |

# 13.6 SINA function blocks and PNU parameters

## 13.6.1 Position control (SINA_POS, FB284)

When using telegram 111, you can achieve the basic positioning control with FB284. The description of the function block is as follows.



| SINA_POS (FB284) | | | | |
|---|---|---|---|---|
| Input pin | Data type | Default | Function | Description |
| ModePos | INT | 0 | Operating mode | 1: relative positioning<br>2: absolute positioning<br>3: positioning as setup<br>4: homing<br>5: regard the current feedback position as the origin<br>6: reserved<br>7: jog<br>8: incremental jogging |
| EnableAxis | BOOL | 0 | Servo On / Off | 0: Servo Off<br>1: Servo On |
| CancelTraversing | BOOL | 1 | Cancel current command | 0: cancel the current command<br>1: disable the function<br>Note: when this pin = 0, you need to reissue the command. |
| IntermediateStop | BOOL | 1 | Pause current command | 0: pause the current command<br>1: disable the function |
| Positive | BOOL | 0 | Direction for signal positioning and signal setting-up: positive | 0: disable the function<br>1: operate in positive direction |
| Negative | BOOL | 0 | Direction for signal positioning and signal setting-up: negative | 0: disable the function<br>1: operate in negative direction |
| Jog1 | BOOL | 0 | Jog operation direction: forward | 0: disable the function<br>1: jog in forward direction |
| Jog2 | BOOL | 0 | Jog operation direction: reverse | 0: disable the function<br>1: jog in reverse direction |
| FlyRef | BOOL | 0 | Reserved | - |
| AckError | BOOL | 0 | Servo alarm reset | 0: disable the function<br>↑: reset the servo alarm |
| ExecuteMode | BOOL | 0 | Execute commands of signal positioning and signal setting-up | 0: disable the function<br>↑: execute commands of signal positioning and signal setting-up |
| Position | DINT | 0 | Position setpoint when ModePos = 1 and 2 | Unit: PUU |
| Velocity | DINT | 0 | Speed setpoint when ModePos = 1, 2, and 3 | Unit: 1000 PUU/min |
| OverV | INT | 100 | Velocity override | Setting range: 1 - 199% |
| OverAcc | INT | 100 | Acceleration override | Setting range: 1 - 199% |
| OverDec | INT | 100 | Deceleration override | Setting range: 1 - 199% |

| SINA_POS (FB284) | | | | |
|---|---|---|---|---|
| Input pin | Data type | Default | Function | Description |
| ConfigEPos | DWORD | 16#3 | Control bit of telegram 111 | The motor runs only when Bit 0 and Bit 1 are both 1. |
| | | | Bit 0: free-run stop (AL013 triggering; OFF2) | 0: trigger servo alarm AL013<br>1: clear servo alarm AL013 |
| | | | Bit 1: PFQS (AL35F triggering; OFF3) | 0: trigger servo alarm AL35F<br>1: clear servo alarm AL35F |
| | | | Bit 2: software limit switch | 0: disable the function<br>1: enable the software limit switch |
| | | | Bit 3: hardware limit switch | 0: disable the function<br>1: enable the hardware limit switch |
| | | | Bit 4 - 7: reserved | - |
| | | | Bit 8: how the command becomes effective when ModePos = 2 or 3 | 0: effective when ExecuteMode is triggered<br>1: effective once the command setting value is changed |
| | | | Bit 9 - 15: reserved | - |
| HWIDSTW | HW_IO | 0 | Hardware ID of telegram 111 | Hardware ID in the Device view; same as HWIDZTW. |
| HWIDZSW | HW_IO | 0 | Hardware ID of telegram 111 | Hardware ID in the Device view; same as HWIDSTW. |

| SINA_POS (FB284) | | | | |
|---|---|---|---|---|
| Output pin | Data type | Default | Function | Description |
| AxisEnabled | BOOL | 0 | Servo state | 0: Servo Off<br>1: Servo On |
| AxisPosOk | BOOL | 0 | Target position reached | 1: target position reached |
| AxisSpFixed | BOOL | 0 | Position setpoint reached | 1: position setpoint reached |
| AxisRef | BOOL | 0 | Set reference point | 1: reference point set |
| AxisWarn | BOOL | 0 | Servo error (warning) | 1: servo error (warning) |
| AxisError | BOOL | 0 | Servo error (alarm) | 1: servo error (alarm) |
| Lockout | BOOL | 0 | Servo switching-on inhibited | 1: inhibited; check if Bit 0 and Bit 1 of ConfigEPos are both 1 |
| ActVelocity | DINT | 0 | Actual velocity | The actual velocity.<br>Read value (unit: 16#40000000h indicates 100% of the motor rated speed) * rotary motor rated speed / 16#40000000h = Actual velocity (unit: rpm) |
| ActPosition | DINT | 0 | Actual position | Actual position (unit: PUU) |
| ActMode | INT | 0 | Current operating mode | Displays the setting value of ModePos |
| EPosZSW1 | WORD | 0 | Status of EPosZSW1 | Displays the status of EPosZSW1 |
| EPosZSW2 | WORD | 0 | Status of EPosZSW2 | Displays the status of EPosZSW2 |
| ActWarn | WORD | 0 | Servo error code (Warning) | Displays the error code of the warning. For instance, when the servo displays AL013, the output is 16#0013h. |
| ActFault | WORD | 0 | Servo error code (Alarm) | Displays the error code of the alarm. For instance, when the servo displays AL02A, the output is 16#002Ah. |
| Error | BOOL | 0 | Controller error present | 0: controller is in normal operation<br>1: controller is in error |
| Status | WORD | 0 | Status | 16#7002: no error; function block is in execution<br>16#8401: servo is in error<br>16#8402: servo switching-on inhibited<br>16#8403: in operation; homing cannot be executed<br>16#8600: DPRD_DAT error<br>16#8601: DPWR_DAT error<br>16#8202: incorrect operating mode<br>16#8203: incorrect setpoints parameterized<br>16#8204: incorrect program block (the function is not yet supported) |
| DiagID | WORD | 0 | Communication error when SFB is called | SFB (system function block) is in error |

## 13.6.2 Speed control (SINA_SPEED, FB285)

When using telegram 1, you can achieve the basic speed control with FB285. The description of the function block is as follows.

13

| SINA_SPEED (FB285) | | | | |
|---|---|---|---|---|
| Input pin | Data type | Default | Function | Description |
| EnableAxis | BOOL | 0 | Servo On / Off | 0: Servo Off<br>1: Servo On |
| AckError | BOOL | 0 | Servo alarm reset | ⤒: reset the servo alarm |
| SpeedSp | REAL | 0.0 | Speed setpoint | Unit: rpm |
| RefSpeed | REAL | 0.0 | Motor rated speed | Unit: rpm |
| ConfigAxis | WORD | 16#003F | Control bit of telegram 1 | The motor runs only when Bit 0 and Bit 1 are both 1. |
| | | | Bit 0: free-run stop (AL013 triggering; OFF2) | 0: trigger servo alarm AL013<br>1: when STW1.2 is also set to 1, the servo can be switched to On |
| | | | Bit 1: PFQS (AL35F triggering; OFF3) | 0: trigger servo alarm AL35F<br>1: when STW1.1 is also set to 1, the servo can be switched to On |
| | | | Bit 2: ready for operation | 0: disable operation<br>1: enable operation |
| | | | Bit 3: Quick stop | 0: servo is in the Quick stop state<br>1: disable the function |
| | | | Bit 4: Halt | 0: halt the operation; once the halt function is disabled, the operation continues until the command is complete<br>1: disable the function |
| | | | Bit 5: command triggering | 0: disable the function<br>1: trigger the command |
| | | | Bit 6 - 15: reserved | - |
| HWIDSTW | HW_IO | 0 | Hardware ID of telegram 1 | Hardware ID in the Device view; same as HWIDZSW. |
| HWIDZSW | HW_IO | 0 | Hardware ID of telegram 1 | Hardware ID in the Device view; same as HWIDSTW. |

| SINA_SPEED (FB285) | | | | |
|---|---|---|---|---|
| Output pin | Data type | Default | Function | Description |
| AxisEnabled | BOOL | 0 | Axis enabled | 0: disable<br>1: enable |
| Lockout | BOOL | 0 | Servo switching-on inhibited | 0: disable<br>1: enable |
| ActVelocity | REAL | 0.0 | Actual velocity | Unit: rpm |
| Error | BOOL | 0 | Servo error present | 0: servo is in normal operation<br>1: servo is in error |
| Status | INT | 0 | Status | 16#7002: no error; function block is in execution<br>16#8401: servo is in error<br>16#8402: servo switching-on inhibited<br>16#8403: in operation; homing cannot be executed<br>16#8600: DPRD_DAT error<br>16#8601: DPWR_DAT error<br>16#8202: incorrect operating mode<br>16#8203: incorrect setpoints parameterized<br>16#8204: incorrect program block (the function is not yet supported) |
| DiagID | WORD | 0 | Communication error when SFB is called | SFB (system function block) is in error |

### 13.6.3 Acyclic reading / writing (SINA_PARA_S, FB287)

When using telegram 1, 3, 105, 102, or 111, you can read and write PNU parameters with FB287. The description of the function block is as follows.

13

| SINA_PARA_S (FB287) | | | | |
|---|---|---|---|---|
| Input pin | Data type | Default | Function | Description |
| Start | BOOL | 0 | Start | 0: disable the function<br>↑: start accessing |
| ReadWrite | BOOL | 0 | Accessing | 0: enable the reading function<br>1: enable the writing function |
| Parameter | INT | 1 | PNU parameter | PNU parameters are written through ValueWrite2<br>10: Homing method<br>11: Home offset<br>12: Homing speeds 1<br>13: Homing speeds 2<br>14: Homing acceleration<br>15: Profile acceleration<br>16: Profile deceleration<br>Servo parameter definition: PA.B = C<br>20: P parameter group (A)<br>21: P parameter number (B)<br>22: P parameter value (C) |
| Index | INT | 0 | PNU parameter index | Refer to the description of each PNU parameter. |
| ValueWrite1 | REAL | 0.0 | Field for writing the parameter value (REAL) | No need to use this pin. |
| ValueWrite2 | DINT | 0 | Field for writing the parameter value (DINT) | Currently used for all PNU parameters. |
| AxisNo | INT | 1 | Axis number | No need to change the number. |
| hardwareId | HW_IO | 0 | Hardware ID | Hardware ID in the Device view |

| SINA_PARA_S (FB287) | | | | |
|---|---|---|---|---|
| Output pin | Data type | Default | Function | Description |
| Ready | BOOL | 0 | Ready | 1: ready to read / write parameters |
| Busy | BOOL | 0 | Command being processed | 1: reading / writing parameters |
| Done | BOOL | 0 | Command complete | 1: finished reading / writing parameters |
| ValueRead1 | REAL | 0 | Field for reading the parameter value (REAL) | No need to use this pin. |
| ValueRead2 | DINT | 0 | Field for reading the parameter value (DINT) | The read value of PNU parameter |
| Format | INT | 0 | Format of the read parameter | Parameter format code defined by the PROFIdrive profile |
| ErrorNo | INT | 0 | Display of error code | Error code defined by the PROFIdrive profile |
| Error | BOOL | 0 | Error present | 1: controller error present |
| ErrorId | DWORD | 0 | Error ID | Contact the controller manufacturer. |
| DiagId | WORD | 0 | Communication error when SFB is called | SFB (system function block) is in error. |

### 13.6.4 PNU parameters

**Important: currently all PNU parameters are volatile.**

PNU10: Homing method

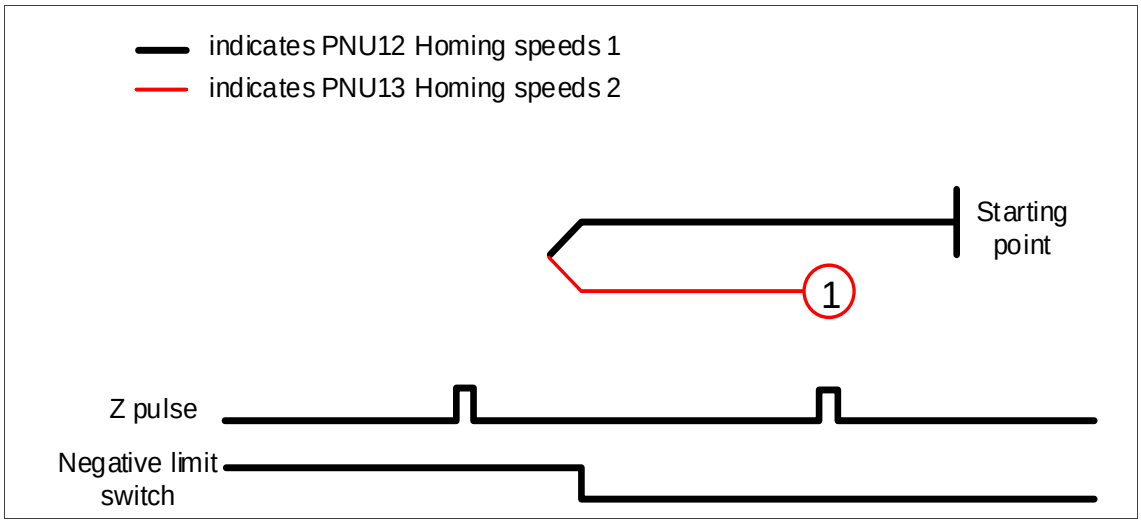
| | |
|---|---|
| Index | 0 |
| Name | Homing method |
| Data type | INTEGER8 |
| Access | RW |
| Setting range | 0 - 35 |
| Default | 0 |

Function:

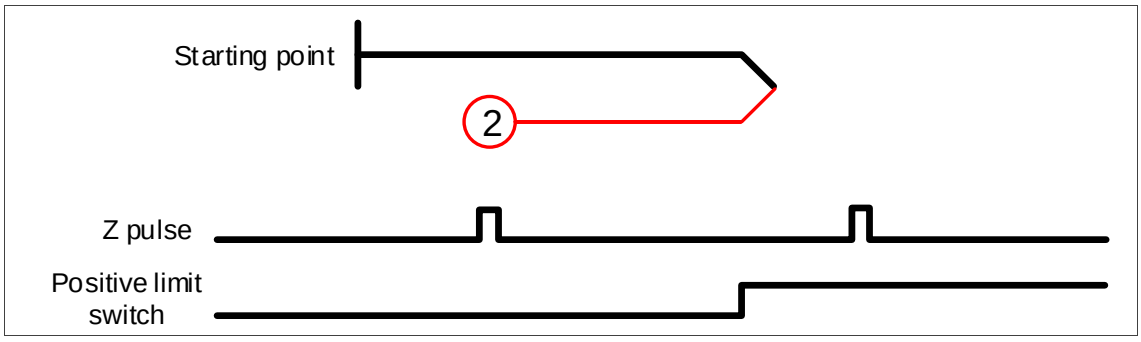
The homing methods include looking for the Z pulse (Methods 1 - 14, 33, 34), not looking for the Z pulse (Methods 17 - 30), and defining the current position as the origin (Method 35). Methods 15, 16, 31, and 32 are reserved.

To use Methods 1 to 35, set PNU10 as 1 to 35.

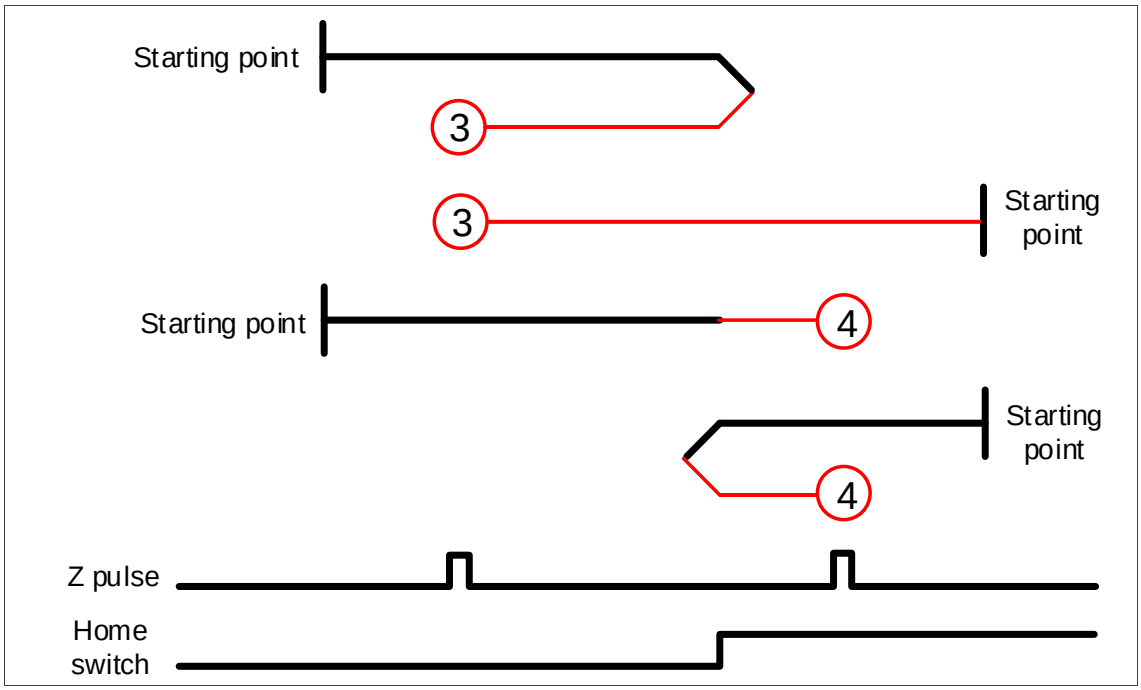
Method 1: homing on the negative limit switch and Z pulse



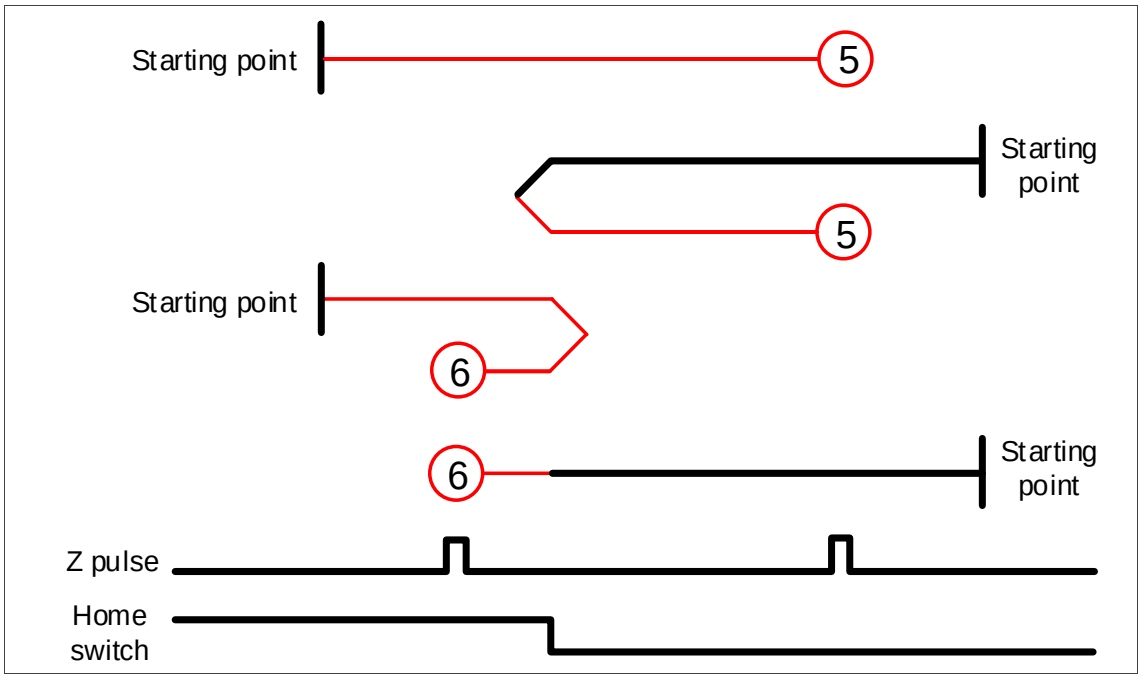

Method 2: homing on the positive limit switch and Z pulse



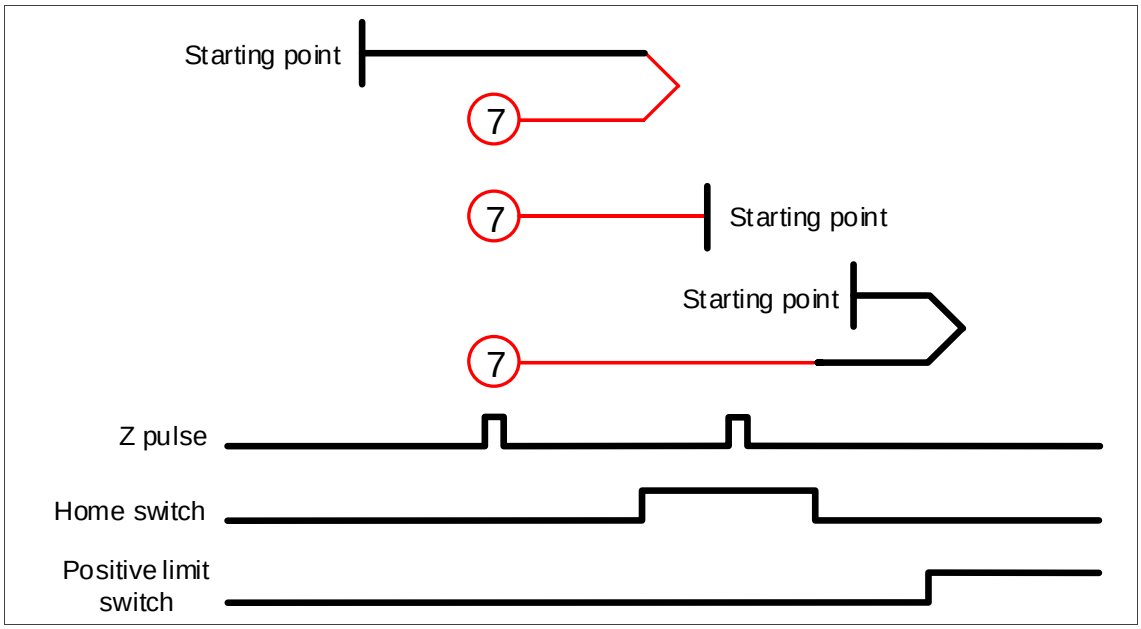
Methods 3 and 4: homing on the home switch and Z pulse



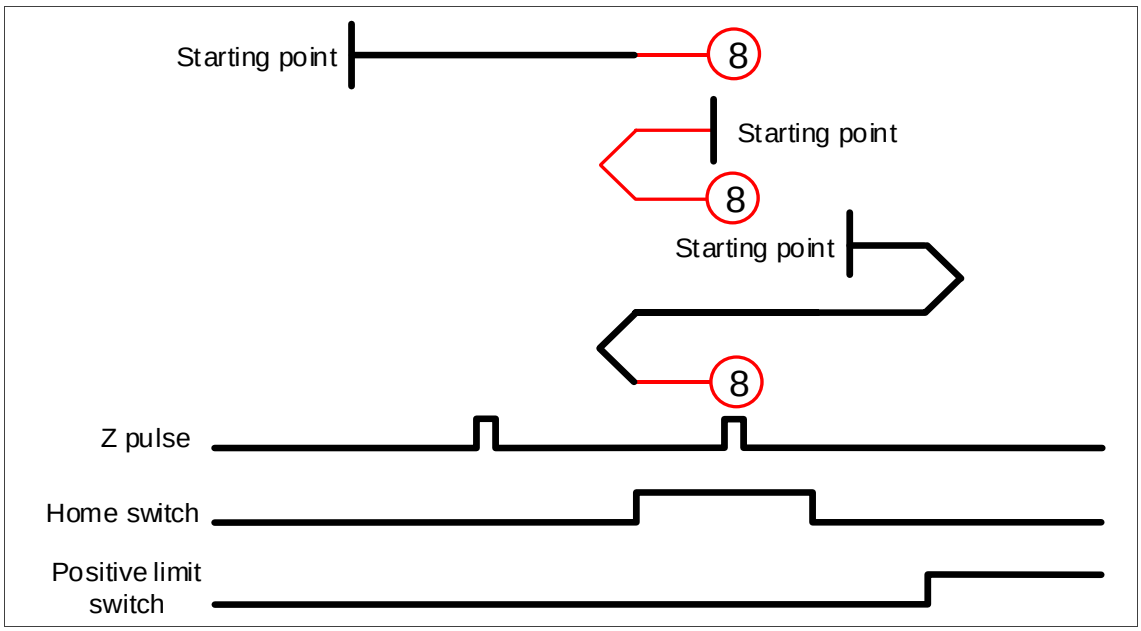

Methods 5 and 6: homing on the home switch and Z pulse



Method 7: homing on the positive limit switch, home switch, and Z pulse

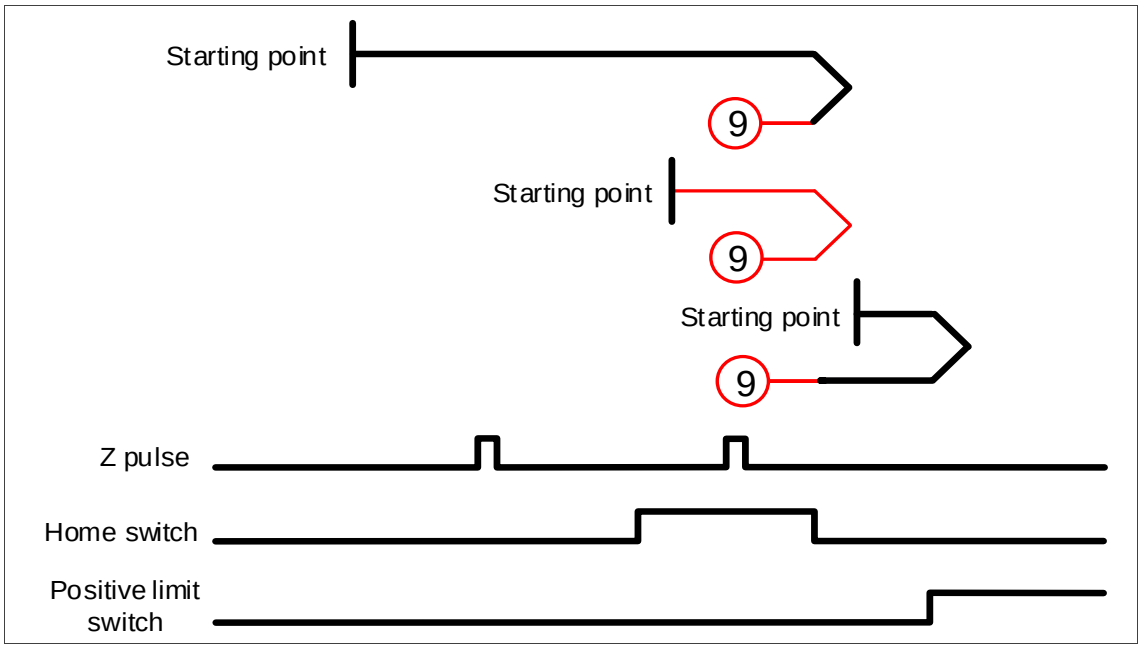



Method 8: homing on the positive limit switch, home switch, and Z pulse

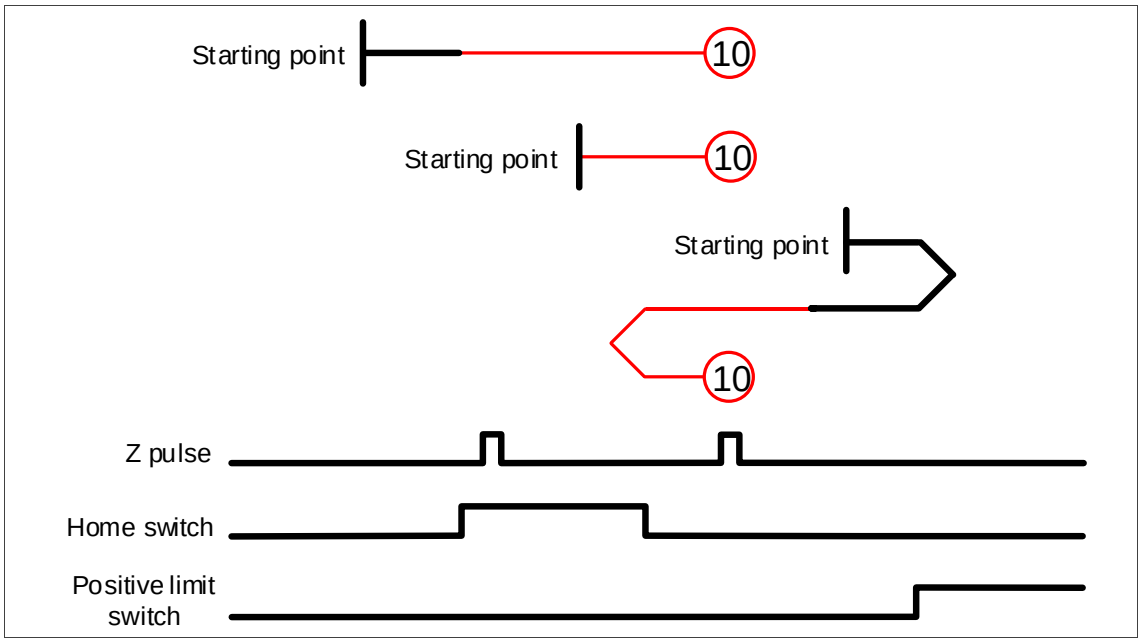


13

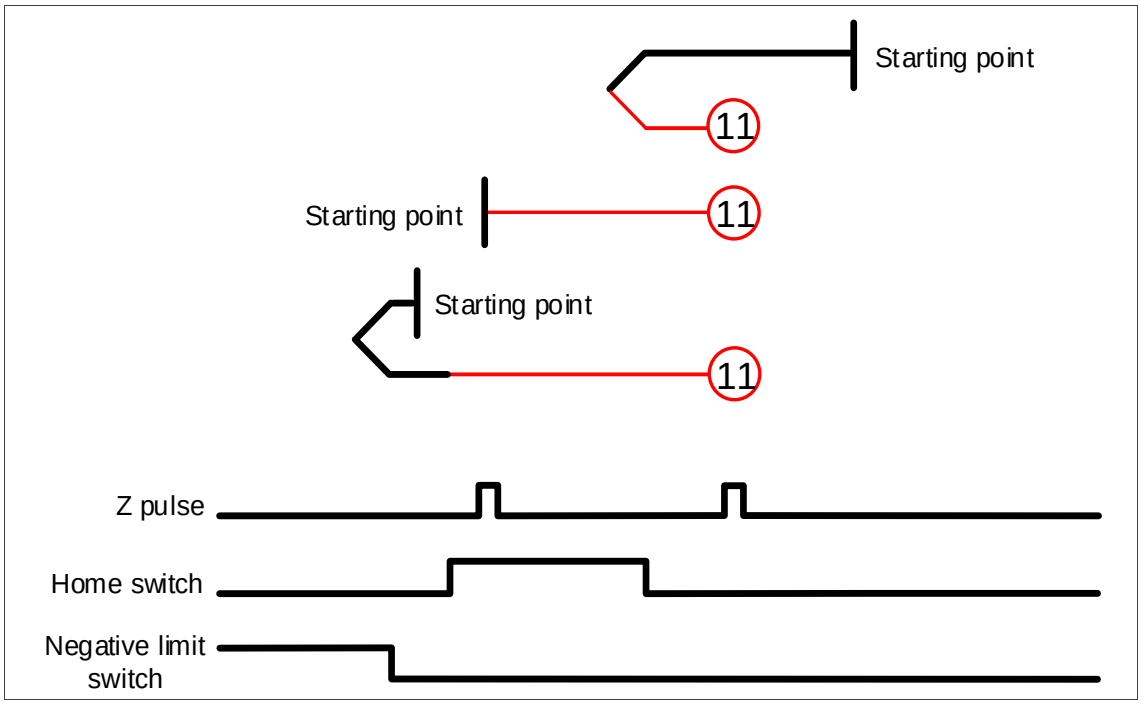
Method 9: homing on the positive limit switch, home switch, and Z pulse



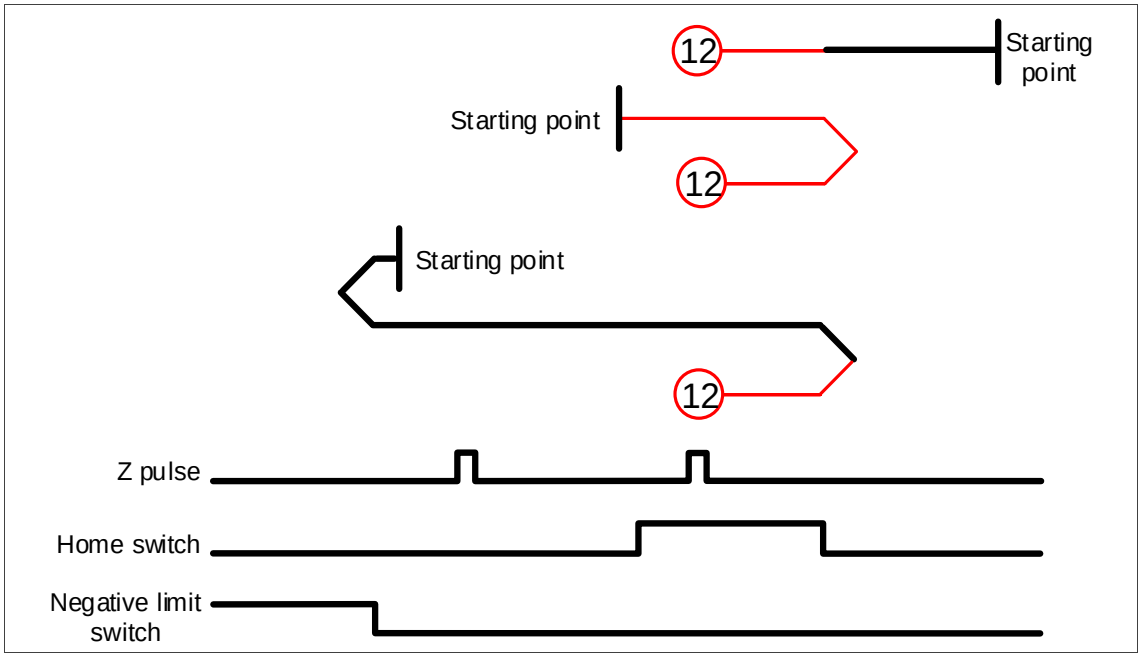

Method 10: homing on the positive limit switch, home switch, and Z pulse



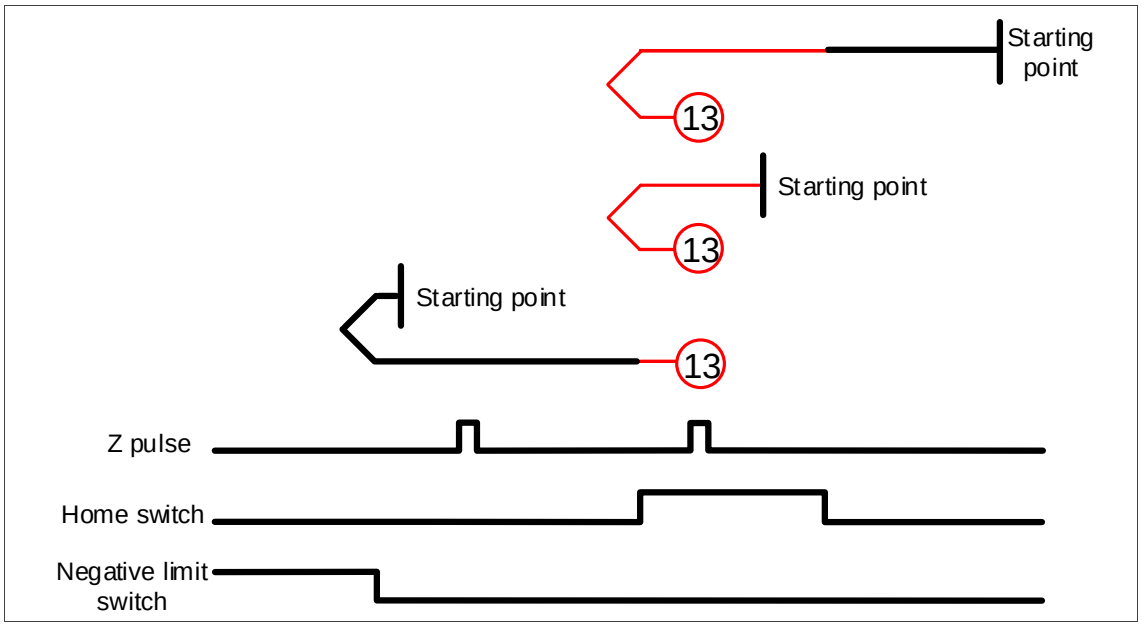
Method 11: homing on the negative limit switch, home switch, and Z pulse



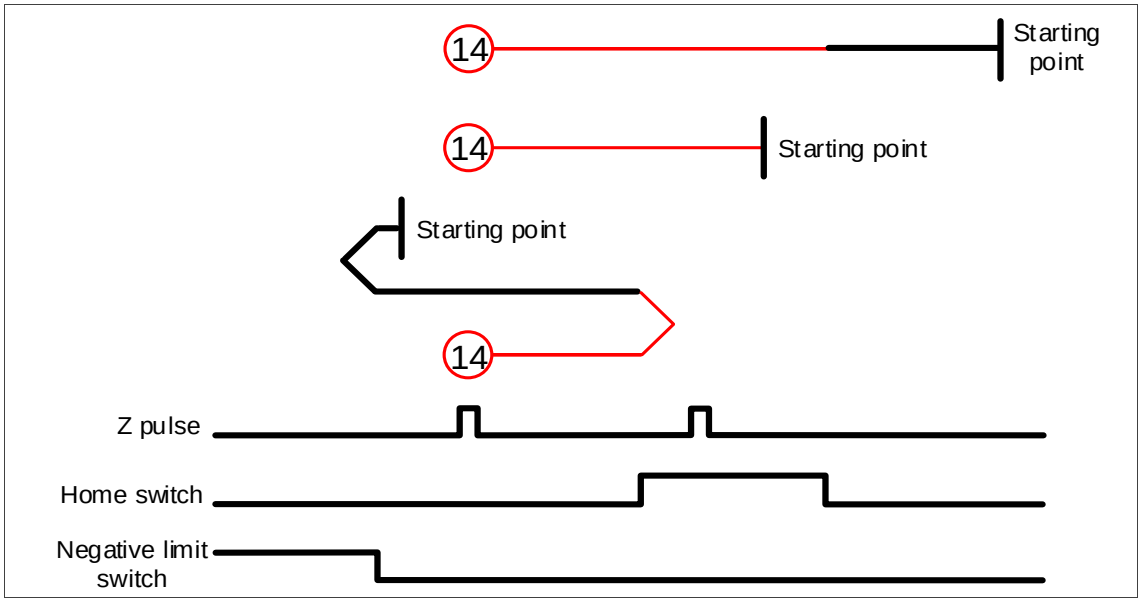

Method 12: homing on the negative limit switch, home switch, and Z pulse



Method 13: homing on the negative limit switch, home switch, and Z pulse

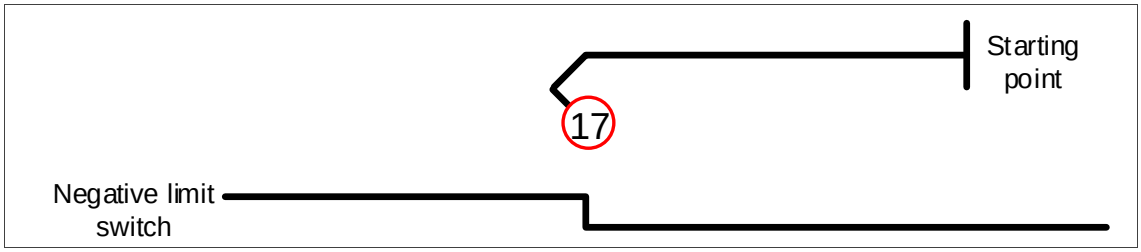



Method 14: homing on the negative limit switch, home switch, and Z pulse

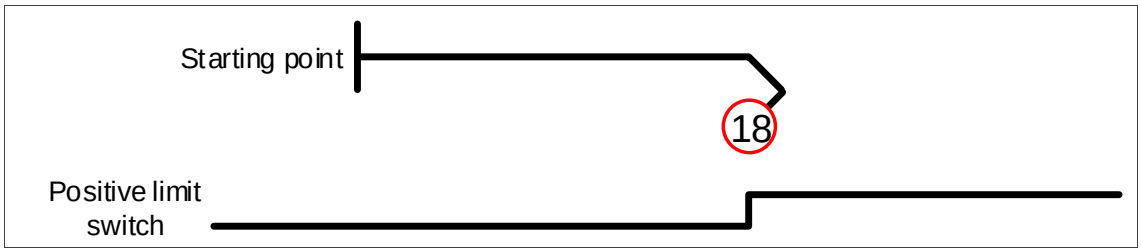



Methods 15 and 16: reserved

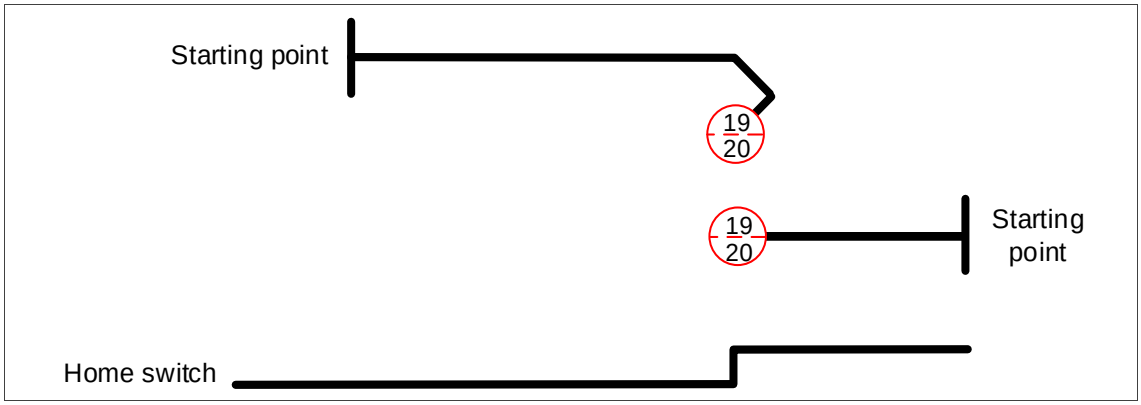
Method 17: homing on the negative limit switch



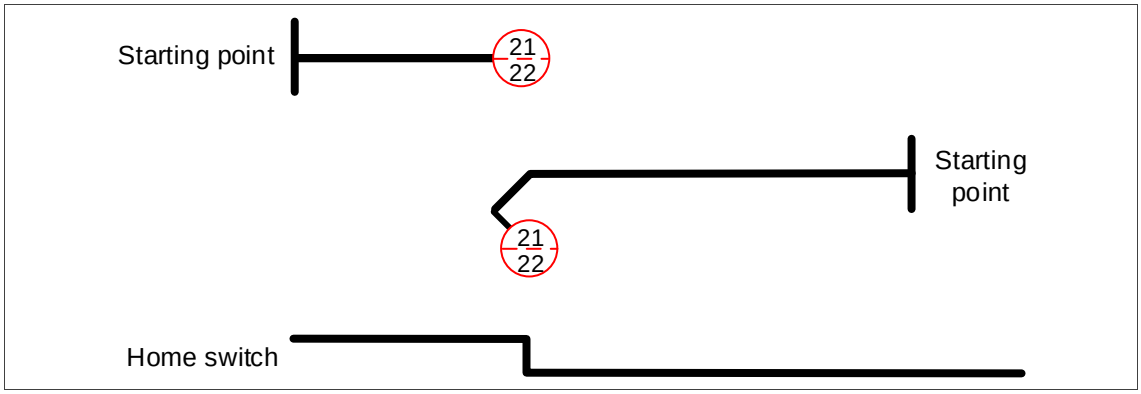
Method 18: homing on the positive limit switch




Methods 19 and 20: homing on the home switch

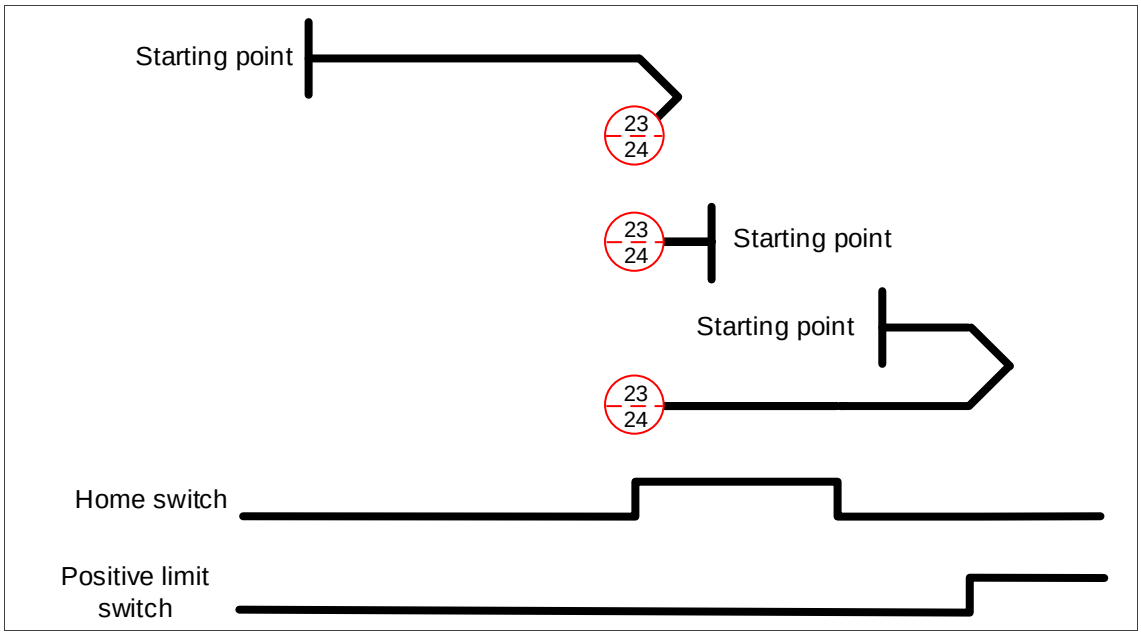



Methods 21 and 22: homing on the home switch

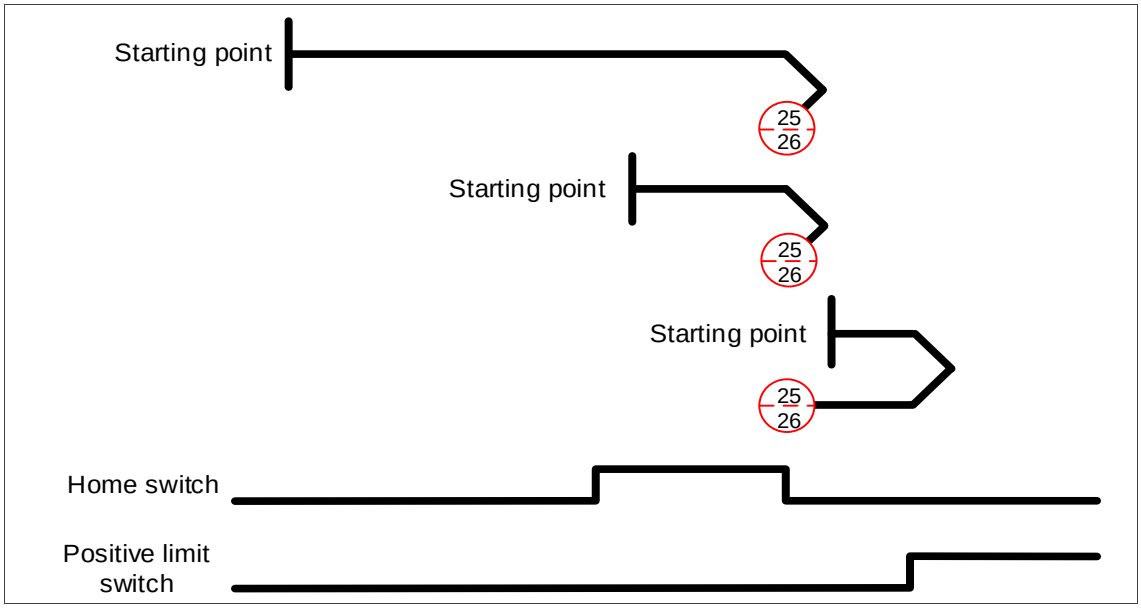


13

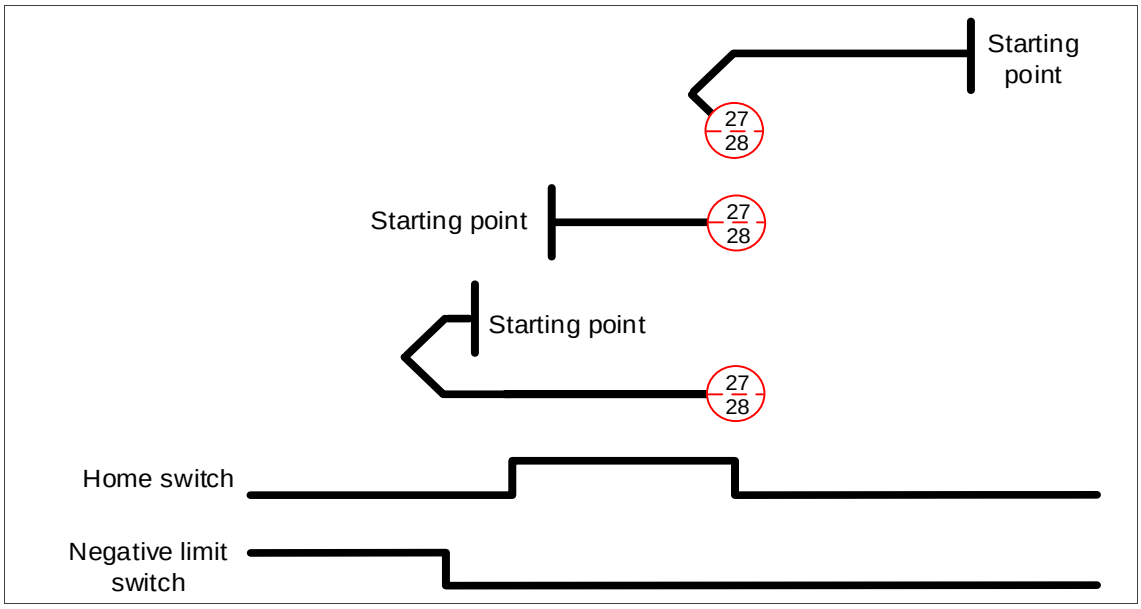
Methods 23 and 24: homing on the positive limit switch and home switch



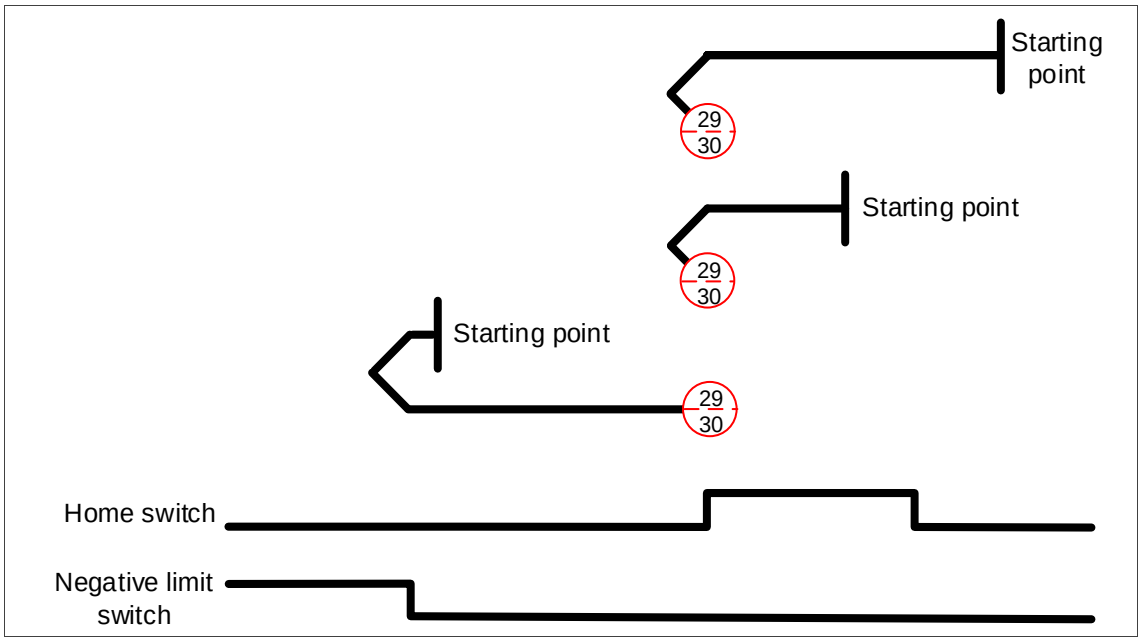

Methods 25 and 26: homing on the positive limit switch and home switch



Methods 27 and 28: homing on the negative limit switch and home switch

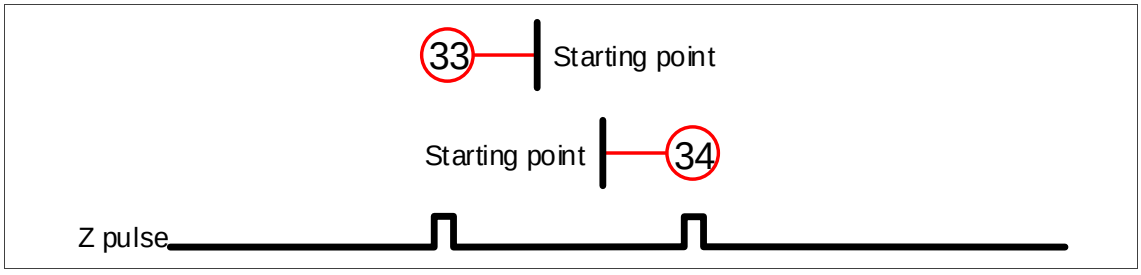



Methods 29 and 30: homing on the negative limit switch and home switch




Methods 31 and 32: reserved

Methods 33 and 34: homing on the Z pulse



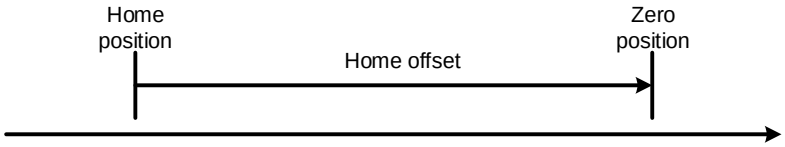

Method 35: define the current feedback position as the origin

PNU11: Home offset

| | |
|---|---|
| Index | 0 |
| Name | Home offset |
| Data type | INTEGER32 |
| Access | RW |
| Setting range | INTEGER32 |
| Default | 0 |
| Unit | PUU |

Function:

The origin reference point the system looks for during the homing procedure is Home Position, such as the origin sensor and Z pulse. When the origin reference point is found, the position offset from this point is the user-defined origin (Zero position), and the offset value is Home offset.




PNU12: Homing speeds 1

| | |
|---|---|
| Index | 0 |
| Description | Speed during search for switch |
| Data type | UNSIGNED32 |
| Access | RW |
| Setting range | 1 - 20000 |
| Default | 100 |
| Unit | 0.1 rpm |

PNU13: Homing speeds 2

| | |
|---|---|
| Index | 0 |
| Description | Speed during search for zero |
| Data type | UNSIGNED32 |
| Access | RW |
| Setting range | 1 - 5000 |
| Default | 20 |
| Unit | 0.1 rpm |

PNU14: Homing acceleration

| | |
|---|---|
| Index | 0 |
| Name | Homing acceleration |
| Data type | UNSIGNED32 |
| Access | RW |
| Setting range | 1 - 65500 |
| Default | 100 |
| Unit | ms |

Function:

The time slope set by this parameter is the time required for the motor to accelerate from 0 rpm to 3,000 rpm and decelerate from 3,000 rpm to 0 rpm. This parameter only works in Homing mode when telegram 111 is used.

PNU15: Profile acceleration

| | |
|---|---|
| Index | 0 |
| Name | Profile acceleration |
| Data type | UNSIGNED32 |
| Access | RW |
| Setting range | 1 - 65500 |
| Default | 200 |
| Unit | ms |

Function:

The time slope set by this parameter is the time required for the motor to accelerate from 0 rpm to 3,000 rpm. This parameter only works when telegram 111 (Profile Position mode) or telegram 1 (Profile Velocity mode) is used.

PNU16: Profile deceleration

| | |
|---|---|
| Index | 0 |
| Name | Profile deceleration |
| Data type | UNSIGNED32 |
| Access | RW |
| Setting range | 1 - 65500 |
| Default | 200 |
| Unit | ms |

Function:

The time slope set by this parameter is the time required for the motor to decelerate from 3,000 rpm to 0 rpm. This parameter only works when telegram 111 (Profile Position mode) or telegram 1 (Profile Velocity mode) is used.

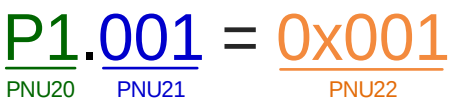
13

PNU20: P parameter group

| | |
|---|---|
| Index | 0 |
| Name | P parameter group |
| Data type | UNSIGNED32 |
| Access | RW |
| Parameter input pin of FB287 | ValueWrite2 |
| Setting range | 1 - 7 |
| Default | 0 |
| Unit | N/A |

Function:

This parameter sets the servo parameter group number, shown in green as follows.

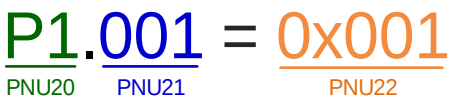



PNU21: P parameter number

| | |
|---|---|
| Index | 0 |
| Name | P parameter number |
| Data type | UNSIGNED32 |
| Access | RW |
| Parameter input pin of FB287 | ValueWrite2 |
| Setting range | 0 - 16#7FFFFFFF |
| Default | 0 |
| Unit | N/A |

Function:

This parameter sets the servo parameter number, shown in blue as follows.



PNU22: P parameter value

| | |
|---|---|
| Index | 0 |
| Name | P parameter value |
| Data type | SIGNED32 |
| Access | RW |
| Parameter input pin of FB287 | ValueWrite2 |
| Setting range | 0 - 16#7FFFFFFF |
| Default | 0 |
| Unit | N/A |

Function:

This parameter sets the servo parameter value, shown in orange as follows.

P1.001 = 0x001
PNU20 PNU21 PNU22

## 13.7 Troubleshooting

This section provides troubleshooting information related to communication or interference with the controller. For information about the servo alarms, refer to Chapter 14.



1. Query the PZD and PNU values

   The PZD and PNU are mapped to the OD objects, so you can query their values using the Scope function of ASDA-Soft. The mapping data is as follows.

| Name | Function | Mapping OD |
|---|---|---|
| MDI_TARPOS | MDI position | OD 607Ah |
| MDI_VELOCITY | MDI velocity | OD 6081h |
| NSOLL | Speed setpoint | OD 60FFh |
| XIST | Position actual value | OD 6064h |
| NIST | Speed actual value | OD 606Ch |
| PNU10 | Homing method | OD 6098h |
| PNU11 | Home offset | OD 607Ch |
| PNU12 | Homing speeds 1 | OD 6099h sub1 |
| PNU13 | Homing speeds 2 | OD 6099h sub2 |
| PNU14 | Homing acceleration | OD 609Ah |
| PNU15 | Profile acceleration | OD 6083h |
| PNU16 | Profile deceleration | OD 6084h |

2. Eliminate interference

   Packets are particularly sensitive to interference in high-speed network communication applications. To achieve fast and high-precision control, the selection of the wire is extremely important. Use shielded cables for the communication wiring, and make sure that the shielded connector is firmly connected to the servo drive communication port. Also, ensure the ground wire is properly connected and grounded.

3. Device name in the PROFINET tool does not match the servo device name

   The controller activates the servo by using a PROFINET tool (e.g., Siemens TIA Portal software) to connect to the servo and assign the servo device name based on DNS (Domain Name System). Therefore, the device name shown in the PROFINET tool (e.g., PROFINET device name in TIA Portal) must match the servo device name, and each device name must be unique.

   If using the servo for the first time or desiring to change the servo via the controller, you have to assign the corresponding device name in the PROFINET tool.

4. Technology object error

   Make sure you have cleared the check boxes of **Automatically apply drive values at runtime**, **Torque data**, and **Automatically apply encoder values at runtime**.