# ED3M (Standard) Series AC Servo User Manual

Version: V1.01

ESTUN AUTOMATION TECHNOLOGY CO., LTD

Total Solution Supplier

# Revision History

| No. | Date | Version | Description |
|---|---|---|---|
| 1 | Nov, 2018 | V1.00 | Initial release. |
| 2 | Jun, 2019 | V1.01 | Correction: 9.3.4 Data Communication Address in Servodrive. |

# About this Manual

## Purpose

This manual describes the following information required for designing and maintaining ED3M (Standard) Series AC Servo Drives.

- Specification of the Servodrives and Servomotors.
- Procedures for installing the Servodrives and Servomotors.
- Procedures for wiring the Servodrives and Servomotors.
- Procedures for operating of the Servodrives.
- Procedures for using the panel operator.
- Communication protocols.
- Ratings and characteristics.

Read and understand this manual to ensure correct usage of ED3M (Standard) Series AC Servo Drives.

Keep this manual in a safe place so that it can be referred to whenever ecessary.

## Intended Audience

This document is intended for:

- Those who designing ED3M (Standard) Series AC Servo Drives.
- Those who installing or wiring ED3M (Standard) Series AC Servo Drives.
- Those who performing trial operation or adjustments of ED3M (Standard) Series AC Servo Drives.
- Those who maintaining or inspecting ED3M (Standard) Series AC Servo Drives.

## Technical Terms

The following terms are used in this manual.

| Term | Meaning |
|---|---|
| Servomotor | An EM3A Series or EMG Series Rotary Servo Motor. |
| Servodrive | An ED3M (Standard) Series AC Servo Drive, which is used for controlling the motion of Rotary Servo Motor. |
| Servo System | A servo control system that includes a Seromotor, a Servodrive with a host controller and peripheral devices. |
| Servo ON | Supplying power to the motor. |
| Servo OFF | Not supplying power to the motor. |

# Symbol Conventions

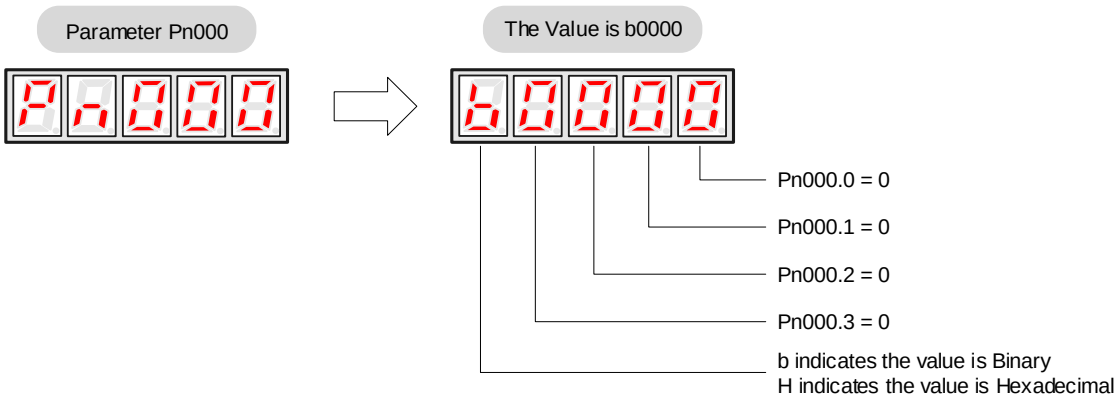
The symbols that may be found in this document are defined as follows.

| Symbol | Description |
|---|---|
| DANGER | Indicates a hazard with a high level of risk that, if not avoided, will result in death or serious injury. |
| WARNING | Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury. |
| CAUTION | Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results. |
| IMPORTANT | Indicates precautions or restrictions that must be observed.<br>Also indicates alarm displays and other precautions that will not result in machine damage. |
| NOTE | Provides additional information to emphasize or supplement important points of the main text. |

The names of reverse signals (i.e., ones that are valid when low) are written with a forward slash (/) before the signal abbreviation. For example:

$\overline{\text{S-ON}}$ is written as /S-ON; $\overline{\text{P-CON}}$ is written as /P-CON.

The names of sub-parameters are written as the format of Pn*xxx.x*. The following takes the parameter Pn000 as an example, whose sub-parameters Pn000.0, Pn000.1, Pn000.2 and Pn000.3 corresponding to one bit of its value respectively.



# Safety Precautions

**DANGER**

- Never connect the Servomotor directly to the local electrical network.
- Never plug or unplug connectors from the Servodrive when power is on.
- Wait for five minutes after turning OFF the power supply and then make sure that the CHARGE indicator is not lit before starting wiring or inspection work.
  Never touch the power supply terminals while the CHARGE lamp is lit after turning OFF the power supply, because high voltage may remain in the Servodrive.

**WARNING**

- Provide at least 10mm space between the Servodrives and the control panel or with other devices. In addition, the longitudinal space between them are above 50mm.
  Whenever possible, choose a layout that is conducive to heat dissipation.
  and the installation environment is not affected by condensation, vibration or shock.
- Install a high-sensitivity Ground Fault Detector against overloads and short-circuiting.
- Never perform any extreme adjustments or settings.
- Always start or stop the Servomotor by using reference pulses.
  Never operate the Servomotor by turning the power ON and OFF.
- Always match the proper encoder for each Servomotor.
- Check and confirm that the cables for each Servomotor has been properly connected to the Servodrive.

**CAUTION**

Comply with the following instructions to avoid noise generated by signal lines.

- Separate high-voltage cables from low-voltage cables.
- Use cables as short as possible.
- Connect the ground terminals on the Servodrive and Servomotor to ground poles according to local electrical codes (100 Ω or less).
- Never use a line filter for the power supply in the circuit.

# Contents



# Chapter 1 Basic Informations on Product

## 1.1 Checking Products

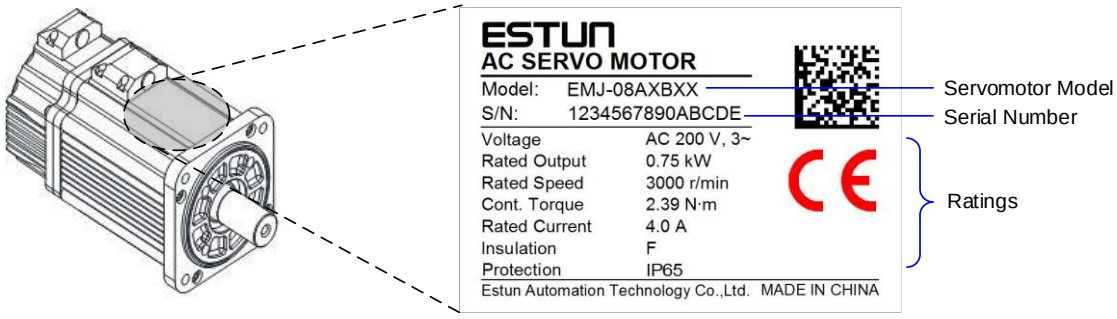
| Check Items | Comments |
|---|---|
| Are the delivered products theones that were ordered? | Check the model numbers marked on the nameplate on theservomotor and servo drive. |
| Is there any damage? | Check the overall appearance, and check for damage or scratches that may have occurred during shipping. |
| Does the servomotor shaft rotate smoothly? | The servomotor shaft is normal if it can be turned smoothly by hand.<br>Servomotors with brakes, however, cannot be turned manually. |

<NOTE>: If any of the above items are faulty or incorrect, contact your ESTUN representative or the dealer from whom you purchased the products.

### 1.1.1 Servomotor

**Nameplate Example**

The following figure shows the nameplate of EMJ model Servomotor as an example. Nameplates of the EML model and EMG model are similar.




**Model Designations**

EMJ - A5 A S A 2 2

EMJ Model Servomotor

1st +2nd digits — Rated output

| | |
|---|---|
| A5 | 50 W |
| 01 | 100 W |
| 04 | 400 W |

3rd digit — Supply voltage

| | |
|---|---|
| A | 200V |

4th digit — Encoder specification

| | |
|---|---|
| F | 20-bit incremental encoder |
| S | 17-bit absolute encoder |

5th digit — Design revision order

| | |
|---|---|
| A ~ Z | Order A ~ Order Z |

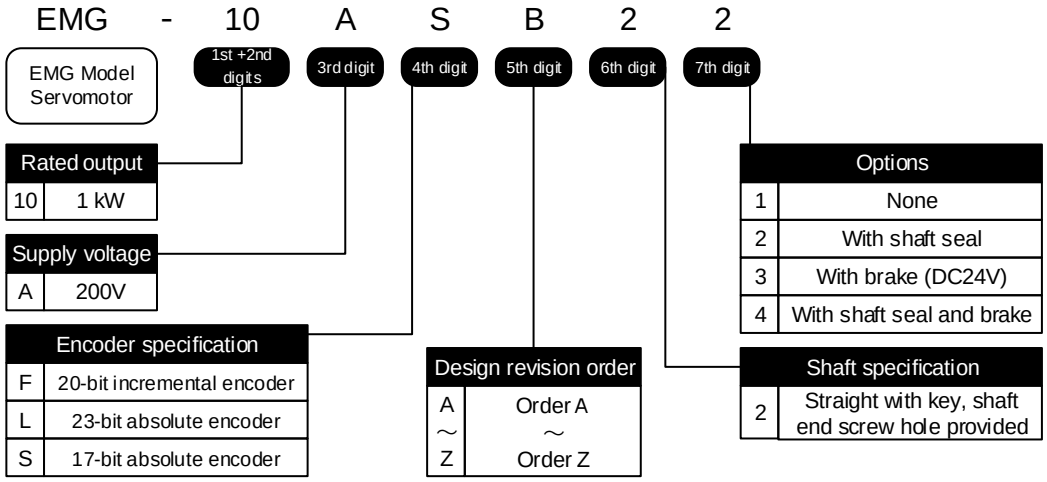
6th digit — Options

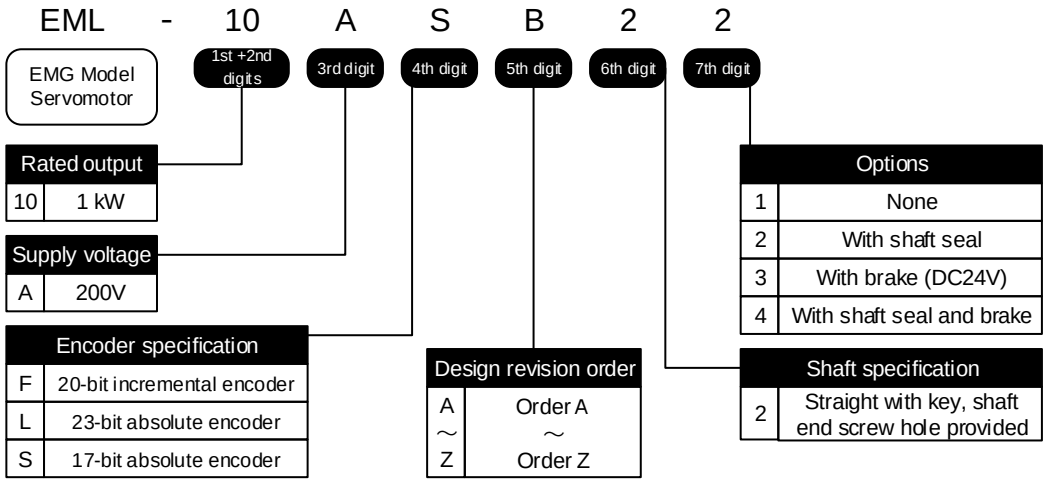
| | |
|---|---|
| 1 | None |
| 2 | With shaft seal |
| 3 | With brake (DC24V) |
| 4 | With shaft seal and brake |

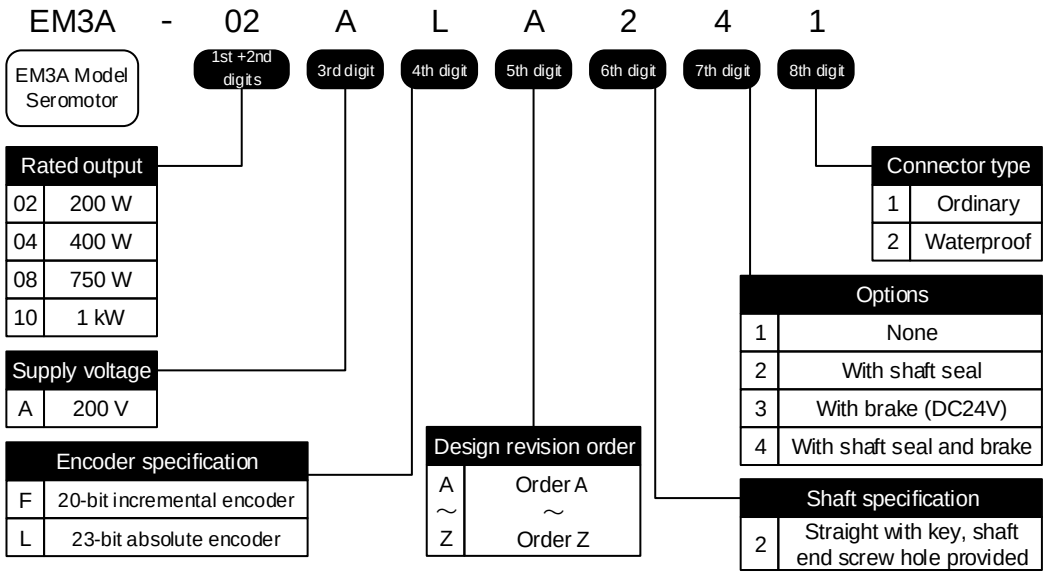
7th digit — Shaft specification

| | |
|---|---|
| 2 | Straight with key, shaft end screw hole provided |



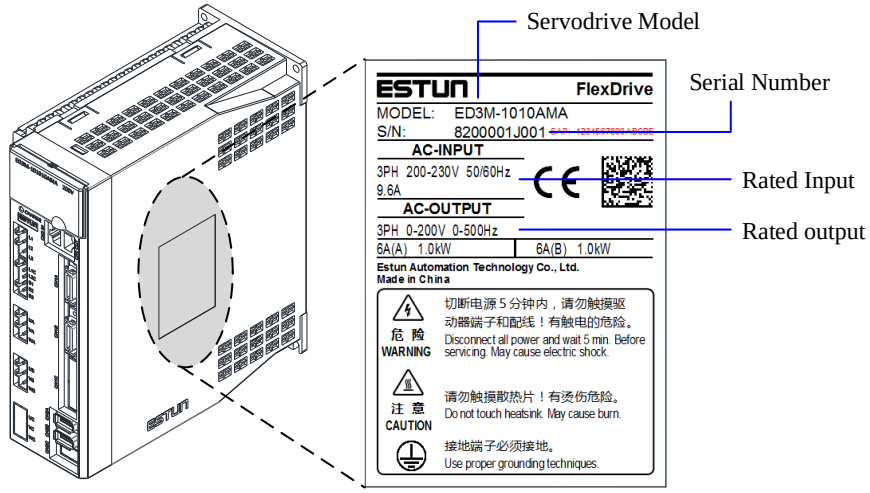



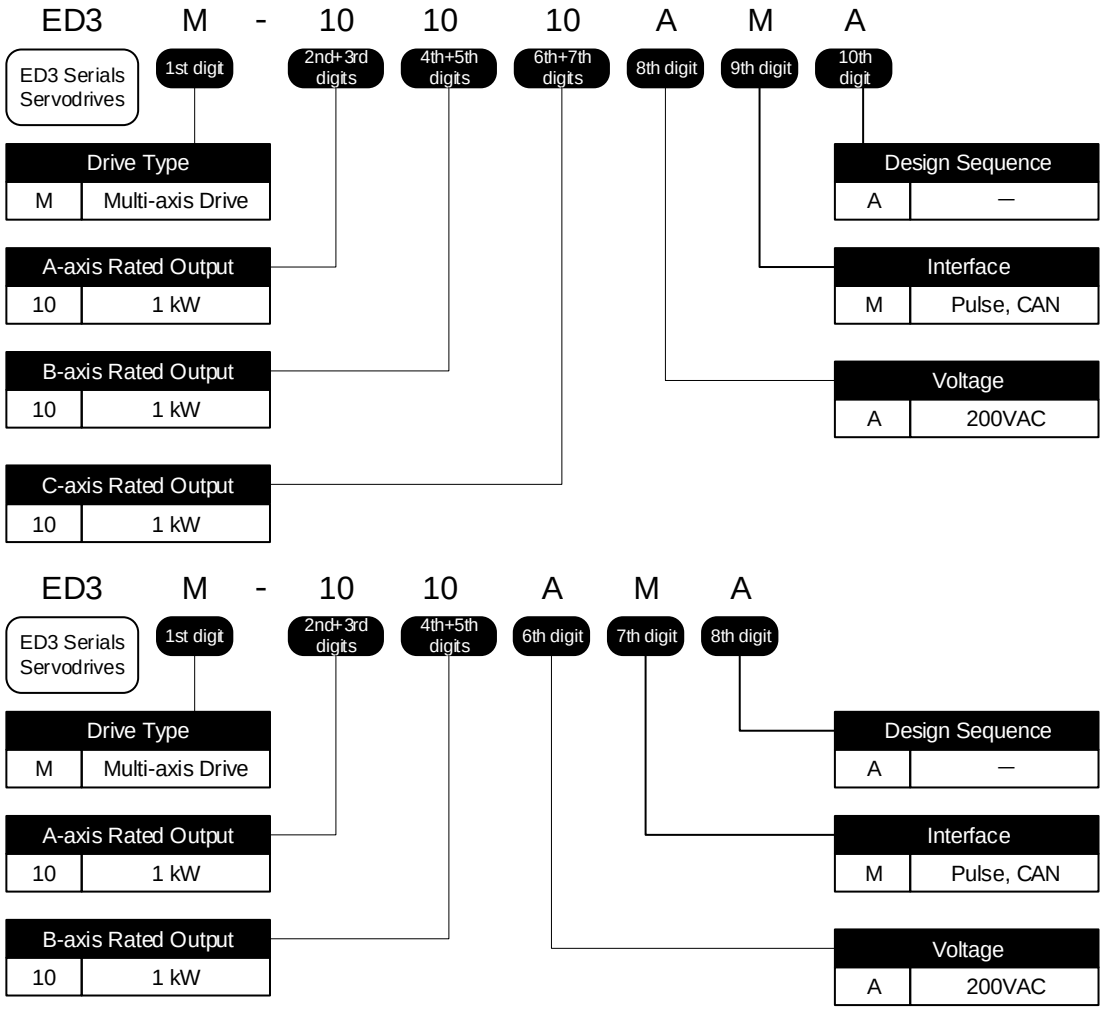
## 1.1.2 Servodrive

### Nameplate Example

The following figure shows the nameplate of ED3M model Double-Axis Servodrive as an example. The nameplate of the ED3M model Treble-Axis Servodrive is similar.




### Model Designations



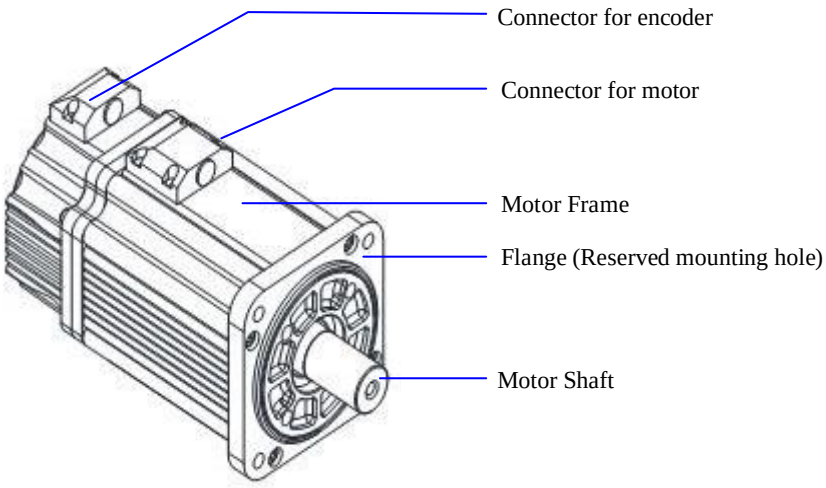

[NOTE] All the axis of ED3M servodrive controlled can be applied for the motor power of 50W to 1kW, those also can be set up to treble in overload capability.

# 1.2 Part Names

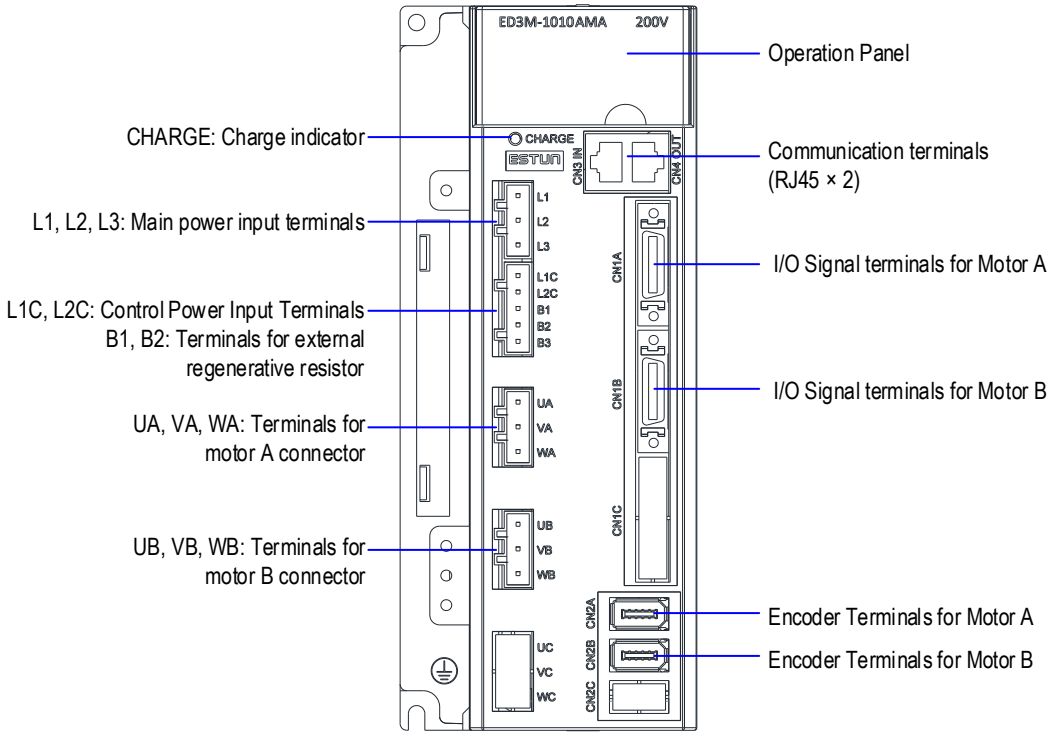
## 1.2.1 Servomotor

The following figure shows the part names of EMJ model Servodmotor as an example. The part names of the EML model and EMG model Servodrives are similar.




## 1.2.2 Servodrive

The following figure shows the part names of ED3M model Double-Axis Servodrive as an example. The part names of the ED3M model Treble-Axis Servodrive is similar.




**WARNING**

Even if you turn OFF the main circuit power supply, the CHARGE indicator will be lit as long as the internal capacitor remains charged. Never touch the main circuit or motor terminals while this indicator is lit, to avoid the electric shock.

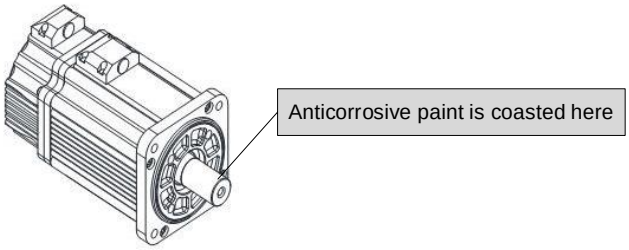
  

# Chapter 2 Installation

## 2.1 Servomotor

Anticorrosive paint is coated on the edge of the motor shaft to prevent it from rusting during storage. Clean off the anticorrosive paint thoroughly using a cloth moistened with thinner before installing the motor.

CAUTION




### 2.1.1 Conditions

**Storge**

When the Servomotor is to be stored with the power cable disconnected, store it in the following temperature range:
Between -20 and 60℃.

**Installation Sites**

The servomotor is designed for indoor use. Install the servomotor in an environment which meets the following conditions:

- Free from corrosive and explosive gases.
- Well-ventilated and free from dust and moisture.
- Ambient temperature of 0 to 40℃.
- Relative humidity of 26% to 80% (non-condensing).
- Inspection and cleaning can be performed easily.

**Installation Orientation**

You can install the Servomotor either horizontally or vertically.

However, you shall install the Servomotor according to the actual use of the machine, which makes the motor work best with the machine for ensuring the service life of the Servomotor or avoiding accidents.

**Using Servomotors with Holding Brakes**

This section gives precautions for using Servomotors with Holding Brakes.

- The holding brakes have a limited service life. Although the quality and reliability of a holding brake has been sufficiently confirmed, stress factors, such as emergency braking, can results in problems in the holding operation. In applications in which safety is a concern, such as for a load falling on a vertical axis, determine if safety measures are required on the machine, such as adding a redundant fall-prevention mechanism.
- For a Servomotor with a Holding Brake, there is a small amount of rotational play in the motor shaft (1.5° max. initially) because of the backlash in the holding brake, even when the brake power is OFF.

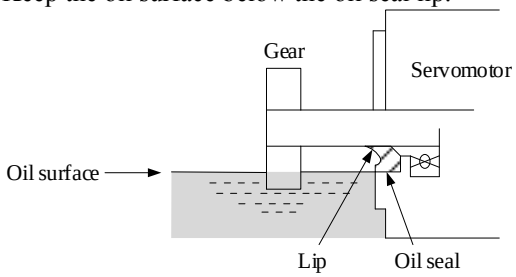
  

- For a Servomotor with a Holding Brake, the brake's rotating disc may sometimes generate murmur from friction during acceleration, stopping, and low-speed operation.

### Using Servomotors with Oil Seals

This section gives the operating conditions for using Servomotors with Oil Seals.

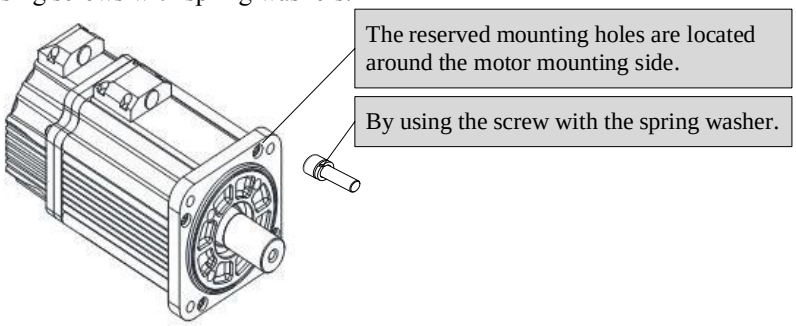
- Keep the oil surface below the oil seal lip.




- Use the oil seal in favorably lubricated condition with only splashing of oil.
- Never let the oil collect in the oil seal lip.
- Never use the Servomotor where the oil seal would be below the oil surface. If you do, oil will enter the Servomotor, which may damage the Servomotor.

## 2.1.2 Coupling to the Machine

For instlling the motor to the client, connect the motor with the load via the reserved mounting hole by using screws with spring washers.




### Using a Coupling

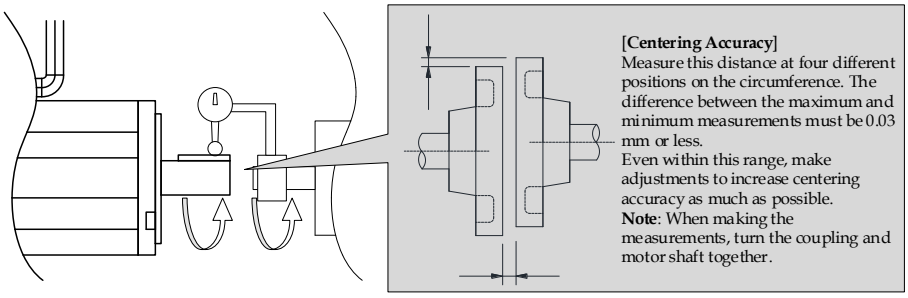
**IMPORTANT**

- Use a flexible coupling that is designed for Servomotors.
- Select a suitable size of coupling for the operating conditions. An inappropriate coupling may cause damage.

1. Wipe off all of the anticorrosive coating from the motor shaft.
2. If you are using a Servomotor with a Key, attach the key enclosed with the Servomotor or the specified size of key to the shaft.
3. Confirm that the centering accuracy is within the specified range using a dial gauge or other means. If a dial gauge is not available, slide the coupling along both shafts and make adjustments so that it does

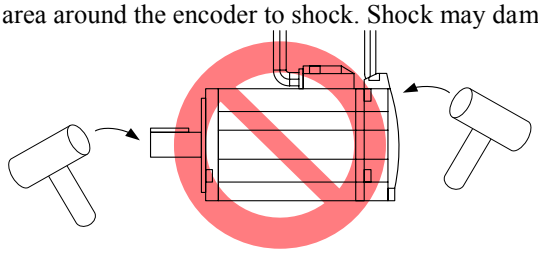
not catch.




4. Align the shaft of the Servomotor with the shaft of the machine, and then connect the shafts with the coupling.

- When you couple the shafts, make sure that the required centering accuracy is achieved. Vibration will damage the bearings and encoders if the shafts are not properly centered.
- When you attach the coupling, do not subject the shaft to direct shock. Also, do not subject the area around the encoder to shock. Shock may damage the encoder.

**CAUTION**



- If the coupling makes any abnormal noise, center the shafts again until the noise is eliminated.
- Make sure that the thrust load and radial load are within specifications. Refer to the specifications for each type of Servomotor for the thrust load and radial load.

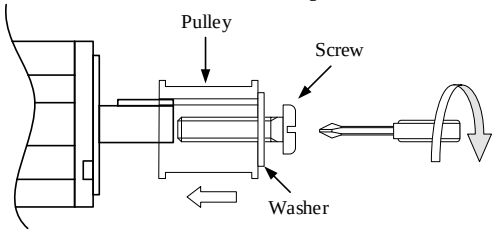
## Using a Belt

**IMPORTANT**

Select a coupling belt that is suitable for the allowable radial load of the Servomotor and the Servomotor output.

When the Servomotor accelerates or decelerates, the counterforce from the acceleration/deceleration torque adds tension to the initial belt tension. Take this additional tension into consideration when you select the coupling belt.

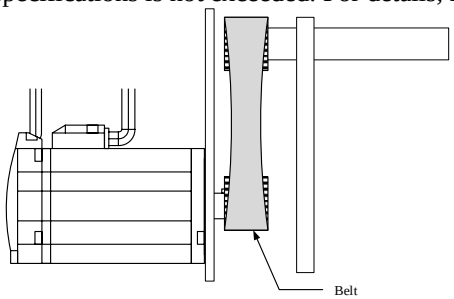
1. Wipe off all of the anticorrosive coating from the motor shaft.
2. If you are using a Servomotor with a Key, attach the key enclosed with the Servomotor or the specified size of key to the shaft.
3. If you need to attach a pulley to the Servomotor with a Key, use a screwdriver to tighten the screw in the end of the motor shaft to press in and attach the pulley.




4. Couple the Servomotor to the machine with a belt.
When you attach the belt, adjust the belt tension so that the allowable radial load given in the Servomotor

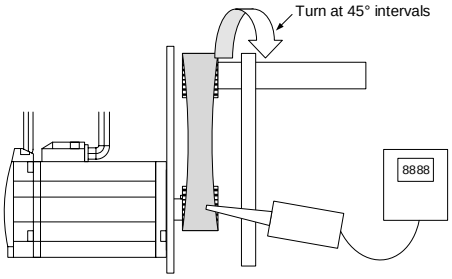
  

specifications is not exceeded. For details, refer to the catalog of the belt manufacturer.




IMPORTANT

Adjust the belt tension to adjust the radial load. Measure the belt tension at 45° intervals of the machine shaft. Turn the shaft and take measurements with a belt tension meter at each point.




# 2.2 Servodrive

## Storge

When the Servomotor is to be stored with the power cable disconnected, store it in the following temperature range:
Between -20 and 85℃.

## Environmental Conditions

- Ambient temperature is from 0℃ to 55℃.
- Ambient humidity is low than 90%RH, and free from condensation.
- Vibration is low than 4.9m/s$^{2.}$
- It is recommended the ambient temperature shall be below 45℃ to ensure the stable operation.

## Installation Sites

The following table lists some precautions on installation sites.

| Situation | Precautions on Installation |
|---|---|
| When installed in a control panel | Design the control panel size, unit layout, and cooling method so that the temperature around the periphery of the Servodrive shall not more than 55℃. |
| When installed near a heating unit | Suppress radiation heat from the heating unit and a temperature rise caused by convection so that the temperature around the periphery of the Servodrive shall not more than 55℃. |
| When installed near a source of vibration | Install a vibration isolator underneath the Servodrive to prevent it from receiving vibration. |
| When installed in a place receiving corrosive gases | Corrosive gases do not immediately affect the Servodrive but will eventually cause contactor-related devices to malfunction. Take appropriate action to prevent corrosive gases. |

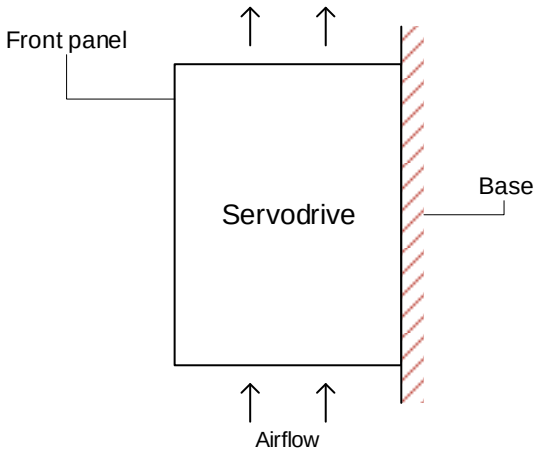
  

| Situation | Precautions on Installation |
|---|---|
| Others | Avoid installation in a hot and humid place or where excessive dust or iron powder is present in the air. |

## Mouting Orientation

As is shown in the following figure, the Servodrive is installed perpendicular to the base.

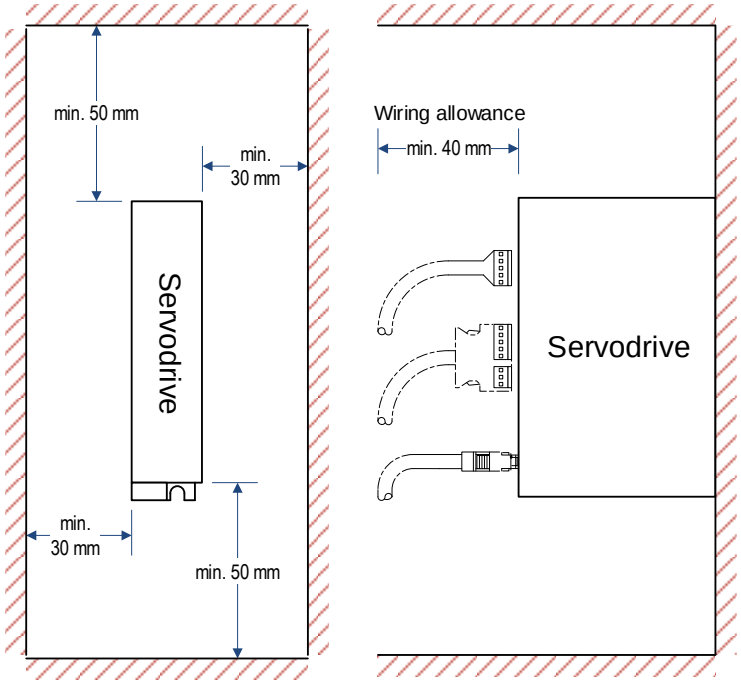
Prepare two mounting holes for the Servodrive and mount it securely in the mounting holes. In addition, let the front panel of the Serovdrive is facing toward the operator.




A fan can be added to force cooling the Servodrive if necessary.

## Mounting Interval

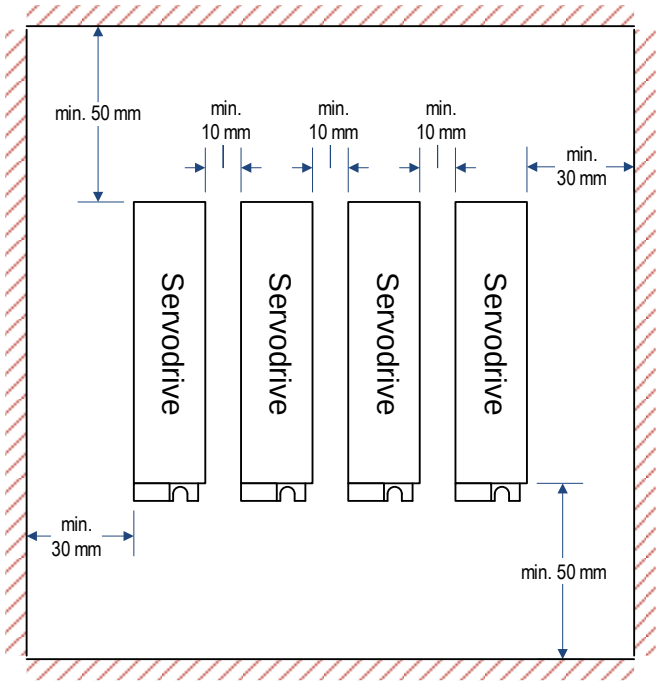
- When you install one Servodrive in the control panel, provide the following spaces around the Servodrive.




- When you install more than one Servodrive in the control panel, provide the following intervals between the Servodrives and spaces around the Servodrives.
  Install cooling fans above the Servodrives so that hot spots do not occur around the Servodrives.
  Provide sufficient intervals and spaces as shown in the following figure to enable cooling by the fans

and natural convection.



# Chapter 3 Wiring

## 3.1 Main Circuit Wiring


Please observe the following precautions when wiring.

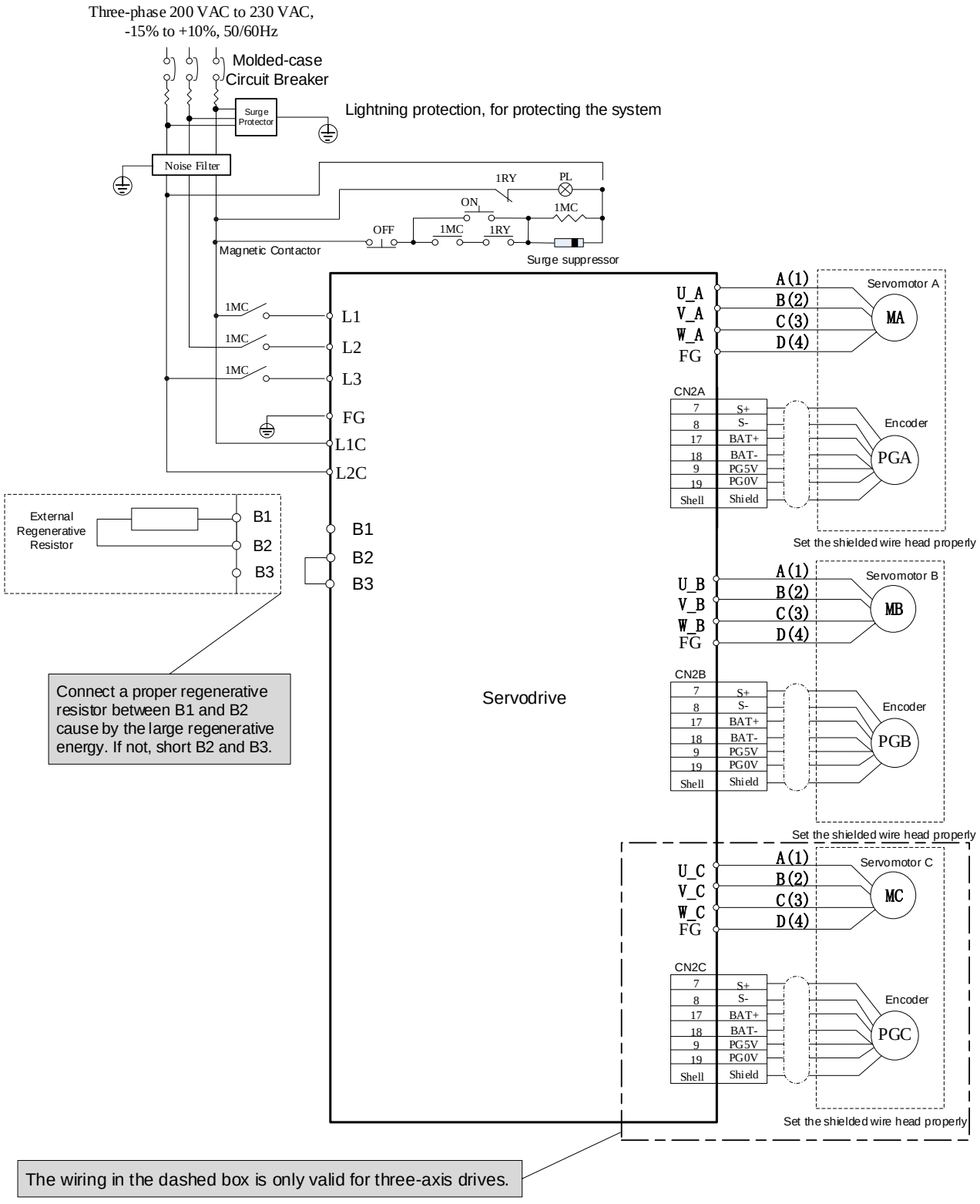
- Never bundle or run power and signal lines together in the same duct. Keep power and signal lines separated by at least 300 mm.
- Use twisted-pair shielded wires or multi-core twisted-pair shielded wires for signal and encoder feedback lines.
- The maximum length is 3 m for reference input lines and 20 m for encoder feedback lines.
- Never touch the power terminals for 5 minutes after turning power OFF because high voltage may still remain in the servo drive.

### 3.1.1 Names and Functions

| Symbol | Name | Functions |
|---|---|---|
| L1, L2, L3 | Main circuit power supply input terminal | Three-phase 200 VAC to 230 VAC, -15% to +10%, 50 Hz or 60 Hz |
| U_A, V_A, W_A | Servomotor terminals for axis A | Connect to the Servomotor of axis A. |
| U_B, V_B, W_B | Servomotor terminals for axis B | Connect to the Servomotor of axis B. |
| U_C, V_C, W_C | Servomotor terminals for axis C | Connect to the Servomotor of axis C. |
| L1C, L2C | Control power supply terminals | Single-phase 200 VAC to 230 VAC, -15% to +10%, 50 Hz or 60 Hz |
| ⏚ | Ground terminals | Connects to the power supply ground terminals and servomotor ground terminal. |
| B1, B2, B3 | External regenerative resistor connection terminal | If using an internal regenerative resistor, please short B2 and B3. Remove the wire between B2 and B3 and connect an external regenerative resistor (provided by customer) between B1 and B2, if the capacity of the internal regenerative resistor is insufficient. |

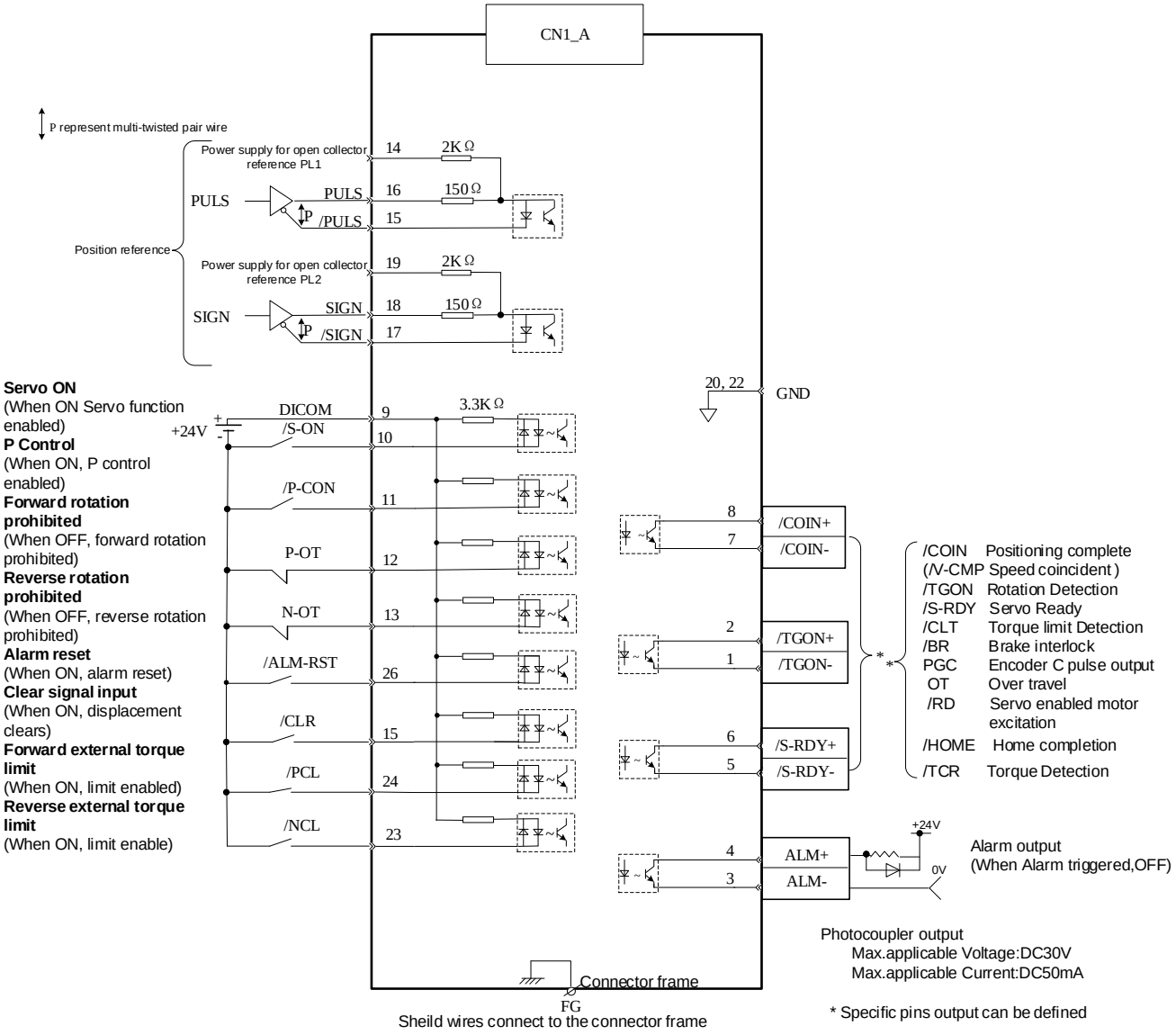
## 3.1.2 Wiring Example



# 3.2 Inputs and Outputs

## 3.2.1 Wiring Example




**NOTE**

- The wirings of CN1A, CN1B and CN1C are the same.
- The signal names mentioned in the figure for the I/O ports are all defined by the factory setting values. See the **Appendix A Parameters List** for reallocated them.

## 3.2.2 Terminal Layout

| Pin No. | Name | Description |
|---|---|---|
| 1<br>2<br>5<br>6<br>7<br>8 | 0: /COIN(/VCMP)<br>1: /TGON<br>2: /S-RDY<br>3: /CLT<br>4: /BK<br>5: PGC<br>6: OT<br>7: /RD<br>8: /HOME<br>9: /TCR | 0: Positioning completion (speed agree detection)<br>1: Running signal output<br>2: Servo ready<br>3: Torque limit output<br>4: Brake interlock output<br>5: C pulse output<br>6: Over travel signal output<br>7: Servo enabled motor excitation output<br>8: Home completion output<br>9: Torque detection output |
| 3 | ALM- | Servo alarm:Turns off when an error is detected. |
| 4 | ALM+ | |
| 9 | DICOM | Control power supply input for I/O signals:<br>Provide the +24V DC power supply |
| 10<br>11<br>12<br>13<br>23<br>24<br>25<br>26 | 0: /S-ON<br>1: /P-CON<br>2: P-OT<br>3: N-OT<br>4: /ALM-RST<br>5: /CLR<br>6: /PCL<br>7: /NCL<br>8: /G-SEL<br>9: /JDPOS-JOG+<br>A: /JDPOS-JOG-<br>B: /JDPOS-HALT<br>C: Reserved<br>D: SHOME<br>E: ORG(ZPS) | 0: Servo ON<br>1: P/PI control input<br>2: Forward run prohibited<br>3: Reverse run prohibited<br>4: Alarm reset<br>5: Position error pulseclear input<br>6: Forward torque limitinput<br>7: Reverse torque limitinput<br>8: External switch gain switching<br>9: Position control (contact reference) -forward direction JOG<br>A: Position control (contact reference) -reverse direction JOG<br>B: Position control (contact reference) -stop JOG<br>C: Reserved<br>D: Hometrigger<br>E: Zero position |
| 14 | PPIP | Collector open circuit power supply |
| 19 | PPIS | |
| 15 | PULS- | Pulse signal |
| 16 | PULS+ | |
| 17 | SIGN- | Direction signal |
| 18 | SIGN+ | |
| 20, 22 | GND | Digital Ground |
| Shell | FG | Frame Ground |

**NOTE**

- The list of CN1A, CN1B and CN1C about I/O Signal Names and Functions are the same.
- Spare terminals can not be used for relay purpose.
- Connect shielded cable wires of I/O signals to connector shell (frame ground).

## 3.2.3 Names and Functions

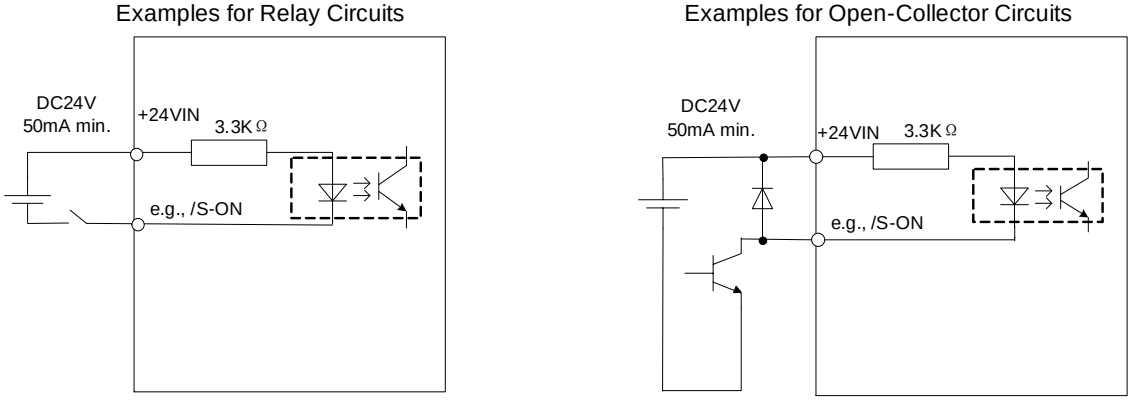
| Name | Terminal No. | Function | |
|---|---|---|---|
| DICOM | 9 | Control power supply input for I/O signals: Provide the +24V DC power supply. | |
| /S-ON | 10 | Servo ON:Turns the servomotor on. | The function of I/O are default, it can be changed by setting parameters. |
| /P-CON | 11 | It has deferent means depends on deferent control mode. | |
| P-OT | 12 | Forward run prohibited | |
| N-OT | 13 | Reverse run prohibited | |
| /ALM-RST | 26 | Alarm reset: Releases the servo alarm state. | |
| /CLR | 25 | Positional error pulse clear input: Clear the positional error pulse during position control. | |
| /PCL | 24 | Forward external torque limit | |
| /NCL | 23 | Reverse external torque limit | |
| PPIP | 14 | Power supply input for open collector reference (pulse) | |
| PPIS | 19 | Power supply input for open collector reference (direction) | |
| PULS- | 15 | Reference pulse input<br>Reference sign input | Pulse reference input mode:<br>• Sign + pulse train<br>• CCW + CW pulse<br>• Two-phase pulse |
| PULS+ | 16 | | |
| SIGN- | 17 | | |
| SIGN+ | 18 | | |
| /COIN- (/V-CMP-) | 7 | Positioning completion(Speed coincidence):<br>Turns ON when the number of positional error pulses reaches the value set. | The functions of I/O are default; you can change them by setting parameters. |
| /COIN+ (/V-CMP+) | 8 | | |
| /TGON- | 1 | Motor rotation detection: when the servomotor is rotating at a speed higher than the motor speed setting. | |
| /TGON+ | 2 | | |
| /S-RDY- | 5 | Servo ready: turn ON if there is no servo alarm when the control/main circuit power supply is turned ON. | |
| /S-RDY+ | 6 | | |
| ALM- | 3 | Servo alarm: Turns off when an error is detected. | |
| ALM+ | 4 | | |
| GND | 20, 22 | Grounding | |
| FG | Shell | Connect frame to ground if the shield wire of the I/O signal cable is connected to the connector shell. | |

## 3.2.4 I/O Circuits

### Sequence Input Circuits

Examples for Relay Circuits and Open-Collector Circuits are as shown in the following figure.




Select a low-current relay for the relay circuits, or a faulty contact may be caused.

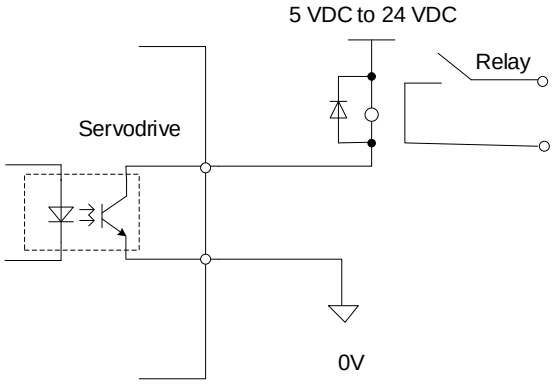
### Encoder Output Circuits

The signals of the encoder pulses (PAO+, PAO-, PBO+, PBO-) and origin signal pulses (PCO+, PCO-) are output by the bus drive circuits. The circuits are usually used in the host controller for forming a position control system, which connects a encoder output circuit to line-receiver circuit.

For details ahout the encoder circuit, see the section 3.3 Encoder Wiring.

### Sequence Output Circuits

Photocoupler output circuits are used for the ALM (Servo Alarm), /P-CON (Position Complted), /BK (Brake Interlock) and other sequence output signals.




**CAUTION**

The maximum allowable voltage and current range for photocoupler output circuits are as follows:

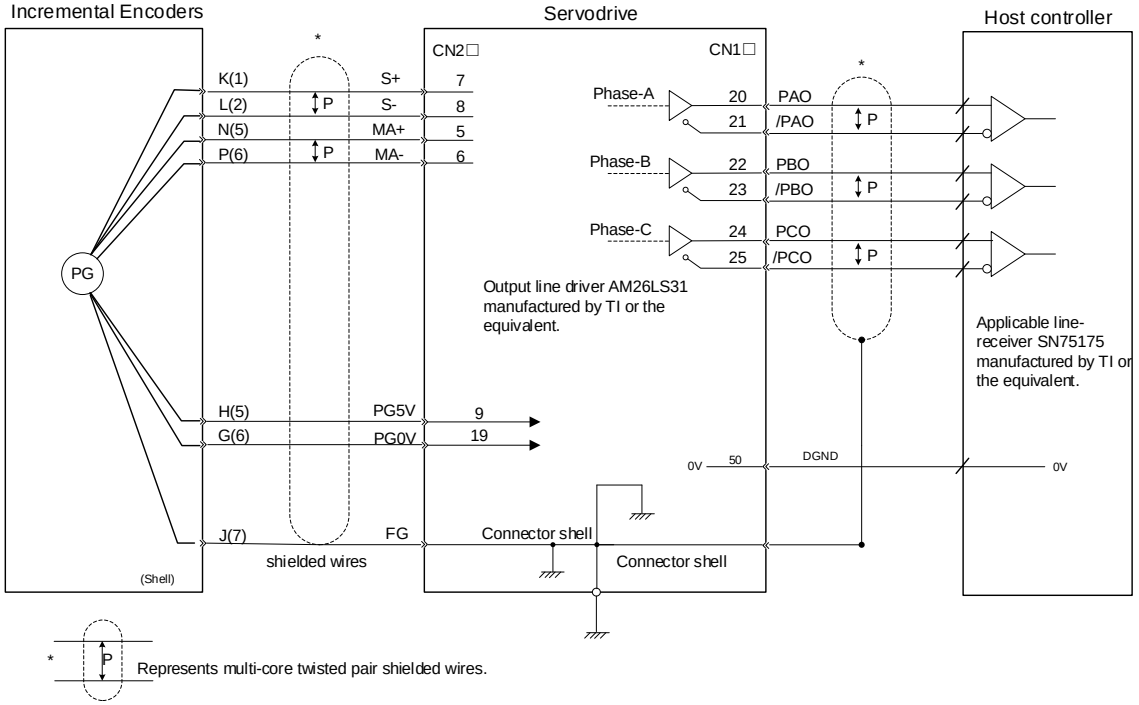
- Maximum allowable voltage: 30 VDC
- Current range: 5 mA to 50 mA DC

# 3.3 Encoder Wiring

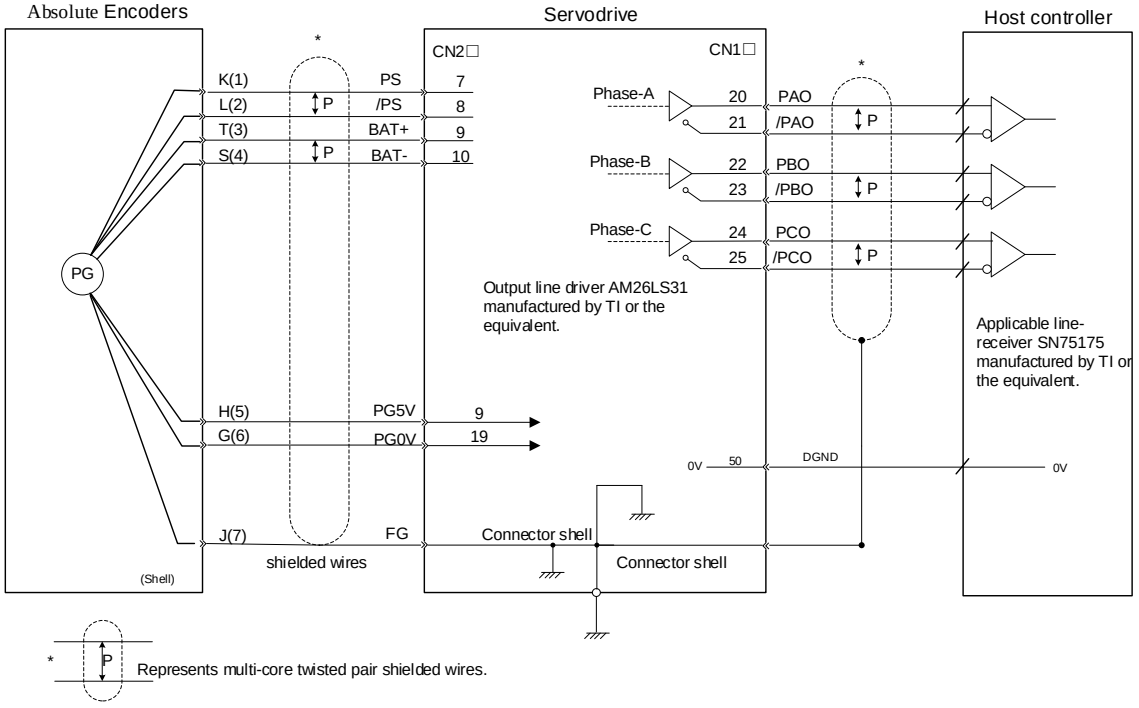
## 3.3.1 Wiring Diagram

### Incremental Encoder




**NOTE**: The pin numbers for the connector wiring differ depending on the servomotors.

### Absolute Encoder




**NOTE**: The pin numbers for the connector wiring differ depending on the servomotors.

### 3.3.2 Terminal Layout

**Incremental Encoder**

| Pin No. | Name | Description |
|---|---|---|
| 1 | PG5V | PG power supply (+5V) |
| 2 | PG0V | PG power supply (0V) |
| 5 | MA+ | PG serial signal output |
| 6 | MA- | PG serial signal output |
| 7 | S+ | PG serial signal input |
| 8 | S- | PG serial signal input |

NOTE: Other pins are vacant.

**Absolute Encoder**

| Pin No. | Name | Description |
|---|---|---|
| 1 | PG5V | PG power supply (+5V) |
| 2 | PG0V | PG power supply (0V) |
| 7 | PS | PG serial signal input |
| 8 | /PS | PG serial signal input |
| 9 | BAT+ | Battery (+) |
| 10 | BAT- | Battery (-) |

NOTE: Other pins are vacant.

## 3.4 Communication Wiring

| Pin No. | Name | Description |
|---|---|---|
| 1 | – | Reserved |
| 2 | – | |
| 3 | 485+ | RS-485 communication terminal (+) |
| 4 | GND | Grounding |
| 5 | GND | |
| 6 | 485- | RS-485 communication terminal (-) |
| 7 | CANH | CAN communication terminal (High level) |
| 8 | CANL | CAN communication terminal (Lowe level) |

# 3.5 Wiring for Noise Control

## 3.5.1 Noise Control

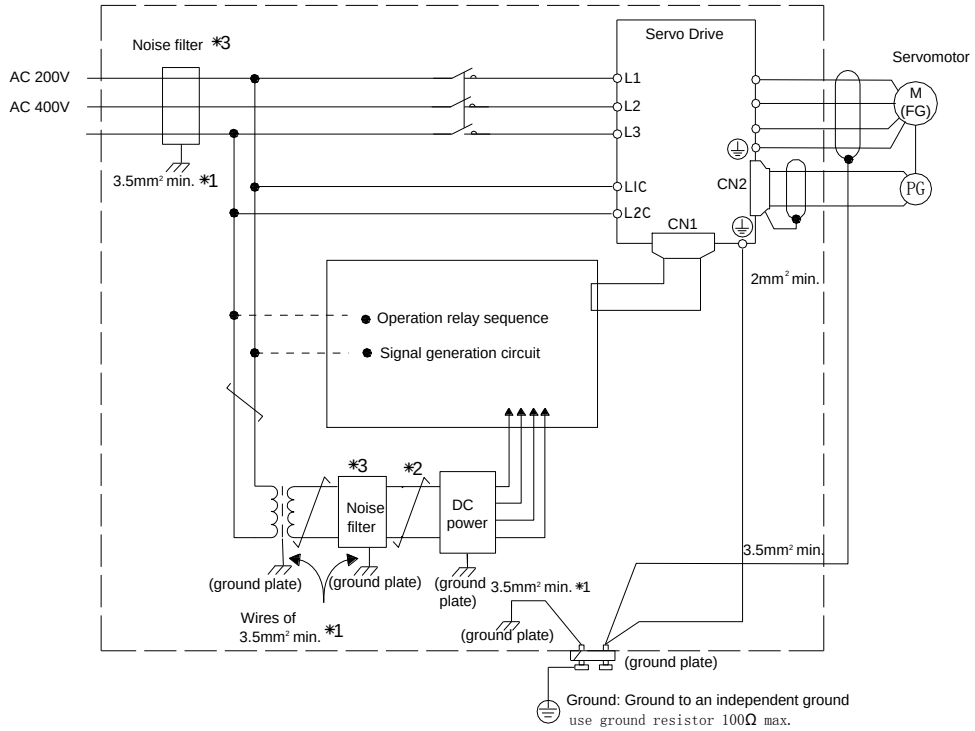
The servodrive uses high-speed switching elements in the main circuit. It may receive "switching noise"from these high-speed switching elements.

To prevent malfunction due to noise, take the following actions:

- Position the input reference device and noise filter as close to the Servodrive as possible.
- Always install a surge absorber in the relay, solenoid and electromagnetic contactor coils.
- The distance between a power line (servomotor main circuit cable) and a signal line must be at least 30 cm.Do not put the power and signal lines in the same duct or bundle them together.
- Do not share the power supply with an electric welder or electrical discharge machine. When the servo drive is placed near a high-frequency generator, install a noise filter on the input side of the power supplyline. As for the wiring of noise filter, see the seciton **Noise Filter**.
- For proper grounding technique, see the section **Correct Grounding.**

### Noise Filter

Please install a noise filter in the appropriate place to protect the servo drive from external noise interference.




When using a noise filter, always observe the following wiring instructions:

- For a ground ground wire to be connected to the casing, use a thick wire with a thickness of at least 3.5 $mm^2$ (preferably, plain stitch cooper wire).
- For wires indicated by P↕, use twisted-pair cables whenever possible.

### Correct Grounding

Take the following grounding measures to prevent the servo drive from malfunctioning due to noise.

- Grounding the Motor Frame

  If the servomotor is grounded via the machine, a switching noise current will flow from the servo drive main circuit through the servomotor stray capacitance.

Always connect servomotor frame terminal FG to the servodrive ground terminal. Also be sure to ground the ground terminal ⏚

- Noise on the I/O Signal Line

  If the I/O signal line receives noise, ground the 0 V line (SG) of the reference input line. If the main circuit wiring for the motor is accommodated in a metal conduit, ground the conduit and its junction box. For all grounding, ground at one point only.

**Precautions on installing on the control panel**

- When the servo drive is installed on the control panel, a piece of metal plate should be fixed. It is used for fixing the servo drive and other peripheral devices. The noise filter should be installed on the metal plate, and closed to the hole drill through power lines on control panel. Use screws to fix the noise filter to the metal plate. The grounding terminals of noise filter connects to the grounding terminals of control panel.
- Servo drive should be fixed on a piece of metal plate. Make sure the heat sink towards ground. The grounding terminals of servo drive connect to the grounding terminals of control panel.

## 3.5.2 Precautions on Connecting Noise Filter

**Noise Filter Brake Power Supply**

Use the noise filter Manufactured by SCHAFFNER at the brake power input for servomotors with holding brakes.

Relationship between servo drive power and noise filter current:

| Servomotor Power | Noise Filter Current for single motor |
|---|---|
| 50W | 1.5A |
| 100W | 1.5A |
| 200W | 2A |
| 400W | 3A |
| 750W | 5A |
| 1.0kW | 6A |

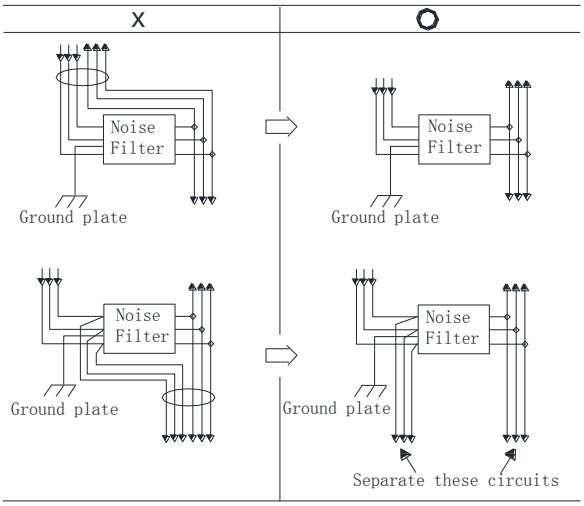
NOTE

- A single-phase servomotor should apply a two-phase filter. A three-phase servo drive should apply a three-phase filter.
- Choose the right filter according the specifications of operating voltage, current, and manufacturer.

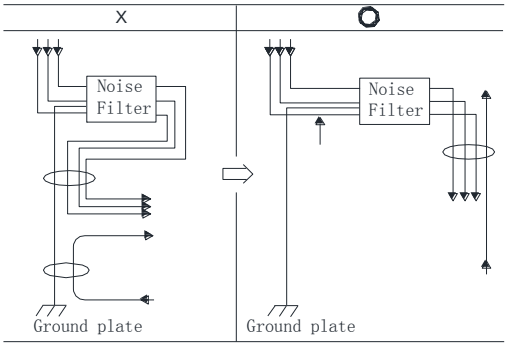
  

## Precautions on Using Noise Filters

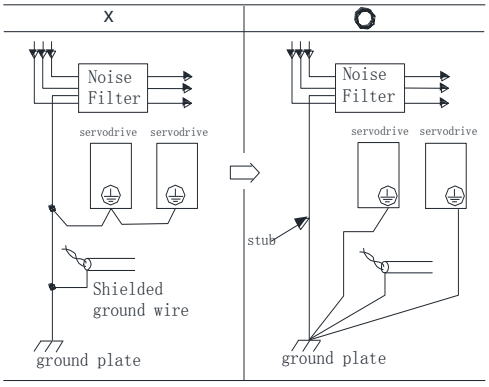
- Do not put the input and output lines in the same duct or bundle them together.




- Separate the noise filter ground wire from the output lines.
  Do not accommodate the noise filter ground wire, output lines and other signal lines in the same duct or bundle them together.

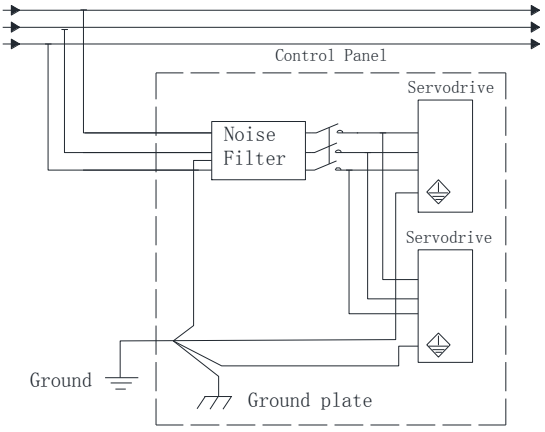



- Connect the noise filter ground wire directly to the ground plate. Do not connect the noise filter ground wire to other ground wires.




- If a noise filter is located inside a control panel, connect the noise filter ground wire and the ground wires from other devices inside the control panel to the ground plate for the control panel first, then ground these wires.



# Chapter 4 Panel Operator

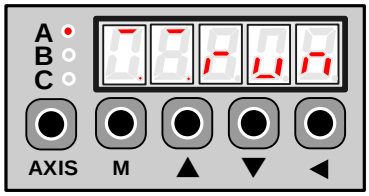
## 4.1 Basic Operation

### 4.1.1 Functions on Panel Operator

The panel operator is a built-in operator that consists of display section and keys located on the front panel of the servo drive.

Parameter setting, status display ,and execution of utility function are enabled using the panel operator.

The names and functions of the keys on the panel operator are shown as follows:




| Symbol | Name | Description |
|---|---|---|
| AXIS | Axis key | Under the 1st-level menu, press the [**AXIS**] key to switch the control among axis-A, axis-B and axis-C. |
| ▲ | INC key | Press these keys to choose the desired parameters or set the value of the parameters. |
| ▼ | DEC key | |
| M | Mode key | Press [**M**] key to switch the mode among Status Display, Parameter Setting, Monitor and Utility Function.<br>In addition, Press [**M**] key to save the setting of the parameter value and then back to the Parameter Setting mode. |
| ◄ | Enter key | Press [◄] key to display the parameters and values, and release the alarm. |

### 4.1.2 Axis Switching

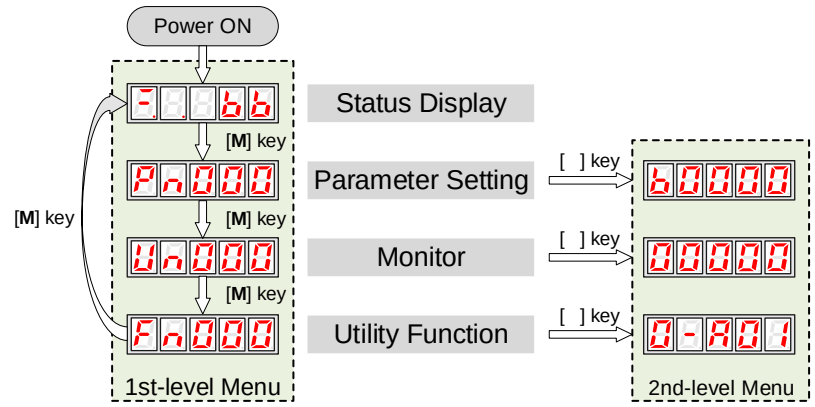
Since the Servodrive only can set and monitor only one axis at a time, the use can press the [**AXIS**] key to switch the control into another axis. The indicator lamp behind each axis can show the work status:

- Lit indicates the axis is controlled at present.
- Not lit indicates the axis is not controlled at present.
- Blinking indicates an alarm occurred in the axis not controlled at present.

### 4.1.3 Mode Switching

As is shown in the following figure, press [**M**] key can switch the mode among Status Display, Parameter Setting, Monitor and Utility Function in turns.



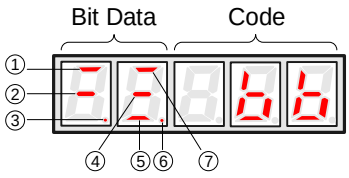

**Note**: Only under the 1st-level menu, press the [**AXIS**] key to switch the control among axis-A, axis-B and axis-C.

# 4.2 Status Display

The status display mode displays the servo drive status as bit data and codes.

The status display mode is selected when the power supply is turned ON. If it is not displayed, select this mode by pressing [**M**] key.

Note that the display differs between the speed/torque control and position control.




The following table lists the description of each bit data in speed/torque control and position control.

| No. | Speed/Torque Control | | Position Control | |
|---|---|---|---|---|
| | **Bit Data** | **Description** | **Bit Data** | **Description** |
| ① | Speed Coincidence | Lit when the difference between the servomotor and reference speed is the same as or less than the preset value.<br>Preset value:Pn501 (factory setting is 10 rpm)<br>Always lit in torque control mode. | Positioning Completion | Lit if error between position reference and actual servomotor position is below preset value.<br>Preset value: Pn500 (factory setting is 10 pulse). |
| ② | Base lock | Lit for base block.<br>Not lit at servo ON. | Base lock | Lit for base block.<br>Not lit at servo ON. |
| ③ | Control power ON | Lit when servo drive control power is ON. | Control power ON | Lit when servo drive control power is ON. |
| ④ | – | Always not lit. | Reference pulse input | Lit if reference pulse is input.<br>Not lit if no reference pulse is input. |
| ⑤ | Torque reference input | Lit if input torque reference exceeds preset value.<br>Not lit if input torque reference is below preset value.<br>Preset value: 10％ of rated torque | Deviation counter clear signal input | Lit when deviation counter clear signal is input.<br>Not lit when deviation counter clear signal is not input. |

| No. | Speed/Torque Control | | Position Control | |
|---|---|---|---|---|
| | Bit Data | Description | Bit Data | Description |
| ⑥ | Power ready | Lit when main circuit power supply is ON and normal.<br>Not lit when main circuit power supply is OFF. | Power ready | Lit when main circuit power supply is ON and normal.<br>Not lit when main circuit power supply is OFF. |
| ⑦ | Rotation detection /TGON | Lit if servomotor speed exceeds preset value.<br>Not lit if servomotor speed is below preset value.<br>Preset value: Pn503 (factory setting is 20 rpm) | Rotation detection /TGON | Lit if servomotor speed exceeds preset value.<br>Not lit if servomotor speed is below preset value.<br>Preset value: Pn503 (factory setting is 20 rpm) |

The following table lists the description of code.

| Code | Description |
|---|---|
|  | Base lock<br>Servo OFF (motor power OFF) |
|  | Run<br>Servo ON (motor power ON) |
|  | Forward Run Prohibited<br>CN1□-10 (P-OT) is OFF |
|  | Reverse Run Prohibited<br>CN1□-10 (N-OT) is OFF |
|  | Alarm Status<br>Displays the alarm number |

NOTE: Press [◄] key to try clearing the current alarms.

## 4.3 Operation in Parameter Setting Mode

The servo drive offers a large number of functions, which can be selected or adjusted by the parameter settings. For details about each parameters description see the section **Appendix A Parameters List**.

The following procedure is an example for changing the setting of parameter Pn102 from 100 to 85.

**Step 1** Press [**M**] key for several times to switch into Parameter Setting mode, after turning the Servodrive ON.

**Step 2** Press [▲] key or [▼] key to select the parameter Pn102.

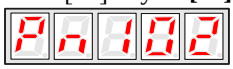
**Note**: press and hold [▲] key or [▼] key to jump the parameter number quickly.

**Step 3** Press [◄] key to display the current value of Pn102.

**Step 4** Press [▼] key or [▼] key to change the value to 00085.
Press and hold [▼] key or [▼] key to jump the setting value quickly.

**Step 5** Press [◄] key or **[M]** key to return to the display of the current parameter.



**----End**

# 4.4 Operation in Monitor Mode

The monitor mode allows the reference values input into the servo drive, I/O signal status, and servo drive internal status to be monitored.

## Using the Monitor Mode

在 The example below shows how to display the value (1500) stored in Un001.

**Step 1** Press [**M**] key for several times to switch into Parameter Setting mode, after turning the Servodrive ON.

**Step 2** Press [▲] key or [▼] key to select the monitor number Un001.

**Step 3** Press [◄] key to display the current value of Un001.

**Step 4** Press [◄] key again to return to the display of the current monitor number.

**----End**

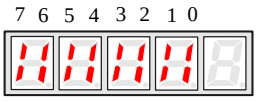
## Descriptions of Monitor Number

| Monitor Number | Description |
|---|---|
| Un000 | Actual servomotor speed. Unit: rpm |
| Un001 | Reserved |
| Un002 | Reserved |
| Un003 | Internal torque reference (relative to the rated torque). Unit: % |
| Un004 | Number of encoder rotation angle pulses |
| Un005 | Input signal monitor |
| Un006 | Encoder signal monitor |
| Un007 | Output signal monitor |
| Un008 | Frequency given by pulse. Unit:1kHZ |
| Un009 | Number of servomotor rotation pulses |
| Un010 | Pulse rate of servomotor rotated ($x10^4$) |
| Un011 | Low 16 bits of the pulse deviation counter |

| Monitor Number | Description |
|---|---|
| Un012 | High 16 bits of the pulse deviation counter |
| Un013 | Number of pulses given |
| Un014 | Number of pulses given ($x10^4$) |
| Un015 | Load inertia percentage |
| Un016 | Servomotor overload ratio |
| Un017 | Bus voltage .Unit: V |

In which, the display meaning of Un005, Un006 and Un007 are shown as following table.

| Displayed | Monitor Number | Meaning |
|---|---|---|
|  | Un005 | 0: /SON<br>1: /P-CON<br>2: P-OT<br>3: N-OT<br>4: /ALM-RST<br>5: /CLR<br>6: /PCL<br>7: /NCL |
| | Un006 | 0: (Not used)<br>1: (Not used)<br>2: (Not used)<br>3: (Not used)<br>4: C 相<br>5: B 相<br>6: A 相<br>7: (Not used) |
| | Un007 | 0: ALM<br>1: /COIN<br>2: /TGON<br>3: /S-RDY |

## 4.5 Operation in Utility Function Mode

In utility function mode, the panel operator can be used to run and adjust the servo drive and servomotor.

The following table lists the functions in the utility function mode.

| Function Number | Description |
|---|---|
| Fn000 | Alarm traceback data display |
| Fn001 | Parameter setting initialization |
| Fn002 | JOG mode operation |
| Fn003 | Reserved |
| Fn004 | Reserved |
| Fn005 | Automatic adjustment of Servomotor current detection |
| Fn006 | Manual adjustment of Servomotor current detection |

| Function Number | Description |
|---|---|
| Fn007 | Software version display |
| Fn008 | Position teaching |
| Fn009 | Moment of Inertia Estimation |
| Fn010 | Reserved |
| Fn011 | Reserved |
| Fn012 | Reserved for the Manufacturer |
| Fn013 | Parameters copying |
| Fn014 | Reserved |

### 4.5.1 Alarm Traceback Data Display

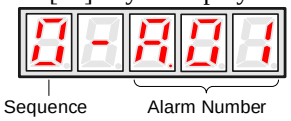
The alarm traceback display can display up to 10 previously occurred alarms.The alarm is displayed on Fn000, which is stored in the alarm traceback data.

Follow the procedures below to confirm alarms which have been generated.

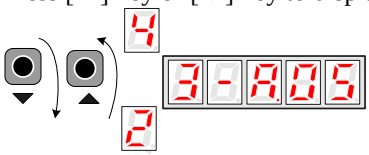
**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the Servodrive ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn000.

**Step 3** Press [◀] key to display the latest alarm number.




**Step 4** Press [▲] key or [▼] key to display another alarm numbers occurred recently.



**Step 5** Press [◀] key to return to the display of the current function number.

**----End**

### 4.5.2 Parameter Settings Initialization

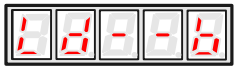
Follow the procedures below to perform the parameter settings initialization.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the Servodrive ON.

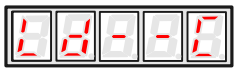
**Step 2** Press [▲] key or [▼] key to select the function number Fn001.

**Step 3** Press [◀] key to prepare for initializing the parameter settings.

Initialize the parameter settings of axis-A

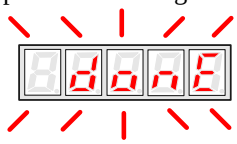


Initialize the parameter settings of axis-B



Initialize the parameter settings of axis-C

**Step 4** Press and hold [◀] key for 1 second or more until "done" is displayed and blinked, which indicates the parameter settings initialization has been completed.




**Step 5** Relase [◀] key to return to the display of the current function number.

**CAUTION**

The parameter settings initialization can not be performed when servo is turned ON.
Turn OFF the servo before this operation.

**----End**

## 4.5.3 JOG Operation

Follow the procedures below to operate the Servomotor in JOG.

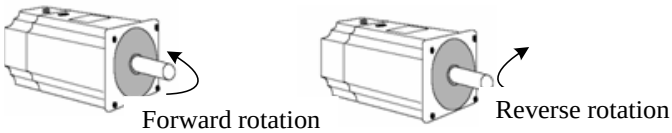
**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn002.


**Step 3** Press [◀] key to enter the JOG running mode.


**Step 4** Press **[M]** key to turn ON the servo.
Here, you can press **[M]** key to turn ON or turn OFF the servo. However, Turn ON the servo if you want to run the Servomotor.

**Step 5** Press [▲] key or [▼] key to run the Servomotor forward or reverse in a certain amount of movement.
Press and hold [▲] key or [▼] key to run the Servomotor continuously.


Forward rotation Reverse rotation

**NOTE**: the rotation direction of the Servomotor depends on the setting of Pn001.0. The above figure shows the default setting.

**Step 6** Relase [◀] key to return to the display of the current function number.
Moreover, the servo is turned OFF automatically.

**----End**

## 4.5.4 Offset Adjustment for Current Detection

Since the Offset Adjustment for Current Detection has been performed before the device leaves the factory, the user does not need to perform this operation generally.

However, Offset Adjustment for Current Detection shall be peformmed when the torque ripple was too large or if you want to further reduce the torque ripple.

This section describes the automatic and manual offset adjustment for current detection.

**CAUTION**

- Offset Adjustment for Current Detection is only can be performed at Servo OFF.
- If this function is inadvertently enabled, especially by manually adjustment, the characteristic deterioration will be occurred.
- It is necessary to perform the Offset Adjustment for Current Detection if the torque ripple is significantly too large.

### Automatic Adjustment

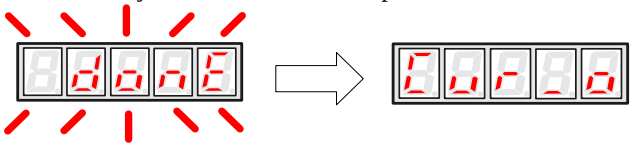
Following the below procedure to perform the automatic adjustment.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn005.


**Step 3** Press [◀] key to prepare for performing the automatic adjustment.


**Step 4** Press and hold [**M**] key for 1 second or more until “done” is displayed and blinked, which indicates the automatic adjustment has been completed.




**Step 5** Press [◀] key to return to the display of the current function number.

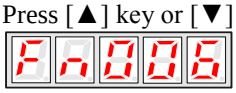
**----End**

### Manual Adjustment

Following the below procedure to perform the manual adjustment.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

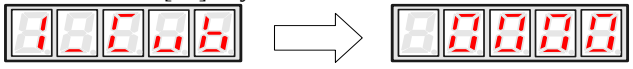
**Step 2** Press [▲] key or [▼] key to select the function number Fn006.




**Step 3** Press [◀] key to prepare for performing the manual adjustment.

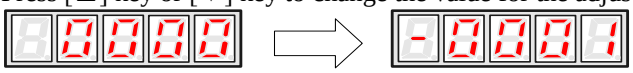

**Step 4** Press **[M]** key to select the desired setting phase between phase-U (o_CuA) and phase-V (1_Cub).


**Step 5** Press and hold [◀] key for 1 second, the detection value of current phase is displayed.



**Step 6** Press [▲] key or [▼] key to change the value for the adjustment.



**Step 7** Press and hold [◄] key for 1 second to return to the setting phase.

**Step 8** Press [◄] key to return to the display of the current function number.

**CAUTION**

The adjusting range of the current detection offset is from -100 to +100.

**----End**

## 4.5.5 Software Version Display

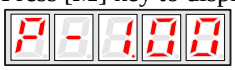
Following the below procedure to perform the Software Version Display.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn007.

**Step 3** Press [◄] key to display the version of DSP software. (The leftmost bit is t).

**Step 4** Press [**M**] key to display the version of FPGA/CPLD software. (The leftmost bit is P).



**Step 5** Press [**M**] key to return to the version of DSP software displayed.

**Step 6** Press [◄] key to return to the display of the current function number.

**----End**

## 4.5.6 Position Teaching Function

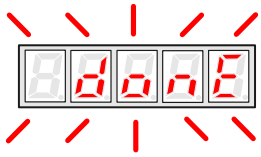
Following the below procedure to perform the postion teaching.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn008.

**Step 3** Press [◄] key to prepare for the position teaching.

**Step 4** Press and hold [◄] key for 1 second or more until “done” is displayed and blinked, which indicates the position teaching has been completed.



**Step 5** Relase [◄] key to return to the display of the current function number.

**----End**

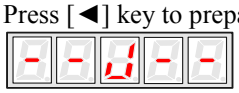
## 4.5.7 Moment of Inertia Estimation

Following the below procedure to perform the Moment of Inertia Estimation.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn009.


**Step 3** Press [◀] key to prepare for the Moment of Inertia Estimation.




**Step 4** Press [**M**] key for performing the operation.
The real-time rotational speed of the Servomotor is displayed.

**Step 5** When the Servomotor is stopped, the estimation value is displayed on the panel operator, and its unit is $kg.cm^2$.


Make sure the Servomotor has completed at least 6 full revolutions in the CCW direction.

**----End**

## 4.5.8 Parameters Copying

Following the below procedure to perform the Parameters Copying.

**Step 1** Press [**M**] key for several times to switch into Utility Function mode, after turning the servo ON.

**Step 2** Press [▲] key or [▼] key to select the function number Fn013.

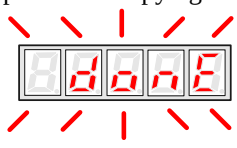

**Step 3** Press [◀] key to prepare for the parameters copying.


PnA-b, meaning the parameter in axis-A will be copied to axis-B.


Pnb-A, meaning the parameter in axis-B will be copied to axis-A.

The meaning of the other displays are similar to the above.

**Step 4** Press and hold [◀] key for 1 second or more until “done” is displayed and blinked, which indicates the parameter copying has been completed.




**Step 5** Relase [◀] key to return to the display of the current function number.

**----End**

# Chapter 5 Operation without CANopen

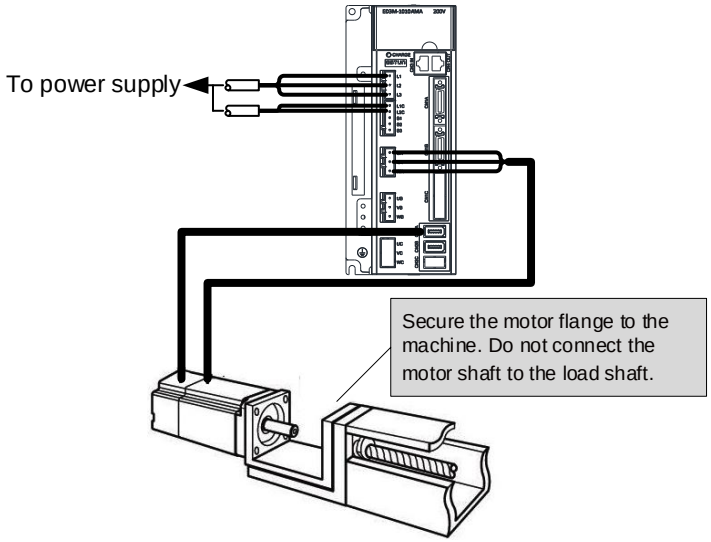
## 5.1 Trial Operation

### 5.1.1 Flow of Trial Operation

Make sure that all the wiring has been completed before the trial operation.

Perform the following three methods of trial operation in order. Instructions are given for speed control mode (standard setting) and position control mode. Unless otherwise specified, the standard parameters for speed control mode (factory settings) are used.

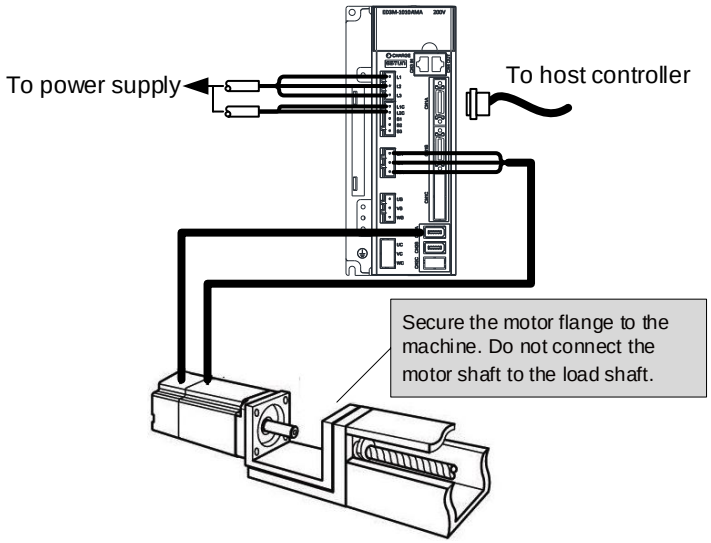
**Trial Operation for Servomotor Without Load**




The servomotor is operated without connecting the shaft to the machine in order to confirm the following wiring is correct.

- Power supply circuit wiring
- Servomotor wiring
- Encoder wiring
- Rotation direction and speed of servomotor.

**Trial operation for servomotor with host reference**

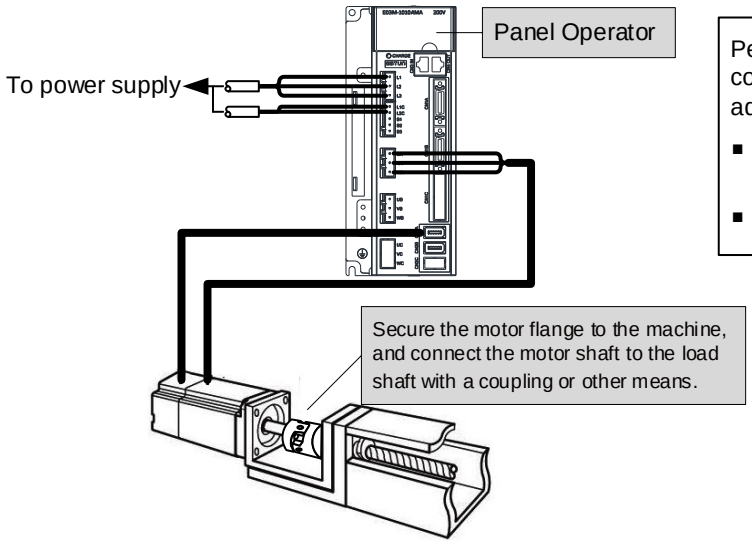



The servomotor is operated without connecting the shaft to the machine in order to confirm the following wiring is correct.

- I/O signal wiring with host controller
- Rotation direction, speed and number of rotations of servomotor
- Check the operation of the brake, overtravel and other protective functions

## Trial operation for servomotor and machine combined




Perform the trial operation with the servomotor connected to the machine. The servo drive is adjusted to match the machine characteristics.

- Servomotor speed and machine travel distance
- Set the necessary parameters

## Procedure for the trial operation

**Step 1 Installation**
Install the servomotor and servo drive according to the installation conditions. (Do not connect the servomotor to the machine because the servomotor will be operated first under the no-load condition for checking.)

**Step 2 Wiring**
Connect the power supply circuit (L1, L2 and L3), servomotor wiring (U, V, W), I/O signal wiring (CN1☐), and encoder wiring (CN2☐). But during Trial Operation for Servomotor Without Load, disconnect the CN1☐ connector.

**Step 3 Turn the power ON**
Turn the power ON. Using the panel operator to make sure that the servo drive is running normally. If using a servomotor equipped with an absolute encoder, please perform the setup for the absolute encoder.

**Step 4 Perform the JOG operation**
Perform the JOG operation with the servomotor alone under the no-load condition.

**Step 5 Connect input signals**
Connect the input signals (CN1☐) necessary for trial operation to the Servodrive.

**Step 6 Check input signals**
Use the internal monitor function to check the input signals.
Turn the power ON, and check the emergency stop, brake, overtravel, and other protective functions for the correct operation.

**Step 7 Input the Servo-ON signal**
Input the Servo-ON signal, and turn ON the servomotor.

**Step 8 Input reference**
Input the reference necessary for control mode, and check the servomotor for correct operation.

**Step 9 Protective operation**
Turn the power OFF, and connect the servomotor to the machine.
If using a servomotor equipped with an absolute encoder, set up the absolute encoder and make the initial settings for the host controller to match the machine's zero position.

**Step 10 Set necessary parameters**
Using the same procedure as you did to input a reference in step 8,operate the servomotor via the host controller and set the parameter to make sure the machine's travel direction, travel distance, and travel speed allcorrespond to the reference.

**Step 11** **Run**

The servomotor can now be operated. Adjust the servo gain if necessary.

**----End**

## 5.1.2 Trial Operation for Servomotor Without Load

**CAUTION**

Release the coupling between the servomotor and the machine, and secure only the servomotor without a load.

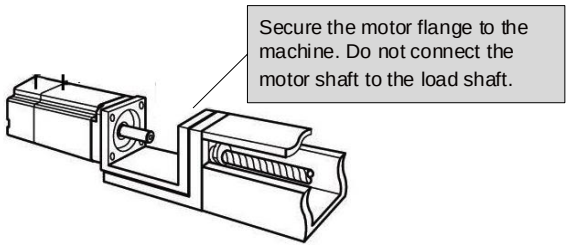
To prevent accidents, initially perform the trial operation for servomotor under no-load conditions (with all couplings and belts disconnected).

In this section, confirm the cable connections of the main circuit power supply, servomotor and encoder. Incorrect wiring is generally the reason why servomotors fail to operate properly during the trial operation.

Confirm the wiring, and then conduct the trial operation for servomotor without load according to the following steps.

**Step 1** Secure the servomotor.

- Secure the servomotor flange to the machine in order to prevent the servomotor frommoving during the operation.
- Do not connect the servomotor shaft to the machine. The servomotor may tip over during rotation.



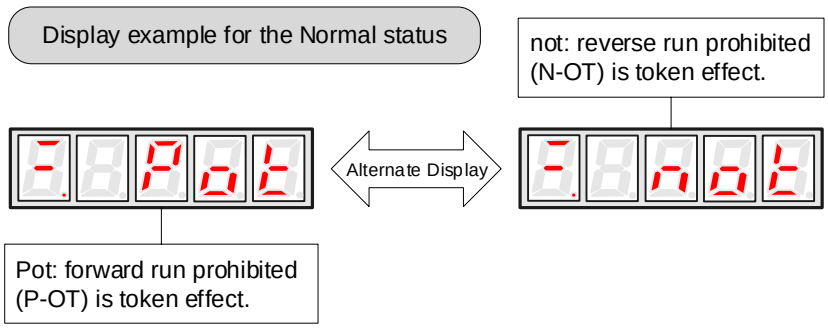

**Step 2** Check the power supply circuit, servomotor, and encoder wiring.

With the I/O signal connector (CN1□) disconnected, check the power supply circuit and Servomotor wiring.

See the section 3.1 Main Circuit Wiring for the details about the wiring of the main circuit.

**Step 3** Turn ON the control power supply and main circuit power supply.

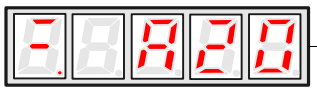
If the power is correctly supplied, the panel operator display on the front panel of the Servodrive will appear as shown on the left. The display on the left indicates that forward run prohibited (P-OT) and reverse run prohibited (N-OT).




If an alarm display appears, the power supply circuit, servomotor wiring, or encoder wiring is incorrect. If an alarm is displayed, turn OFF the power, find the problem, and correct it.

Display example for the Alarm status




Alarm A20 (Power Supply Line Open Phase) occurred

**Step 4** When using a servomotor with a brake, release the brake first before driving the servomotor.

**Step 5** Use the panel operator to operate the servomotor with utility function Fn002 (JOG Operation).

Check the Servomotor rotates in the forward direction by pressing **[▲]** key, and reverse direction by pressing **[▼]** key.

The operation is completed when the operation is performed as described below and no alarm occurs.

Complete the Fn002 (JOG Mode Operation) and turn OFF the power.

The servomotor speed can be changed using the Pn305 (JOG Speed). The factory setting for JOG speed is 500 rpm.

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn305** | JOG Speed | 0 to 6000 | rpm | 500 | Immediately | P | S | – |

You can operate the panel operator instead of the host controller for Jog operation of the Servomotor.

Moreover, the signal of P-OT and N-OT are invalid during the Jog operation.

**----End**

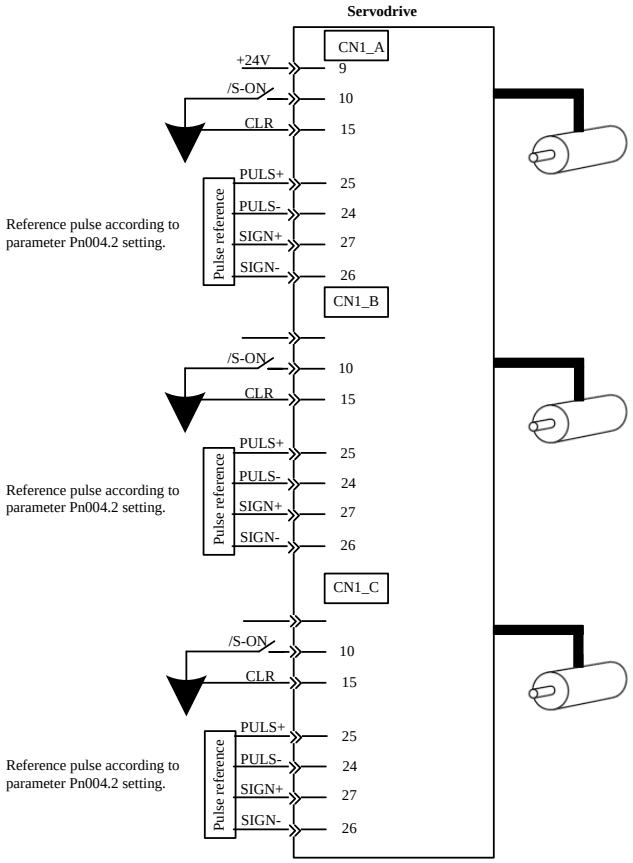
## 5.1.3 Trial Operation for Servomotor without Load from Host Controller

Check that the servomotor move reference or I/O signals are correctly set from the host controller to the servo drive.

Also check the wiring and polarity between the host controller and servo drive, and the servo drive operation settings are correct. This is the final check before connecting the servomotor to the machine.

### Operating Procedure in Position Control Mode (Pn005=H——1—)

The following circuits are required: External input signal circuit or equivalent.




**Step 1** Match the reference pulse form with the pulse output form from the host controller.
Set the reference pulse form with Pn004.2.

**Step 2** Set the reference unit and electronic gear ratio so that it coincides with the host controller setting.
Set the electronic gear ratio with Pn201 (or Pn203)/Pn202.

**Step 3** Turn the power and the servo ON input signal ON.

**Step 4** Send the slow speed pulse reference for the number of servomotor rotation easy to check (for example, one servomotor revolution) from the host controller in advance.
Set the servomotor speed to100rpm for the reference pulse speedbecause such speed is safe.

**Step 5** Check the number of reference pulses input to the servo drive by the changed amount before and after the Un013 and Un014 (input reference pulsecounter)[pulse] were executed.
See the section 4.4 Operation in Monitor for the details about how it is displayed.

**Step 6** Check whether the actual number of servomotor rotations Un009, Un010 coincides with the number of input reference pulses.
See the section 4.4 Operation in Monitor for the details about how it is displayed.

**Step 7** Check that the servomotor rotation direction is the same as the reference.
Check the input pulse polarity and input reference pulse form.

**Step 8** Input the pulse reference with the large number of servomotor rotation from the host controller to obtain the constant speed.
Set the servomotor speed to 100rpm for the reference pulse speed because such speed is safe.

**Step 9** Check the reference pulse speed input to the servo drive using the Un008in Monitor Mode.(input reference pulse speed)[rpm].
See the section 4.4 Operation in Monitor for the details about how it is displayed.

**Step 10** Check the servomotor speed using the Un000 in Monitor Mode. (servomotor speed) [rpm].
See the section 4.4 Operation in Monitor for the details about how it is displayed.

**Step 11** Check the rotation of the Servomotor.
To change the servomotor rotation direction without changing the input reference pulse form, see the

section 5.3.2 Rotation Direction.
Perform the operation from step 8 again after the servomotor rotation direction is changed.

**Step 12** When the pulse reference input is stopped and servo OFF status is entered, the trial operation for servomotor without load in position control mode is complete.

**----End**

## 5.1.4 Trial Operation with the Servomotor Connected to the Machine


Follow the procedure below for trial operation precisely as given.
Malfunctions that occur after the servomotor is connected to the machine not only damage the machine, but may also cause an accident resulting in death or injury.

Follow the procedure below to perform the trial operation.

**Step 1** Turn the power ON, and make the settings for the mechanical configuration related to protective functions such as overtravel and brake.
See the section 5.3 Basic Funtions Setting.
When a servomotor with brake is used, take advance measures to prevent vibration due to gravity acting on the machine or external forces before checking the brake operation. Check that both servomotor and brake operations are correct.

**Step 2** Set the necessary parameters for the control mode used.
Refer to the sections 5.5 Speed Control and 5.6 Position Control according to the control method.

**Step 3** Connect the servomotor to the machine with the coupling,etc.,while the power is OFF.

**Step 4** Check that the servo drive is servo OFF status and then turn ON the power to the machine (host controller). Check again that the protective function in step 1 operates normally.
See the section 5.3 Basic Funtions Setting.
For the following steps, take advanced measures for an emergency stop so that the servomotor can stop safely when an error occurs during operation.

**Step 5** Perform trial operation with the servomotor connected to the machine, following each section in 5.1.2 Trial Operation for Servomotor Without Load.
Check that the trial operation is completed according to the trial operation for servomotor without load.
Also, check the settings for machine such as reference unit.

**Step 6** Check the parameter settings for control mode used in step 2.
Check that the servomotor rotates matching the machine operating specifications.

**Step 7** Adjust the servo gain and improve the servomotor response characteristics, if necessary.
The servomotor will not be broken in completely during trial operation. Therefore, let the system run for a sufficient amount of time to ensure that it is properly broken in.

**----End**

## 5.1.5 Trial Operation for Servomotor with Brakes

Holding brake operation of the servomotor can be controlled with the brake interlock output (/BK) signal of the servo drive.

When checking the brake operation,take advance measures to prevent vibration due to gravity acting on the machine or external forces. Check the servomotor operation and holding brake operation with the servomotor separated from the machine.If both operations are correct, connect the servomotor to the machine and perform trial operation.

# 5.2 Control Method Setting

The control modes supported by the ED3M series Servodrives are described below.

| Parameter No. | Setting | Description |
|---|---|---|
| Pn005 | H – – 0 – | Speed control (parameter reference)<br>Controls servomotor speed using parameter reference. Use in the following instances.<br>• To control speed<br>• For position control using the encoder feedback divisionoutput from the servo drive to form a position loop in the host controller. |
| | H – – 1 – | Position Control (Pulse train reference)<br>Controls the position of the servomotor using pulse train position reference.<br>Controls the position with the number of input pulses, and controls the speed with the input pulse frequency.<br>Use this method when positioning is required. |
| | H – – 2 – | Speed control (contact reference) ↔ Speed control (Zero reference)<br>Use the three input signals /P-CON, /P-CL and /N-CL to control the speed as set in advance in the servo drive.<br>Three operating speeds can be set in the servo drive. (In this case, an analog reference is not necessary.) |
| | H – – 3 –<br>H – – 4 –<br>H – – 5 – | These are swiching modes for using the four control methods described above in combination. Select the control method switching mode that best suits the application. |

# 5.3 Basic Funtions Setting

## 5.3.1 Servo ON

This sets the servo ON signal (/S-ON) that determines whether the Servomotor power is ON or OFF.

**/S-ON Signal**

| Type | Name | Connector Pin | Setting | Meaning |
|---|---|---|---|---|
| Input | /S-ON | CN1□-10<br>CN1□-15<br>[Default] | ON (Low level) | Power ON the Servomotor. |
| | | | OFF (high level) | Power OFF the Servomotor. |

**IMPORTANT**

- Always input the servo ON signal before inputting the input reference to start or stop the servomotor.
  Do not input the input reference first and then use the /S-ON signal to start or stop. Doing so will degrade internal elements and may cause the servo drive to malfunction.
- A parameter can be used to re-allocate the input connector number for the /S-ON signal, refers to the section **3.2.3 Names and Functions**.

**/S-ON Selection**

A parameter can be always used to set the servo ON condition. This eliminates the need to wire /S-ON, but care must be taken because the servo drive can operate as soon as the power is turned ON.

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn000 | b – – – 0 | External S-ON signal enabled (Factory setting) |
| | b – – – 1 | External S-ON signal disabled, the servomotor excitation signal is opened automatically after outputting the S-RDY signal. |

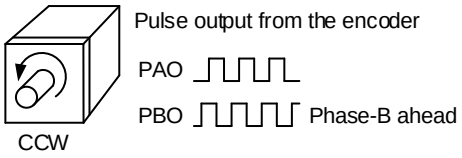
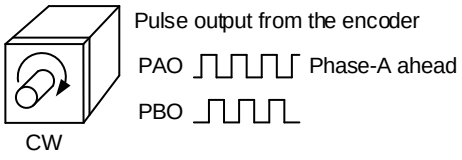
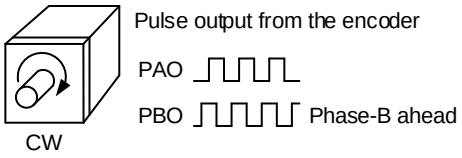
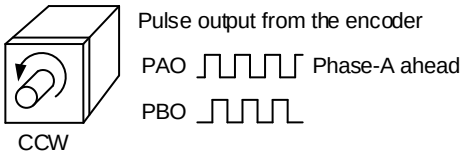
**NOTE**: After changing these parameters, turn OFF the main circuit and control power supplies, and then turn them ON again to enable the new settings.

## 5.3.2 Rotation Direction

The rotation direction of the servomotor can be switched without changing the reference pulse to the servo drive or the reference voltage polarity.

This causes the rotation the servo motor shaft is rotating to change. The output signal polarity, such as the encoder pulse output and the analog monitor signal from the servo drive do not change.

The standard setting for “forward rotation” is counterclockwise as viewed from the servomotor load end.

| Parameter No. | Setting | Name | Reference | Feedback |
|---|---|---|---|---|
| Pn001 | b – – – 0 | Use CCW as the forward direction. | Forward |  |
| | | | Reverse |  |
| | b – – – 1 | Use CW as the forward direction. | Forward |  |
| | | | Reverse |  |

NOTE: The direction of P-OT and N-OT change. For Pn001=b – – – 0 (standard setting), counterclockwise is P-OT. For Pn001=b – – – 1 (reverse rotation mode), clockwise is P-OT.

## 5.3.3 Overtravel

The overtravel limit function forces movable machine parts to stop if they exceed the allowable range of motion and turn ON a limit switch.

### Overtravel Connection

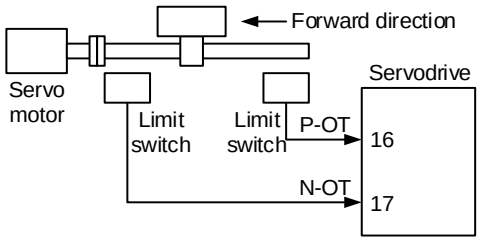
To use the overtravel function, connect the following overtravel limit switch to the corresponding pin number of servo drive CN1□ connector correctly.

| Type | Name | Pin No. | Setting | Meaning |
|---|---|---|---|---|
| Input | P-OT | Allocate them by Pn509 | ON (low level) | Forward rotation allowed. (Normal operation status.) |
| | | | OFF (high level) | Forward rotation prohibited. (Forward overtravel) |
| Input | N-OT | | ON (low level) | Reverse rotation (Normal operation status.) |
| | | | OFF (high level) | Reverse rotation prohibited. (Reverse overtravel) |

Connect limit switches as shown below to prevent damage to the devices during linear motion.

Rotation in the opposite direction is possible during overtravel.

For example, reverse rotation is possible during forward overtravel.




**IMPORTANT**

- When using overtravel to stop the servomotor during position control, the position error pulses are present. A clear signal(CLR)input is required to clear the error pulses.
- When using the servomotor on a vertical axis, the workpiece may fall in the overtravel condition.
- To prevent this, always set the zero clamp after stopping with Pn004.0=5.

### Overtravel Selection

A parameter can be set to disable the overtravel signal. If the parameter is set, there is no need to wire the overtravel input signal.

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn000 | b – – 0 – | Inputs the forward rotation prohibited (P-OT) signal from CN1□-12. |
| | b – – 1 – | Disables the forward rotation prohibited (P-OT) signal. |
| | b – 0 – – | Inputs the reverse rotation prohibited (P-OT) signal from CN1□-13. |
| | b – 1 – – | Disables the reverse rotation prohibited (P-OT) signal. |

NOTE: After changing these parameters, turn OFF the main circuit and control power supplies, and then turn them ON again to enable the new settings.

## Stop Method

This is used to set the stop method when an overtravel(P-OT,N-OT)signal is input while theservomotor is operating.

| Parameter No. | Setting | Stop Method | After Motor Stop | Description |
|---|---|---|---|---|
| Pn004 | H – – – 0 | Stop by dynamic brake | Coast | Rapidlly stops the servomotor by dynamic braking (DB), then places it into coast (power OFF) mode. |
| | H – – – 1 | Coast to a stop | | Stops the servomotor in the same way as when the servo is OFF (coast to a stop ), then places it into coast (power OFF) mode. |
| | H – – – 2 | S-OFF or Overtravel | Coast | Stops the servomotor by dynamic braking (DB) when servo OFF, stops the servomotor by plug braking when overtravel, and then places it into coast (power OFF) mode. |
| | H – – – 3 | | | Makes the servomotor coast to a stop state when servo OFF, stops the servomotor by plug braking when overtravel, and then places it into coast (power OFF) mode. |
| | H – – – 4 | | Zero Clamp | Stops the servomotor by dynamic braking (DB) when servo OFF, stops the servomotor by plug braking when overtravel, and then places it into zero clamp mode. |
| | H – – – 5 | | | Makes the servomotor coast to a stop state when servo OFF, stops the servomotor by plug braking when overtravel, then places it into zero clamp mode. |

NOTE: After changing these parameters, turn OFF the main circuit and control power supplies, and then turn them ON again to enable the new settings.

**NOTE**

- Stop by dynamic brake: Stops by using the dynamic brake (short circuiting its electrical circuit).
- Coast to a stop: Stops naturally, with no brake, by using the friction resistance of the servomotor in operation.
- Plug braking: Stops by using plug braking limit torque.
- Zero Clamp Mode: A mode forms a position loop by using theposition reference zero.

Dynamic brake is an emergency stop function, and one of the general methods to cause a servomotor sudden stop.

Dynamic brake suddenly stops a servomotor by shorting its electrical circuit.

If the servomotor is frequently started and stopped by turning the power ON/OFF or using the servo ON signal(/S-ON), the DB circuit will also be repeatedly operated, degrading the servo drive's internal elements.

Use the speed input reference and position reference to control the starting and the stopping of the servomotor.

## Stop Torque for Overtravel

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn405 | Plug braking torque limit` | 0 to 300 | % | 300 | Immediately | P | S | – |

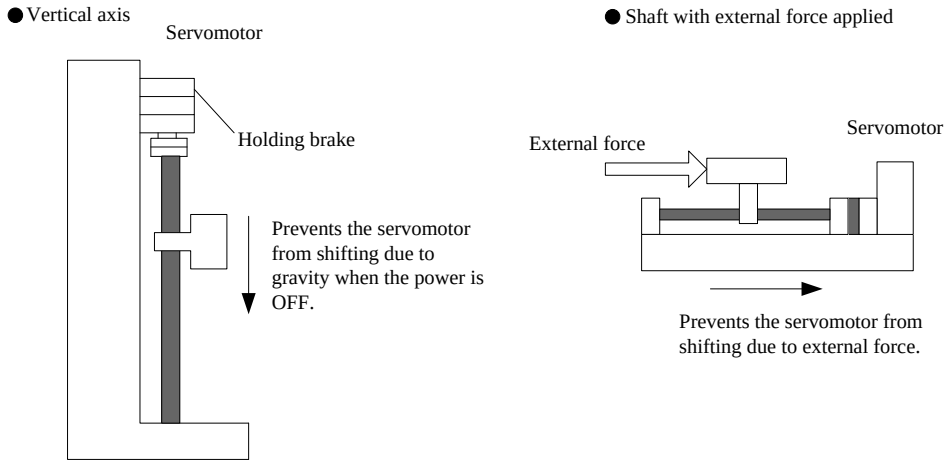
NOTE

- This sets the stop torque for when the overtravel signal(P-OT,N-OT) is input.
- The setting unit is a percentage of the rated torque.(the rated torque is 100%)
- The value large enough to be the servomotor maximum torque, 300% is set as the factory setting for plug braking limit torque.However, the actual output plug braking limit torque is determined by servomotor ratings.

## 5.3.4 Holding Brakes Setting

The holding brake is used when the servo drive controls a vertical axis.

A servomotor with the brake option helps prevent movable parts from shifting due to gravity when power is removed from the servo drive.(Refer to the section 5.1.5 Trial Operation for Servomotor with Brakes.)

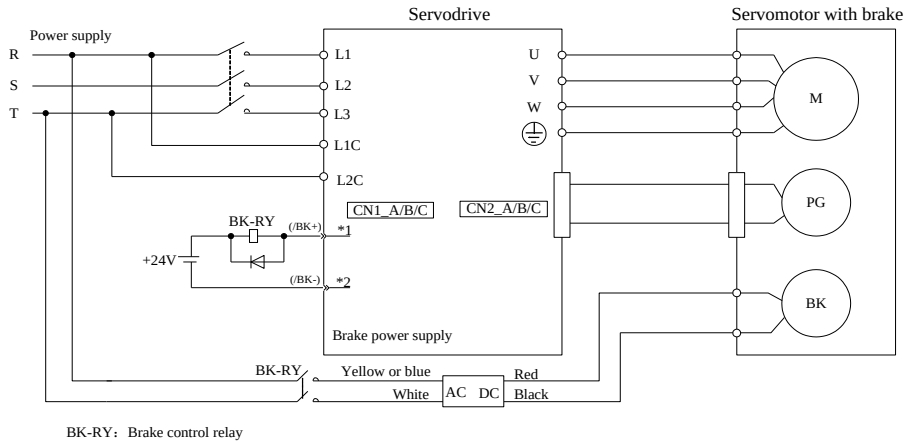



IMPORTANT

- The servomotor with the built in brake, is a de-energization brake. It is used to hold the servomotor and cannot be used as a braking purposes. Use the holding brake only to hold a stopped servomotor.
- When operating using only a speed loop, turn OFF the servo and set the input reference to 0V when the brake is applied.
- When forming a position loop, do not use a mechanical brake while the servomotor is stopped because the servomotor enters servolock status.

### Wiring Example

Use the servo drive sequence output signal /BK and the brake power supply to form a brake ON/OFF circuit. The following diagram shows a standard wiring example.




BK-RY：Brake control relay

1*、2*：The output terminals allocated with Pn511.

### Brake interlock output

| Type | Name | Pin No. | Setting | Meaning |
|---|---|---|---|---|
| Output | /BK | Must be allocated | ON (low level) | Releases the brake. |
| | | | OFF (high level) | Applies the brake. |

NOTE: This output signal controls the brake and is used only for a servomotor with a brake. This output signal is not used with the factory setting.The output signal must be allocated by Pn511. It does not need to be connected for servomotor without a brake.

### /BK Allocation

Brake interlock output (/BK) is not used with the factory setting.The output signal must be allocated.

| Parameter No. | Setting | Pin No. | | Meaning |
|---|---|---|---|---|
| | | + Terminal | − Terminal | |
| Pn511 | H – – – 4 | CN1□-11 | CN1□-12 | The /BK signal is output from output terminal CN1□-11, -12. |
| | H – – 4 – | CN1□-5 | CN1□-6 | The /BK signal is output from output terminal CN1□-5, -6. |
| | H – 4 – – | CN1□-9 | CN1□-10 | The /BK signal is output from output terminal CN1□-9, -10. |

**IMPORTANT**

- The setting of /BK signal is invalid in default setting.
- For the allocation of servo drive output signals other than /BK signal, refer to the section **3.2.3 Names and Functions**.

Parameter Pn511 description is as following:

| Pin No. | Description |
|---|---|
| 0 | /COIN (/V-CMP) output |
| 1 | /TGON rotation detecting output |
| 2 | /S-RDY servo drive get ready output |
| 3 | /CLT torque limit output |
| 4 | /BK brake interlock output |
| 5 | /PGC encoder C pulse output |
| 6 | OT overtravel signal output |
| 7 | /RD servo enabled motor excitation output |
| 8 | /HOME home completion output |
| 9 | /TCR torque detection output |

Relevant parameters are as following:

| Parameter No. | Name | Unit | Range | Default |
|---|---|---|---|---|
| Pn505 | Servo ON waiting time | ms | -2000 to 2000 | 0 |
| Pn506 | Basic waiting flow | 10ms | 0 to 500 | 0 |
| Pn507 | Brake waiting speed | rpm | 10 to 100 | 100 |
| Pn508 | Brake waiting time | 10ms | 10 to 100 | 50 |

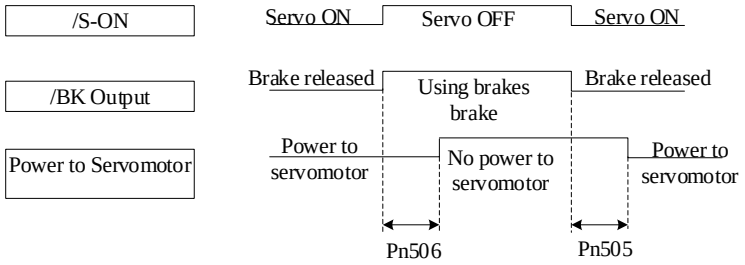
## Brake ON/OFF Setting for Servomotor Stop

The /BK signal is output at the same time as the servo is turned OFF in the default setting. The servo OFF timing can be changed with a parameter.

| Parameter No. | Name | Range | Unit | Default | When Enbaled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn505 | Servo ON waiting time | -2000 to 2000 | ms | 0 | Immediately | P | S | – |
| Pn506 | Basic waiting flow | 0 to 500 | 10ms | 0 | Immediately | P | S | |

When using the servomotor to control a vertical axis, the machine movable parts may shift slightly depending on the brake ON/ OFF timing due to gravity or an external force. By using this parameter to delay turning the servo ON/ OFF, this slight shift can be eliminated.

For details on brake operation while the servomotor is operating, refer to **Brake ON/OFF Setting for Servomotor Running**.




**IMPORTANT**

- The servomotor will turn OFF immediately when an alarm occurs, regardless of the setting of this parameter.
- The machine movable part may shift due to gravity or external force during the time until the brake operates.

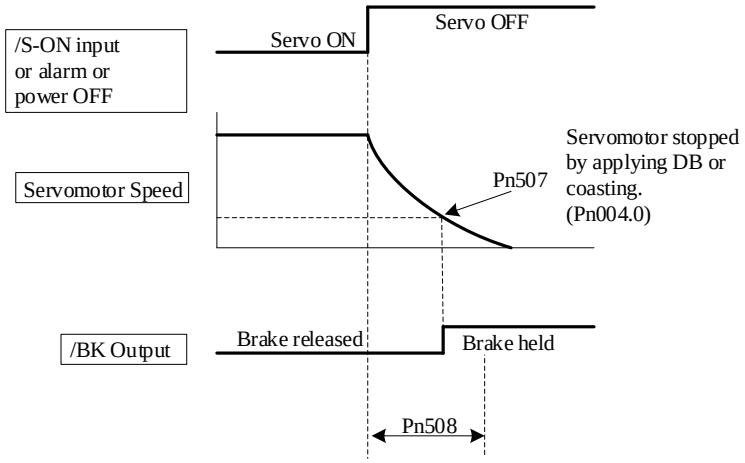
## Brake ON/OFF Setting for Servomotor Running

The following parameters can be used to change the /BK signal output conditions when a stop reference is output during servomotor operation due to the servo OFF or an alarm occuring.

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn507 | Brake Waiting Speed | 10 to 100 | 1rpm | 100 | Immediately | P | S | – |
| Pn508 | Brake Waiting Time | 10 to 100 | 10ms | 50 | Immediately | P | S | – |

The /BK signal goes to high level(brake ON) when either of the following conditions is satisfied:

- When the servomotor speed falls below the level set in Pn507 after servo OFF.
- When the time set in Pn508 is exceeded after servo OFF.




# 5.4 Absolute Encoders

| Type | Resolution | Data Range | Action for over-limit |
|---|---|---|---|
| ED3M model | 16-bit, multiturn<br>17-bit, singleturn | -32768 to +32767 | • When the upper limit (+32767)is exceeded in the forward direction, the multiturn data is -32768<br>• When the lower limit (-32768)is exceeded in the reverse direction, the multiturn data is +32767. |

The absolute position can be read by the MODBUS protocol. In the actual control, the MODBUS protocol can read the initial position when the servomotor is stopped (S-OFF), then the real-time position during the servomotor is running can be found from the number of PG divided output pulses.

## 5.4.1 Absolute Encoder Selection

An absolute encoder can also be used as an incremental encoder.

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn002 | b – 0 – – | Use the absolute encoder as an absolute encoder. (Factory setting) |
| | b – 1 – – | Use the absolute encoder as an incremental encoder. |

NOTE: The back-up battery is not required when using the absolute encoder as an incremental encoder.
After changing these parameters, turn OFF the main circuit and control power supplies and then turn them ON again to enable the new settings.

## 5.4.2 Handling Battery

In order for the absolute encoder to retain position data when the power is turned OFF, the data must be backed up by a battery.

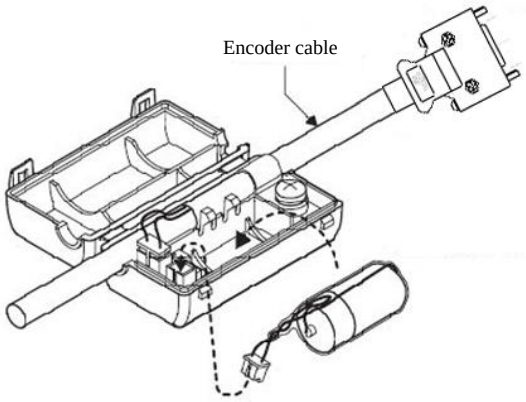
Please purchase the special cable and battery case mabe by Estun if an absolute encoder is used.

Install the battary to the encoder cable:

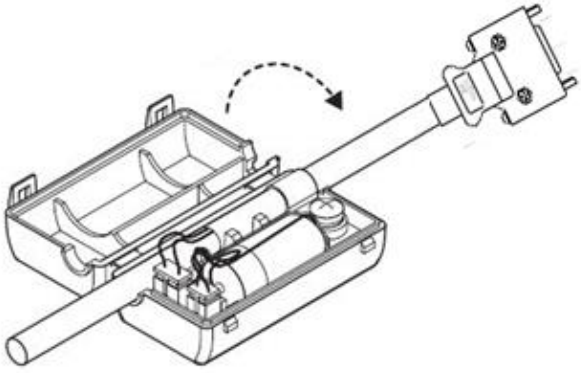
**Step 1** Open the shell of the battery case.

**Step 2** Install the battery according to the following diagram.




**Step 3** Cover the shell of the battery case.



**----End**

## 5.4.3 Replacing Battery

The servo drive will generate an absolute encoder battery alarm (A.48) when the battery voltage drops below about 3.1V.

Battery Replacement Procedure is as following.

**Step 1** Replace the battery with only the servo drive control power supply turned ON.

**Step 2** After replacing the battery, using the panel operator with utility function Fn011 to cancel the absolute encoder battery alarm (A.48).

**Step 3** Turn ON the servo drive power back again. If it operates without any problems, the battery replacement has been completed.

**IMPORTANT**

- The servo drive will generate an absolute encoder battery alarm (A.48) when the battery voltage drops below about 3.1V.
- If an absolute encoder battery alarm (A.47) occurred, it means the battery voltage drops below about 2.5V, and the multiturn data is lost. Please reset the absolute encoder after changing the battery.

**----End**

## 5.4.4 Absolute Encoder Setup (Fn010, Fn011)

Setting up the absolute encoder in the following cases.

- When starting the machine for the first time,set Pn002.2 to 0.
- When an encoder error alarm (A.45 to A.48, A.51) is generated.
  Use the panel operator in the servo drive for setup.

**CAUTION**

- Encoder setup operation is only possible when the servo is OFF.
- If the absolute encoder alarms(A.45 to A.48, A.51 ) are displayed, cancel the alarm by using the same method as the setup. They cannot be cancelled with the servo drive alarm reset input signal(/ALM-RST).
- Any other alarms that monitor the inside of the encoder should be cancelled by turning OFF the power.

# 5.5 Speed Control

## 5.5.1 Parameter Setting

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn005 | H – – 0 – | Control mode selection:Speed control (Internally set speed) [factory setting] |

## 5.5.2 Soft Start

The soft start function converts the stepwise speed reference inside the servo drive to a consistent rate of acceleration and deceleration.

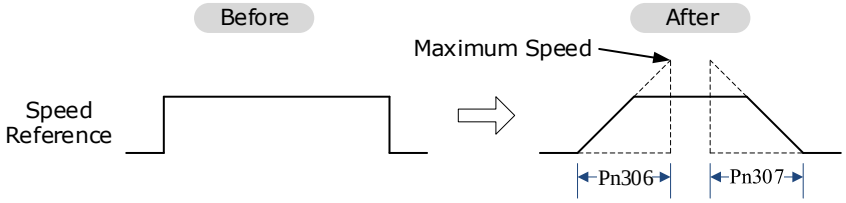
Pn310 can be used to select the soft start form:
: Slope; 1: S curve; 2: 1st-order filter; 3: 2nd-order filter

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn306 | Soft Start Acceleration Time | 0 to 10000 | 1ms | 0 | Immediately | – | S | – |
| Pn307 | Soft Start Deceleration Time | 0 to 10000 | 1ms | 0 | Immediately | – | S | |

The soft start function enables smooth speed control when inputting a stepwise speed reference or when selecting internally set speeds. Set both Pn306 and Pn307 to “0” for normal speed control.

Set these parameters as follows:

- Pn306: The time interval from the time the servomotor starts until the servomotor is 1000rpm.
- Pn307: The time interval from the time the servomotor is 1000rpm until it stops.



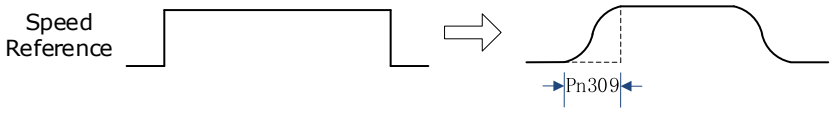
## 5.5.3 Speed Reference Filter Time Constant

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn308 | Speed Reference Filter Time Constant | 0 to 10000 | 1ms | 0 | Immediately | – | S | – |

NOTE: This smooths the speed reference by applying a 1st-order delay filter to the analog speed reference (V-REF) input. A value that is too large, however, will decrease response.

## 5.5.4 S-curve Risetime

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn309 | S-curve Rise time | 0 to 10000 | 1ms | 0 | Immediately | – | S | – |




## 5.5.5 Speed coincidence output

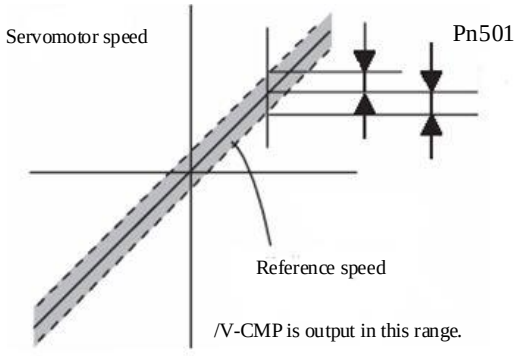
The speed coincidence (/V-CMP) output signal is output when the actual servomotor speed during speed control is the same as the speed reference input. The host controller uses the signal as an interlock.

| Type | Signal Name | Pin No. | Setting | Meaning |
|---|---|---|---|---|
| Output | /V-CMP (/COIN) | CN1□-7,8 [Default setting] | ON (low level) | Speed coincides. |
| | | | OFF (high level) | Speed does not coincide. |

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn501 | Coincidence Difference | 0 to 100 | rpm | 10 | Immediately | – | S | – |

The /V-CMP signal is output when the difference between the speed reference and actual servomotor speed is less than Pn501.

For example, the /V-CMP signal turns ON at 1900 to 2100rpm ifthe Pn501 parameter is set to 100 and the reference speed is 2000rpm

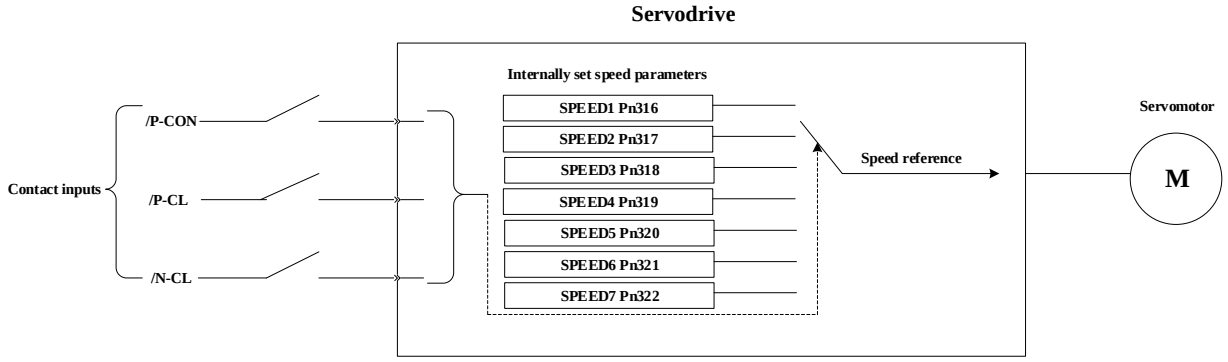


NOTE

This pin outputs the /COIN signal in position control mode, and the /V-CMP signal in speed control mode.

## 5.5.6 Speed control (contact reference)

The function of internally set speed selection allows speed control operation by externally selecting an input signal from among seven servomotor speed setting made in advance with parameters in the servo drive. The speed control operations within the three settings are valid. There is no need for an external speed or pulse generator.




### Parameter setting

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn005 | H – – 2 – | Speed control (contact reference) |

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn316 | Internal set speed 1 | -6000 to 6000 | rpm | 100 | Immediately | – | S | – |
| Pn317 | Internal set speed 2 | -6000 to 6000 | rpm | 200 | Immediately | – | S | – |
| Pn318 | Internal set speed 3 | -6000 to 6000 | rpm | 300 | Immediately | – | S | – |
| Pn319 | Internal set speed 4 | -6000 to 6000 | rpm | -100 | Immediately | – | S | – |
| Pn320 | Internal set speed 5 | -6000 to 6000 | rpm | -200 | Immediately | – | S | – |
| Pn321 | Internal set speed 6 | -6000 to 6000 | rpm | -300 | Immediately | – | S | – |
| Pn322 | Internal set speed 7 | -6000 to 6000 | rpm | 500 | Immediately | – | S | – |

NOTE: The servomotor's maximum speed will be used whenever a speed setting for the Pn316 to Pn322 exceeds the maximum speed.

### Control Method Switching

Use ON/OFF combinations of the following input signals to operate with the internally set speeds.

When Pn005.1=2: Selects the internally set speed (contact reference) ↔ Speed control (zero reference)

| /P-CON | /PCL | /NCL | Speed |
|---|---|---|---|
| OFF(H) | OFF (H) | OFF (H) | Speed control (zero reference) |
| | OFF (H) | ON (L) | SPEED1 |
| | ON (L) | OFF (H) | SPEED2 |
| | ON (L) | ON (L) | SPEED3 |
| ON(L) | OFF (H) | OFF (H) | SPEED4 |
| | OFF (H) | ON (L) | SPEED5 |
| | ON (L) | OFF (H) | SPEED6 |
| | ON (L) | ON (L) | SPEED7 |

NOTE: OFF= High level; ON= Low level

When Pn005.1 = 3, /P-CON, /PCL, /NCL = OFF (H), switches to position control (pulse train reference)

| /P-CON | /PCL | /NCL | Speed |
|---|---|---|---|
| OFF(H) | OFF(H) | OFF(H) | Position control (pulse train reference) |
| | OFF(H) | ON(L) | SPEED1 |
| | ON(L) | OFF(H) | SPEED2 |
| | ON(L) | ON(L) | SPEED3 |
| ON(L) | OFF(H) | OFF(H) | SPEED4 |
| | OFF(H) | ON(L) | SPEED5 |
| | ON(L) | OFF(H) | SPEED6 |
| | ON(L) | ON(L) | SPEED7 |

# 5.6 Position Control

## 5.6.1 Parameter Setting

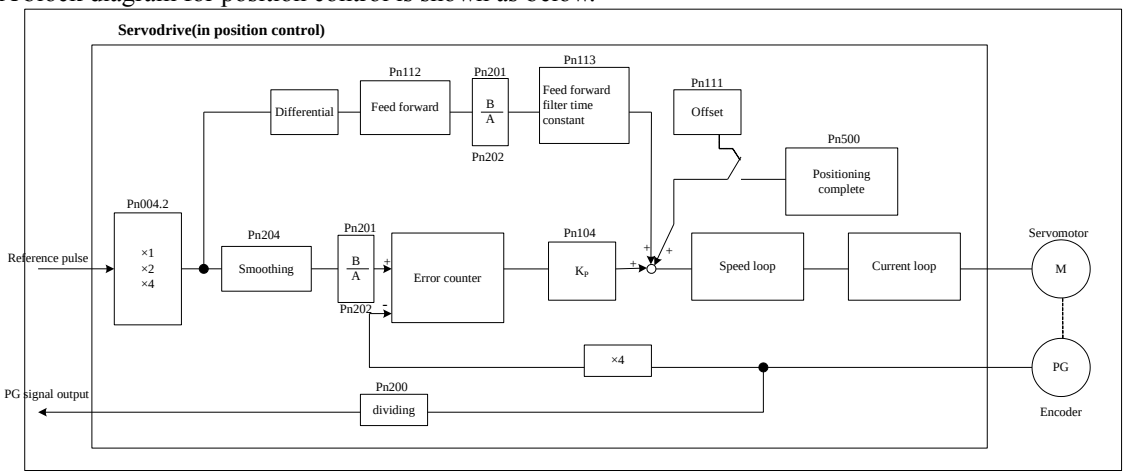
### Control Method Selection

Set the following parameters for position control using pulse trains.

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn005 | H – – 1 – | Position control (pulse train reference) |

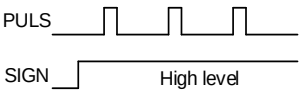
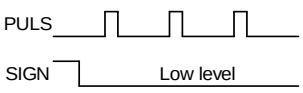
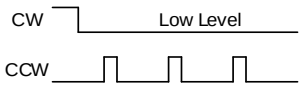
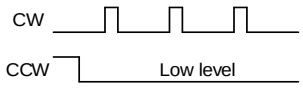
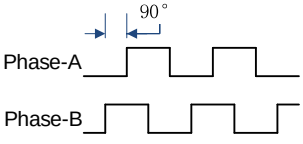
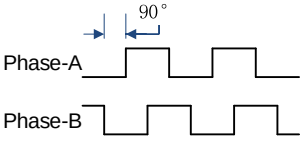
A block diagram for position control is shown as below.




## Setting a reference pulse sign

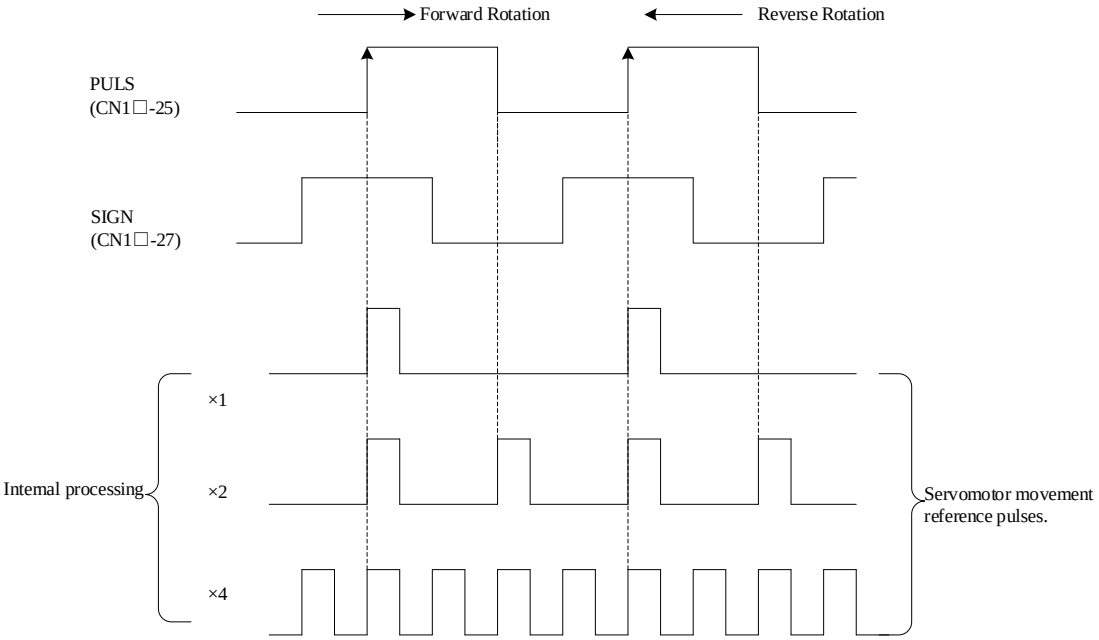
| Type | Axis | Name | Pin No. | Meaning |
|---|---|---|---|---|
| Input | A, B, C | PULS+ | CN1□-16 | Reference pulse input |
| | | PULS- | CN1□-15 | Reference pulse input |
| | | SIGN+ | CN1□-18 | Reference sign input |
| | | SIGN- | CN1□-17 | Reference sign input |

Set the input form for the servo drive using parameter Pn004.2 according to the host controller specifications.

| Parameter No. | Setting | Form | Multiplier | Forward Rotation | Reverse Rotation |
|---|---|---|---|---|---|
| Pn004 | H – 0 – – | Sign+pulse train (positive logic) (factory setting) | – |   |   |
| | H – 1 – – | CW+CCW (positive logic) | – |   |   |
| | H – 2 – – | Two-phase pulse train with 90° phase differential (positive logic) | ×1 |   |   |
| | H – 3 – – | | ×2 | | |
| | H – 4 – – | | ×4 | | |

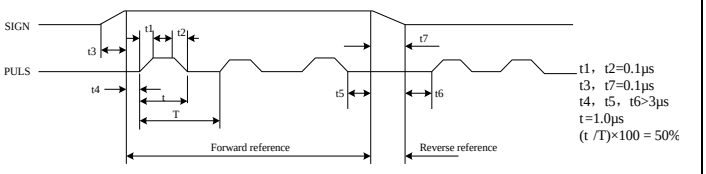
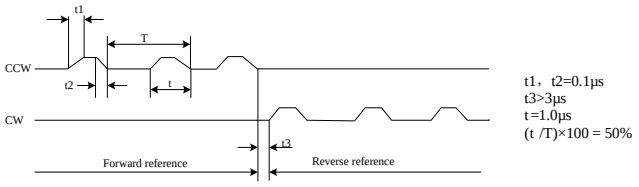
The input pulse multiplier can be set for the two-phase pulse train with 90° phase differential reference pulse form.

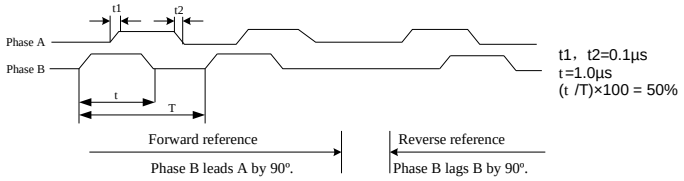



## Inverse PULS and SIGN reference

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn004 | H 0 – – – | Do not inverse PULS reference and SIGN reference. |
| | H 1 – – – | Do not inverse PULS reference; Inverse SIGN reference |
| | H 2 – – – | Inverse PULS reference; Do not inverse SIGN reference |
| | H 3 – – – | Inverse PULS reference and SIGN reference |

## Reference Pulse Input Signal Timing

| Reference pulse signal form | Electrical specifications | Remarks |
|---|---|---|
| Sign+pulse train input (SIGN+PULS signal)<br>Maximum reference frequency: 500kpps(For open-collector output: 200kpps) |  | SIGN<br>H=forward reference<br>L=reverse reference |
| CW pulse+CCW pulse<br>Maximum reference frequency:500kpps<br>(For open-collector output: 200kpps) |  | – |

| Reference pulse signal form | Electrical specifications | Remarks |
|---|---|---|
| Two-phase pulse train with 90° phase differential(phase A +B)<br>Maximum reference frequency:<br>×1 input pulse multiplier: 500kpps<br>×2 input pulse multiplier: 400kpps<br>×4 input pulse multiplier: 200kpps |  | A parameter Pn004.2 can be used to switch of the input pulse multiplier mode. |

## 5.6.2 Clear Setting

### Setting the Clear Signal

| Type | Sign Name | Pin No. | Meaning |
|---|---|---|---|
| Input | /CLR | CN1□-25 | Deviation counter clear |

When the /CLR signal is set to low level, clear deviation counter:

- The deviation counter inside the Servodrive is set to“0”
- Position loop operation is disabled.

### Setting the Clear Signal Mode

In position control mode, pulses will be still presented in the servo drive when servo OFF, thus it should be cleared when servo drive is turned ON. Setting Pn004 to choose whether clearing the pulses automatically when servo OFF.

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn004 | H – – 0 – | Clear the deviation count when Seveo OFF, but do not clear the deviation count when overtravel occurs. |
| | H – – 1 – | Never clear the deviation count. |
| | H – – 2 – | Clear the deviation count when Seveo OFF or overtravel occurs (excep for zero clamp). |

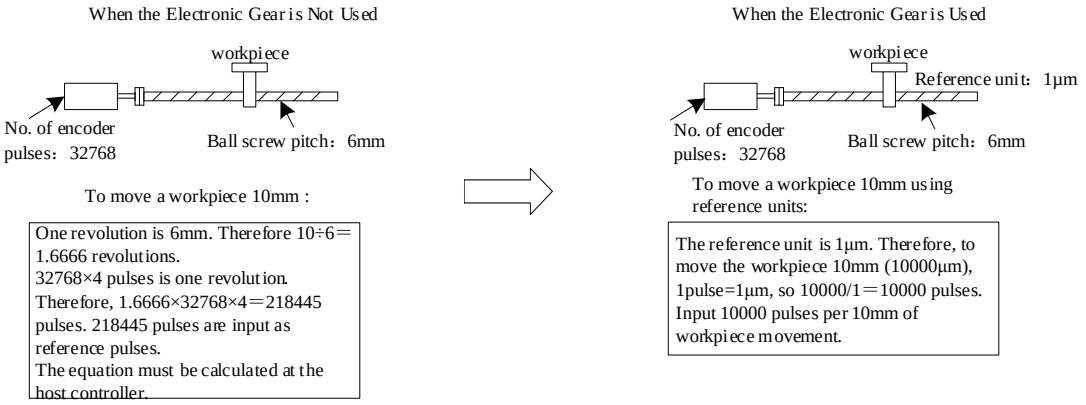
## 5.6.3 Electronic Gear Setting

### Electronic Gear

The electronic gear enables the workpiece travel distance per input reference pulse from the host controller to be set to any value.

One reference pulse from the host controller, i.e., the minimum position data unit, is called a reference unit.




## Relevant Parameters

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn009 | b – 0 – – | Use 16-bit electronic gear parameter |
| | b – 1 – – | Use 32-bit electronic gear parameter |

| Parameter No. | Name | Range | Unit | Default | When Enabled |
|---|---|---|---|---|---|
| Pn201 | 16 Bit Electronic Gear Ratio (Numerator) | 1 to 65535 | – | 1 | After restart |
| Pn317 | 16 Bit Electronic Gear Ratio (Denominator) | 1 to 65535 | – | 1 | After restart |
| Pn705 | 32 Bit Electronic Gear Ratio (Numerator, H) | 1 to 9999 | 10000 | 0 | After restart |
| Pn706 | 32 Bit Electronic Gear Ratio (Numerator, L) | 1 to 9999 | 1 | 1 | After restart |
| Pn707 | 32 Bit Electronic Gear Ratio (Denominator, H) | 1 to 9999 | 10000 | 0 | After restart |
| Pn708 | 32 Bit Electronic Gear Ratio (Denominator, L) | 1 to 9999 | 1 | 1 | After restart |

The deceleration ratio of the servomotor and the load shaft is given as n/m where m is the rotation of the servomotor and n is the rotation of the load shaft.

$$\text{Electronic gear ratio } \frac{B}{A} = \frac{Pn201}{Pn202} = \frac{\text{Encoder resolution} \times 4}{\text{Travel distance per load shaft revolution (reference units)}} \times \frac{m}{n}$$

- When 32 bit electronic gear function is enabled, $\frac{B}{A} = \frac{Pn705 \times 10000 + Pn706}{Pn707 \times 10000 + Pn708}$
- If the ratio is outside the setting range, reduce the fraction (both numerator and denominator) until you obtain integers within the range.
- Note that do not change the electronic gear ratio (B/A).

**IMPORTANT**

- Electronic gear ratio setting range: 0.01≤electronic gear ratio(B/A) ≤ 100
- If the electronic gear ratio is outside this range, the servo drive will not operate properly. In this case, modify the load configuration or reference unit.

## Procedure for Setting the Electronic Gear Ratio

Use the following procedure to set the electronic gear ratio.

**Step 1** Check machine specifications.
Check the deceleration ratio, ball screw pitch and pulley diameter.

**Step 2** Check the number of encoder pulses.
Check the number of encoder pulses for the servomotor used.

**Step 3** Determine the reference unit used.
Determine the reference unit from the host controller, considering the machine specifications and positioning accuracy.

**Step 4** Calculate the travel distance per load shaft revolution.
Calculate the number of reference units necessary to turn the load shaft one revolution based on the previously determined reference units.s

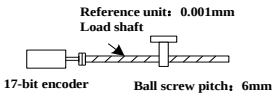
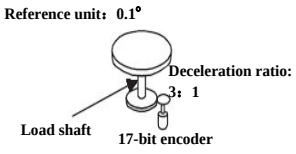
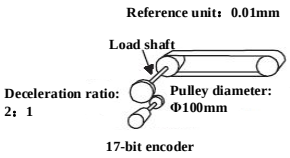
**Step 5** Calculate the electronic gear ratio.
Use the electronic gear ratio equation to calculate the ratio (B/A).

**Step 6** Set parameters.
Set parameters using the calculated values.

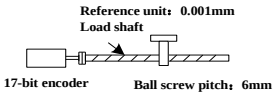
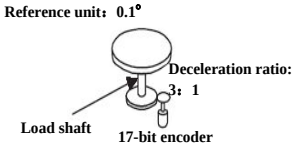
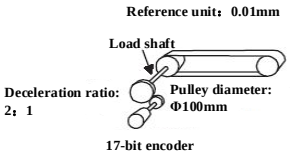
**----End**

## Electronic Gear Ratio Setting Examples

The following examples show electronic gear ratio settings for different load configurations.

| Step | Operation | Load Configuration | | |
|---|---|---|---|---|
| | | **Ball Screw**   | **Disc Table**   | **Belt and Pulley**   |
| 1 | Check machine specifications. | • Ball screw pitch:6mm<br>• Deceleration ratio:1/1 | • Rotation angle per revolution:360°<br>• Deceleration ratio:3/1 | • Pulley diameter:100 mm (pulley circumference:314 mm)<br>• Deceleration ratio:2/1 |
| 2 | Encoder | 17-bit:32768P/R | 17-bit:32768P/R | 17-bit:32768P/R |
| 3 | Determine the reference unit used | 1 reference unit: 0.001mm(1μm) | 1 reference unit:0.1° | 1 reference unit:0.01mm |
| 4 | Calculate the travel distance per load shaft revolution | 6mm/0.001mm=6000 | 360°/0.1°=3600 | 314mm/0.01mm=31400 |
| 5 | Calculate the electronic gear ratio | $\frac{B}{A}=\frac{32768\times4}{6000}\times\frac{1}{1}$ | $\frac{B}{A}=\frac{32768\times4}{3600}\times\frac{3}{1}$ | $\frac{B}{A}=\frac{32768\times4}{31400}\times\frac{2}{1}$ |

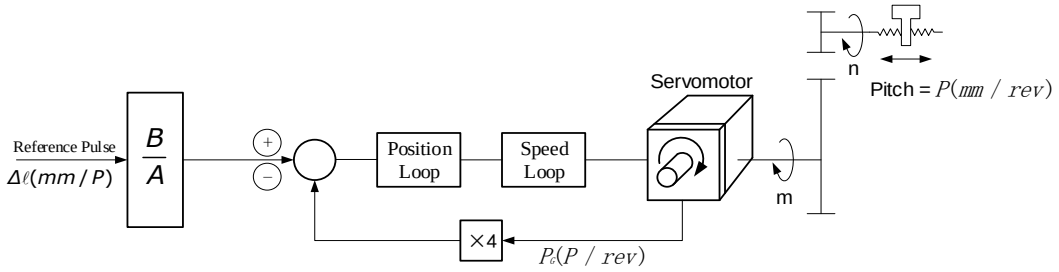
  

| Step | Operation | Load Configuration | | |
|---|---|---|---|---|
| | | **Ball Screw**<br>  <br> | **Disc Table**<br>  <br> | **Belt and Pulley**<br>  <br> |
| 6 | Set parameters | Pn201 = 131072<br>Pn202 = 6000 | Pn201 = 393216<br>Pn202 = 3600 | Pn201 = 262144<br>Pn202 = 31400 |
| 7 | Final result | Pn201 = 32768<br>Pn202 = 1500 | Pn201 = 32768<br>Pn202 = 300 | Pn201 = 32768<br>Pn202 = 3925 |

Reduce the fraction (both numerator and denominator) if the calculated result will not be within the setting range.

For example, reduce the above numerators and denominators by four or other numbers to obtain the final results in step 7 and complete the settings.

**Electronic Gear Ratio Equation**




In which,$\Delta\ell$ is reference unit; $P_G$ is encoder pulse; $P$ is ball screw pitch; $\frac{m}{n}$ is deceleration ratio.

$$\frac{n \times P}{\Delta\ell} \times \frac{B}{A} = 4 \times P_G \times m \Longrightarrow \frac{B}{A} = \frac{4 \times P_G \times m \times \Delta\ell}{n \times P} = \frac{4 \times P_G}{\frac{P}{\Delta\ell}} \times \frac{m}{n}$$

Set A and B by Pn202 and Pn201.

## 5.6.4 Smoothing

A filter can be applied in the servo drive to a constant-frequency reference pulse.

**Selecting a Position Reference Filter**

| Parameter No. | Meaning |
|---|---|
| Pn205 | 0: 1st-order filter |
| | 1: 2nd-order filter |

NOTE: After changing the parameter, turn OFF the power once and turn it ON again to enable the new setting.

### Relevant Parameters

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn204 | Position Reference Acceleration/Deceleration Time Constant | 0 to 32767 | 0.25ms | 0 | Immediately | P | – | – |

**IMPORTANT**

When the position reference acceleration/deceleration time constant (Pn204) is changed, a value with no reference pulse input and a position error of 0 will be enabled. To ensure that the setting value is correctly reflected, stop the reference pulse from the host controller and input the clear signal (CLR), or turn OFF to clear the error.

This function provides smooth servomotor operation in the following cases.

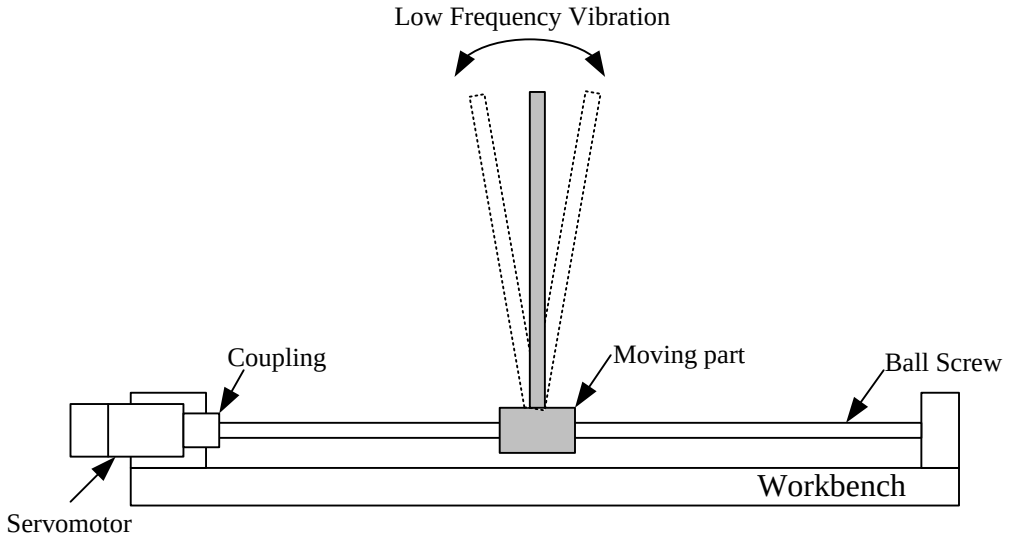
- When the host controller that outputs a reference that cannot perform acceleration/deceleration processing.
- When the reference pulse frequency is too low.
- When the reference electronic gear ratio is too high (i.e., 10× or more)

## 5.6.5 Low Frequency Vibration Suppression

### Function Description

For the low rigidity load, low frequency vibration will occur continually at the front end of the load during fast acceleration or fast deceleration. The vibration may delay positioning time and affect the productive efficiency.

The function of low frequency vibration suppression is embedded in the Servodrives by calculating the load position and compensating.




### Applicable Range

Low frequency vibration suppression function is enabled in both speed control mode and position control mode.

Low frequency vibration suppression function is disabled or can not reach the expected effect in the following conditions.

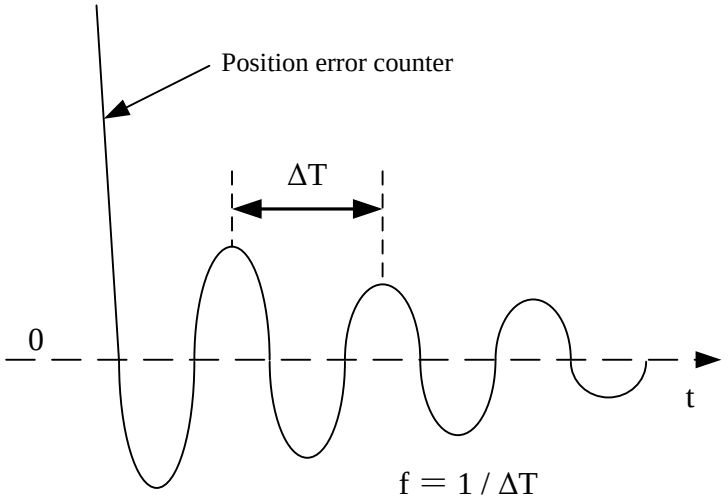
- Vibration is pricked up due to an external force.

- Vibration frequency is between5.0Hz to 50.0Hz.
- There is mechanical clearance at the mechanical connection part.
- The time for movement is less than one vibration period.

## Measuring Vibration frequency

Write the frequency data measured(unit:0.1Hz) directly to Parameter Pn411, if the vibration frequency can be measured by an instrument (such as a laser interferometer).And it also can be measured indirectly by communication software ESView or FFT analsis function.




## Ralated Parameters

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn006 | H – 0 – – | Low frequency vibration suppression function disabled |
| | H – 1 – – | Speed low frequency vibration suppression function enabled |
| | H – 2 – – | Position low frequency vibration suppression function enabled |

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn411 | Low frequency vibration frequency | 50 to 500 | 0.1Hz | 100 | Immediately | P | S | – |
| Pn412 | Low frequency vibration damp | 0 to 200 | – | 25 | Immediately | P | S | – |

Writing the frequency data to parameter Pn411 can adjust Pn411 slightly to obtain the best suppression effect.

If the servomotor stopped with continuous vibration, Pn412(Do not change in general) should be increased properly.

Parameter Pn411 and Pn412 are enabled when Pn006.2=1 or 2(Setting validation: after restart).

## 5.6.6 Positioning Completion Output Signal

This signal indicates that servomotor movement has been completed during position control. Use the signal as an interlock to confirm that positioning has been completedat the host controller.

| Type | Signal | Pin No. | Setting | Measning |
|---|---|---|---|---|
| Output | /COIN | CN1□-7, 8 [Factory Setting] | ON (low level) | Positioning has been completed. |
| | | | OFF (high level) | Positioning hasnot been completed. |

NOTE: This output signal can be allocated to an output terminal with parameter Pn511. Refer to the section **3.2.3 Names and Functions**.

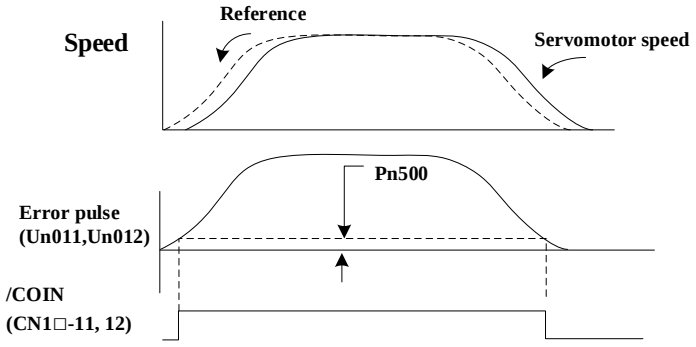
| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn500 | Positioning Error | 0 to 5000 | 1 Pulse | 10 | Immediately | P | – | – |
| Pn520 | Position complete time | 0 to 60000 | 0.25ms | 200 | Immediately | P | – | – |

The positioning completion (/COIN) signal is output when the difference (position error pulse) between the number of reference pulses output by the host controller and the travel distance of the servomotor is less than the value set in this parameter and the stabilization time is more than the value of Pn520.

Set the number of error pulses in reference unit (the number of input pulses defined using the electronic gear).

Too large a value at this parameter may output only a small error during low-speed operation that will cause the /COIN signal to be output continuously.

The positioning error setting has no effect on final positioning accuracy.




NOTE

- /COIN is a position control signal.
- This signal is used for the speed coincidence output /V-CMP for speed control, and it always OFF(high level) for torque control.

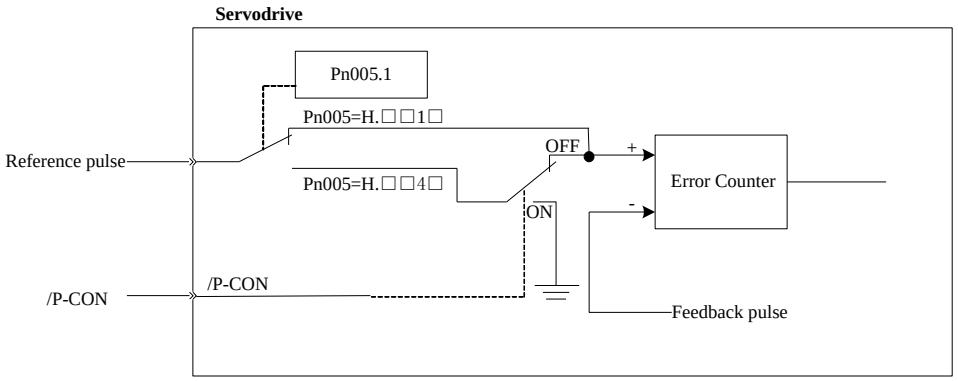
## 5.6.7 Reference Pulse Inhibit Function(INHIBIT)

### Function Description

This function inhibits the servo drive from counting input pulses during position control.

The servomotor remains locked (clamped) while pulses are inhibited.

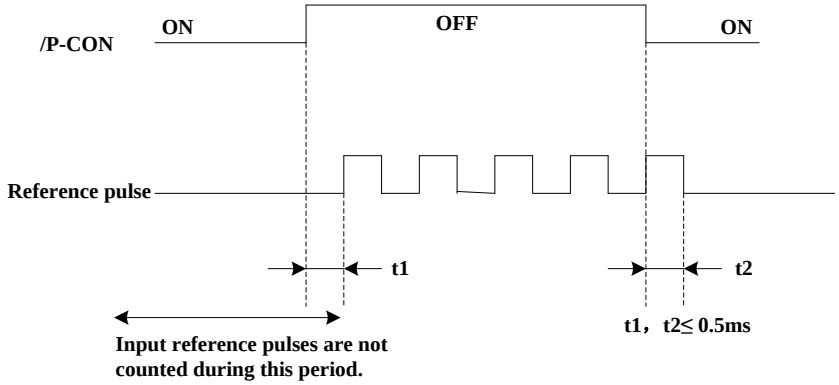
  




**Parameter Setting**

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn005 | H – – 4 – | Position control (pulse train reference) ↔ INHIBIT |

Inhibit(INHIBIT) switching condition: /P-CON signal ON (low level)




**Signal Setting**

| Type | Signal Name | Pin Number | Setting | Measning |
|---|---|---|---|---|
| Input | /P-CON | CN1□-11 | ON (low level) | Turns the INHIBIT function ON. (Inhibit the servo drive from countingreference pulses) |
| | | | OFF (high level) | Turns the INHIBIT function OFF. (Counters reference pulses.) |

## 5.6.8 Position Control (contact reference)

Position control under contact reference (parameter Pn005.1 = 5). In this mode, servo drive can position with a single axes without a host controller.

There are 16 position control points with each being able to set move distance, running speed, constants for position reference filter time, and the stop time when positioning completed. Two speeds (1. speed moving toward distance switch “speed of looking for reference point”. 2. Speed moving away from distance switch “moving speed.”) of reference points could be set as:

- Two position modes: 1. Absolute position mode 2. Relative position mode
- Two running modes: 1. Circling mode 2. Non-circling mode
- Two step switching method: 1. Delay step switching 2. /P-CON signal switching

- Method of looking for reference points: 1. Forward direction 2. Reverse direction

## Offset Adjustment

Offset of each points has two correspondent parameters: one unit of the parameter is [x 10000 reference pulse] and the other is [x 1 reference pulse]. Setting range of both parameters is: (-9999----+9999), while offset value equals sum of those two values.

For example: No.0 offset correspond to parameter Pn600 [x 10000 reference pulse] and Pn601[x 1 reference pulse]. Set Pn600 = 100, Pn601=-100.

$$\begin{aligned}\text{No. 0 offset value} &= \text{Pn600} \times 10000 \text{ reference pulse} + \text{Pn601} \times 1 \text{ reference pulse} \\ &= 100\text{x}10000 \text{ reference pulse} + (-100)\text{ x}1 \text{ reference pulse} \\ &= 999900 \text{ reference pulse}\end{aligned}$$

With the same principle, we can conclude: in order to get the same results, we also can set

$$\text{Pn600} = 99, \text{Pn601} = 9900$$

Thus, we can see when the two parameters are not zero; we can get same result by two ways: one is to set the two parameters both negative or both positive, or one negative the other positive.

## Speed

Speed mentioned here refers to the steady speed during which the motor is running, which is similar to the pulse frequency given from the external pulse reference in position control.However, this speed has nothing to do with the electronic gear; it is the actual speed of the motor.

## Position reference filter time constant

Same as position reference filter time constant Pn204 in common position control.

## Time for change steps after desired position reached

Apply internal delay to change steps to a valid value in parameter Pn681.1.

Time for change steps outputs from positioning completed signal CON/, from Servo ON, or from the time when reference point is found till the Servo performs the program to control position of the point. Such period of time depends on step changing time required by a point number among start point in program.

When running point control program, if deviation counter is set as “not clear deviation counter when Servo OFF”, then the deviation counter might flood. If it does not flood, then the servo drive will probably run at the max. running speed when Servo ON again. **PLEASE PAY ATTENTION TO THE SAFETY OF INSTRUMENT.**

| Parameter No. | Name and description | Unit | Range | Default |
|---|---|---|---|---|
| Pn004.1 | [0] Clear error pulse when S-0FF, not clear error pulse when overtravel.<br>[1] Not clear error pulse<br>[2] Clear error pulse When S-OFF or over travel | – | 0 to 2 | 0 |

## Looking for the reference point

Looking for the reference point is for establishing a zero physical point of the operating platform, which is used as zero point in the coordinates during point position control. And users may choose to find a reference point either in forward or reverse side.

- How to find a reference point
  Mount a limit switch in the forward or reverse side.Find a reference point in the forward direction after connecting to /PCL and in the reverse direction after connecting to /NCL. When the operating platform bumps into the limit the switch, the motor will first stop according to the way set by Pn004.0,

and then rotate again against limit the switch. When the operating platform leaves the limit switch and the motor reaches the position of first photo encoder Phase C pulse,then position of operating platform is set to be the zero point of the coordinates.

- How to find Relevant parameters of reference point
  Speed towards limit switch is called “speed of looking for reference point “, and the moving speed away from limit switch is called “ moving speed”. These two speeds could be set by the following parameters:

| Parameter No. | Description | Unit | Range | Default |
|---|---|---|---|---|
| Pn685 | Speed of looking for reference point (hits the limit switch) | rpm | 0 to 3000 | 1500 |
| Pn686 | Moving speed (move away from limit switch) | rpm | 0 to 200 | 30 |

Usually, the set speed of the reference point (Pn685) is high, and the moving speed (Pn686) is low. Note: if moving speed is too high, precision of finding a reference point would be affected.

When looking for a reference point, /PCL and /NCL are no longer programmed to limit external current.

**Relevant parameter**

| Parameter No. | Function | Description |
|---|---|---|
| Pn681.0 | Choose between cycle run and single run.<br>[0] Cycle run, /PCL as start signal, /NCL reverse to look for reference point.<br>[1] Single run, /PCL as start signal, /NCL reverse to look for reference point.<br>[2] Cycle run, /NCL as start signal, /PCL reverse to look for reference point.<br>[3] Single run, /NCL as start signal, /PCL reverse to look for reference point. | Changing steps will be performed till the end point is completed comma and the next change will start from the start point during multi-points cycle run.<br>Point control program will not change steps after the end point is completed during multi- points single run. |
| Pn681.1 | Change step and start mode<br>[0] Delay changing steps, the start signal is not needed.<br>[1] Change steps by /P-CON, start signal not needed.<br>[2] Delay changing steps, need start signal.<br>[3] Change steps by /P-CON, need start signal. | Change steps by external /P-CON signals. The signal will be valid when drive output reaches the desired position. When input signal changes, the signal is valid, then steps will be changed by consequence from start point to end point. |
| Pn681.2 | Change step input signal mode<br>[0] High or low level<br>[1] Sign pulse | – |
| Pn682 | [0] Incremental<br>[1] Absolute | Incremental: relative moving distance (distance from current point to next point) programming.<br>Absolute: absolute moving distance (distance between operating platform and the reference point) programming. |

## 5.6.9 Internal Homing Function

The servomotor always needs to operate at a fixed position. This position is normally regarded as the zero position. When the host controller is turned on, the zero position adjustment is required before processing.

This zero position will be regarded as the reference point. ESTUN servo drives can perform this function by the homing function.

## Homing Mode Setting

| Parameter No. | Setting | Meaning |
|---|---|---|
| Pn689 | b – – – 0 | Homing in the forward direction |
| | b – – – 1 | Homing in the reverse direction |
| | b – – 0 – | Return to search C-Pulse when homing |
| | b – – 1 – | Directly search C-Pulse when homing |
| | b – 0 – – | Homing function disabled |
| | b – 1 – – | Homing triggered by SHOM signal(rising edge) |

NOTE:
Applicable control mode: position control(contact reference) ↔ speed control(contact reference)
Homing operation can only be operated when /COIN is ON.
Pulses sent from the host controller is disabled when homing
Homing operation is disabled when in switching control mode.
Control mode switching is not allowed during homing.
After changing these parameters, turn OFF the main circuit and control power supplies and then turn them ON again to enable the new settings.
A parameter can be used to re-allocate input connector number for the SHOM and ORG signals.

## Relevant Parameters

| Parameter No. | Name | Range | Unit | Default | When Enabled |
|---|---|---|---|---|---|
| Pn685 | Speed of finding reference point(Hitting the origin signal ORG) | 0 to 3000 | Rpm | 1500 | Immediately |
| Pn686 | Speed of finding reference point(Leaving the origin signal ORG) | 0 to 200 | rpm | 30 | Immediately |
| Pn690 | Number of error pulses during homing | 0 to 9999 | 10000 Pulses | 0 | Immediately |
| Pn691 | Number of error pulses during homing | 0 to 9999 | 1 Pulse | 0 | Immediately |

## Input Signal Setting

| Type | Signal Name | Pin No. | Setting | Meaning |
|---|---|---|---|---|
| Input | SHOM | Allocated by Pn509, Pn510 | ON (rising edge) | Homing is enabled |
| | | | OFF(not rising edge) | Homing is disabled |
| Input | ORG (ZPS) | Allocated by Pn509, Pn510 | ON (high level) | ORG is enabled |
| | | | OFF (low level) | ORG is disabled |
| Output | /HOME | Allocated by Pn511 | ON (level level) | Homing completed |
| | | | OFF (high level) | |

NOTE: After changing Pn509, Pn510 and Pn511 turn OFF the main circuit and control power supplies and then turn them ON again to enable the new settings.

## Homing Operation Description

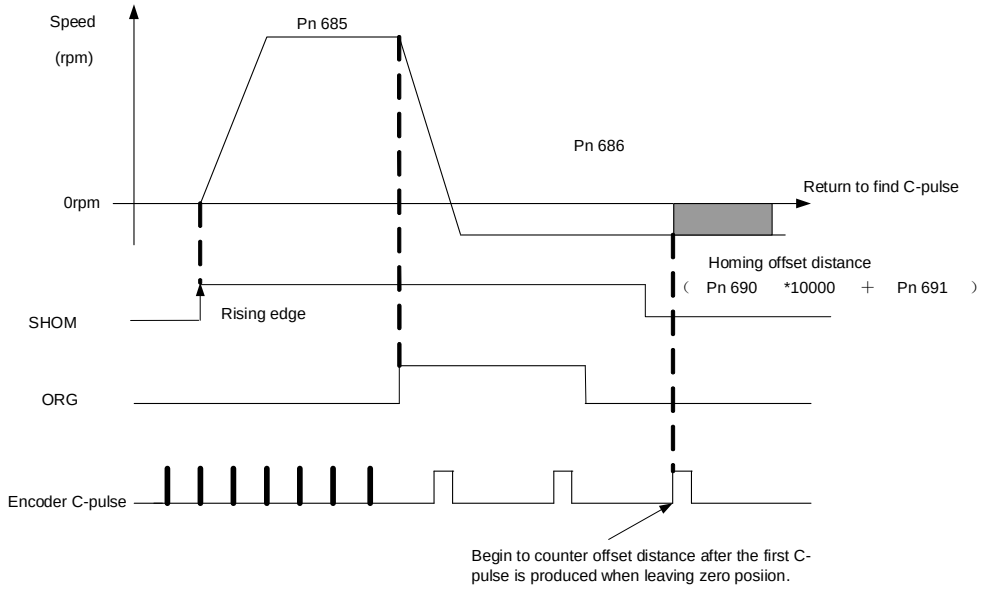
Please set Pn689 according to the actual operation in position control mode. When starting the homing function, the servomotor will run at the speed of Pn685 when detecting the rising edge of SHOM signal; the servomotor will run at the speed of Pn686 according to the setting of Pn689.1 when detecting the valid ORG signal.

When input ORG and the encoder C-Pulse is detected, the servo drive will begin to calculate the number of homing offset pulses. When offset pulses is completed, the servomotor stops and outputs homing completion signal /HOME, then homing control is completed.

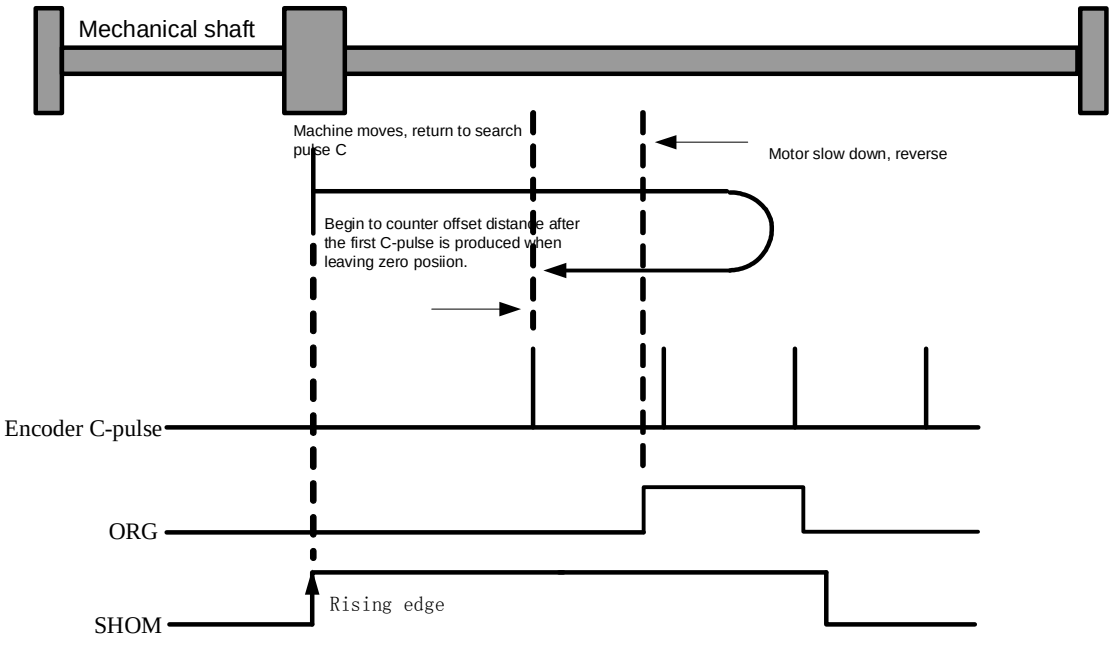
Pn685 (Hitting the origin signal (ORG)) is usually set at high speed, Pn686 (Leaving the origin signal ORG) is usually set at low speed.

Please be attention that if Pn686 is setting too high, the precision of mechanical zero position will be affected.

After hitting the origin signal ORG, the motor will return to find C-pulse; the figure is shown as below:

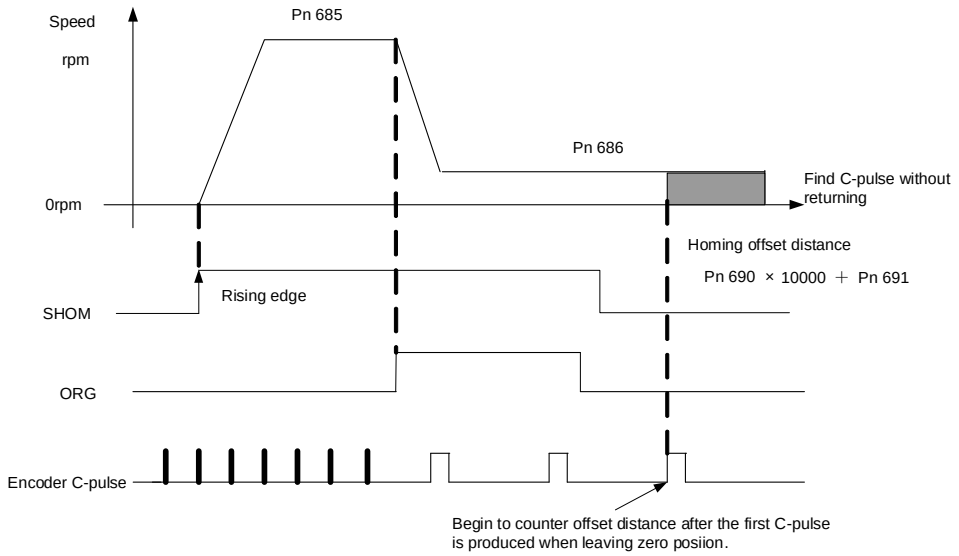



Corresponding position:

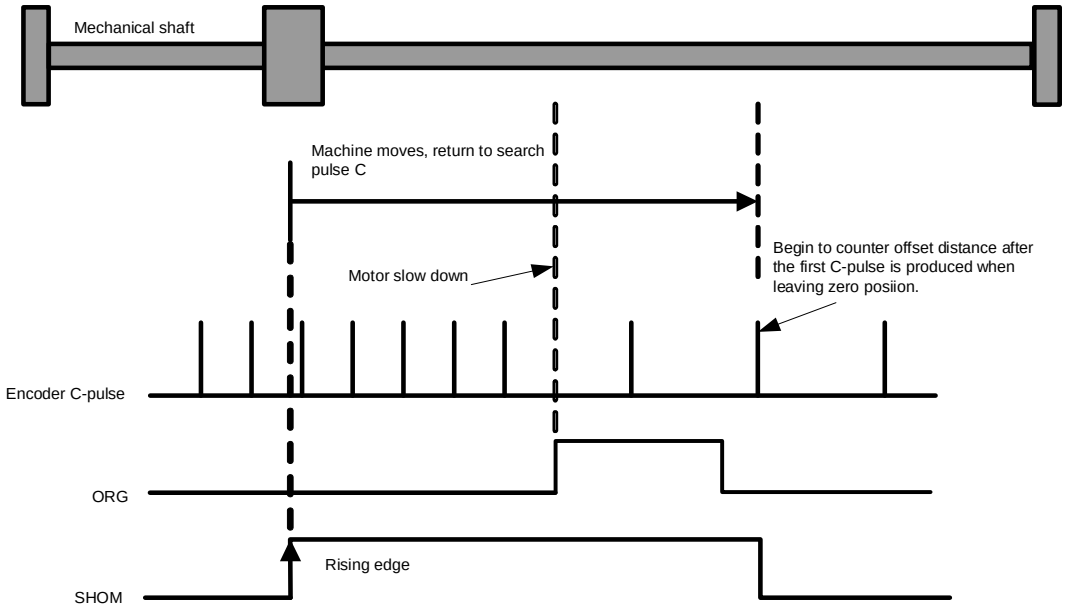


After hitting the origin signal ORG, the motor will find C-pulse directly; the figure is shown as below:




Corresponding position:




# 5.7 Torque Limit

The Servodrive provides internal torque limit/external torque limit for limiting output torque to protect the machine.

## 5.7.1 Internal Torque Limit

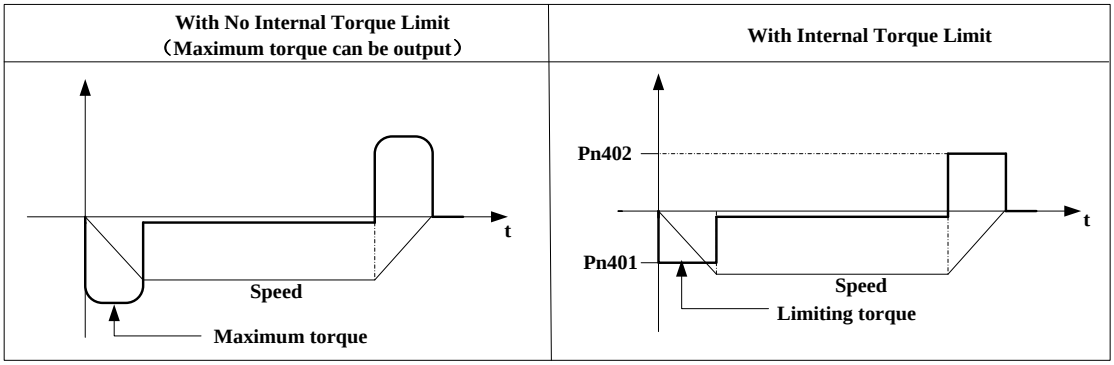
Maximum torque is always limited to the values set in the following parameters.

| Parameter No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn401 | Forward Torque Limit | 0 to 300 | % | 300 | Immediately | P | S | – |
| Pn402 | Reverse Torque Limit | 0 to 300 | % | 300 | Immediately | P | S | – |

The setting unit is a percentage of rated torque.

The maximum torque of the servomotor is used, even though the torque limit is set higher than the maximum torque of the servomotor. (as is the case with the 300% factory setting)




NOTE: Too small a torque limit setting will result in insufficient torque during acceleration and deceleration.

## 5.7.2 External Torque Limit

This function allows the torque to be limited at specific times during machine operation, for example, during press stops and hold operations for robot workpieces.

An input signal is used to enable the torque limits previously set in parameters.

### Relevant Parameters

| Parameter No. | Name | Range | Uint | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn403 | Forward External Torque Limit | 0 to 300 | % | 100 | Immediately | P | S | – |
| Pn404 | Reverse External Torque Limit | 0 to 300 | % | 100 | Immediately | P | S | – |

NOTE: The setting unit is a percentage of rated torque (i.e., the rated torque is 100%).

### Input Signals

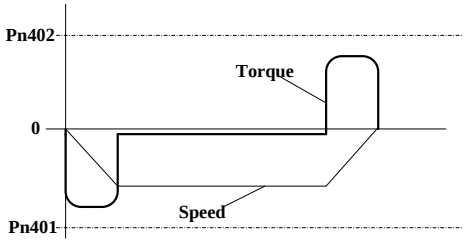
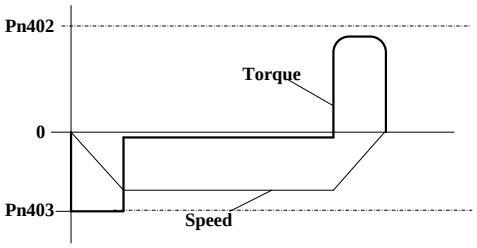
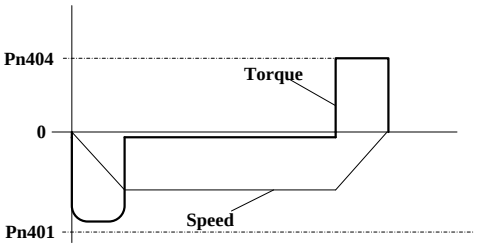
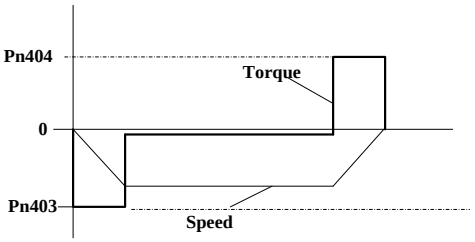
| Type | Signal Nmae | Allocation | Setting | Meaning | Limit Value |
|---|---|---|---|---|---|
| Input | /PCL | Pn509.0=6 | ON (low level) | Forward external torque limit | Pn403 |
| | | | OFF (high level) | Forward internal torque limit | Pn401 |
| Input | /NCL | Pn509.0=7 | ON (low level) | Reverse external torque limit | Pn404 |
| | | | OFF (high level) | Reverse internal torque limit | Pn402 |

NOTE: When using this function, make sure that there are no other signals allocated to the same terminals as /P-CL and /N-CL.

### Changes in Output Torque during External Torque Limiting

For example: External torque limit (Pn401, Pn402) set to 300%

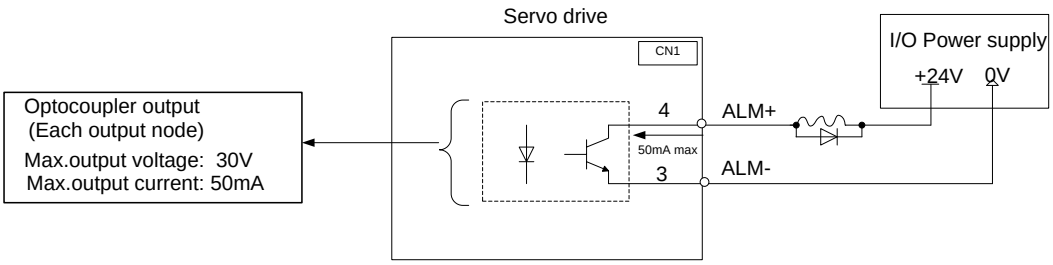
| /PCL | /NCL | |
|---|---|---|
| High level | High level |  |
| High level | Low level |  |
| Low level | High level |  |
| Low level | Low level |  |

NOTE: Select the servomotor rotation direction by setting Pn001=b－ － －0 (standard setting, CCW=Forward direction).

# 5.8 Other Output Signals

## 5.8.1 Servo alarm output

The following diagram shows the right way to connect the Alarm Output.

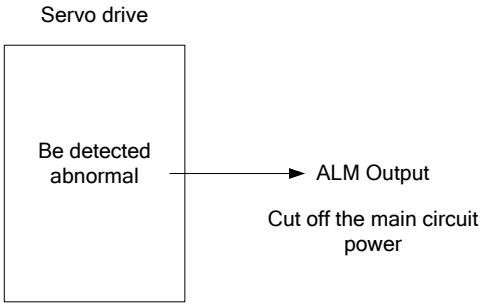



An external +24V I/O power supply is required since there is no +24V power source available inside the servo drive.

| | | |
|---|---|---|
| Output → ALM+ | CN1□-4 | Servo alarm output |

| | |
|---|---|
| Output → ALM- | CN1□-3 |

ALM outputs a signal when the servo drive is detected in an abnormal state.




Normally, the external circuit consists of /ALM should be able to switch off the power of servo drive.

| Signal | Status | Output Level | Description |
|---|---|---|---|
| ALM | ON | CN1□-4: Low level | Normal state |
| | OFF | CN1□-4: High level | Alarm state |

When “servo alarm(ALM)” happens, always remove alarm reasons first , and then turn the input signal "/ALM-RST" to ON position to reset alarm status.

## 5.8.2 Others

Set Pn511.9 to allocate the desired output signal.

| Pn511.0 | SignalName | Connector PinNumber | Setting | Meaning |
|---|---|---|---|---|
| 0 | /COIN(/VCMP) | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Positioning is complete. |
| | | | OFF=H | Positioning is not complete |
| 1 | /TGON | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Servomotor is operating(Servomotor speed is above the setting in Pn503). |
| | | | OFF=H | Servomotor is not operating(Servomotor speed is below the setting in Pn503). |
| 2 | /S-RDY | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Servo is ready. |
| | | | OFF=H | Servo is not ready. |
| 3 | /CLT | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Motor output torque under limit (Internal torque reference is higher than setting value). |
| | | | OFF=H | No torque limit (Internal torque reference is lower than setting value). |

| Pn511.0 | SignalName | Connector PinNumber | Setting | Meaning |
|---|---|---|---|---|
| 4 | /BK | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Releases the brake. |
| | | | OFF=H | Applies the brake. |
| 5 | PGC | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | With encoder C pluse output |
| | | | OFF=H | Without encoder C pluse output |
| 6 | OT | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Without forward rotation Prohibited(POT) and reverse rotation prohibited(NOT)signal |
| | | | OFF=H | With forward rotation Prohibited(POT)and reverse rotation prohibited(NOT)signal |
| 7 | /RD | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Servo enabled motor excitation |
| | | | OFF=H | Servo disabled motor not excitation |
| 8 | /HOME | CN1□-7/8<br>CN1□-1/2<br>CN1□-5/6 | ON=L | Homing is enabled |
| | | | OFF=H | Homing is disabled |
| 9 | /TCR | Not including this setting in the default setting,please choose terminal output by setting parameter Pn511 | ON=L | Motor output torque is higher than Pn529 setting value. |
| | | | OFF=H | Motor output torque is lower than Pn529 setting value. |

# 5.9 Online Autotuning

## 5.9.1 Function Description

Online autotuning calculates the load moment of inertia during operation of the servo drive and sets parametersso that the servo gains are consistent with the machine rigidity.

Online autotuning may not be effective in the following cases:

- The motor high speed is lower than 100 rpm.
- The motor acceleration or deceleration is lower than 5000rpm/s.
- Load rigidity is low and mechanical vibration occurs easily or friction is high.
- The speed load moment is changed greatly.
- Mechanical gas is very large.

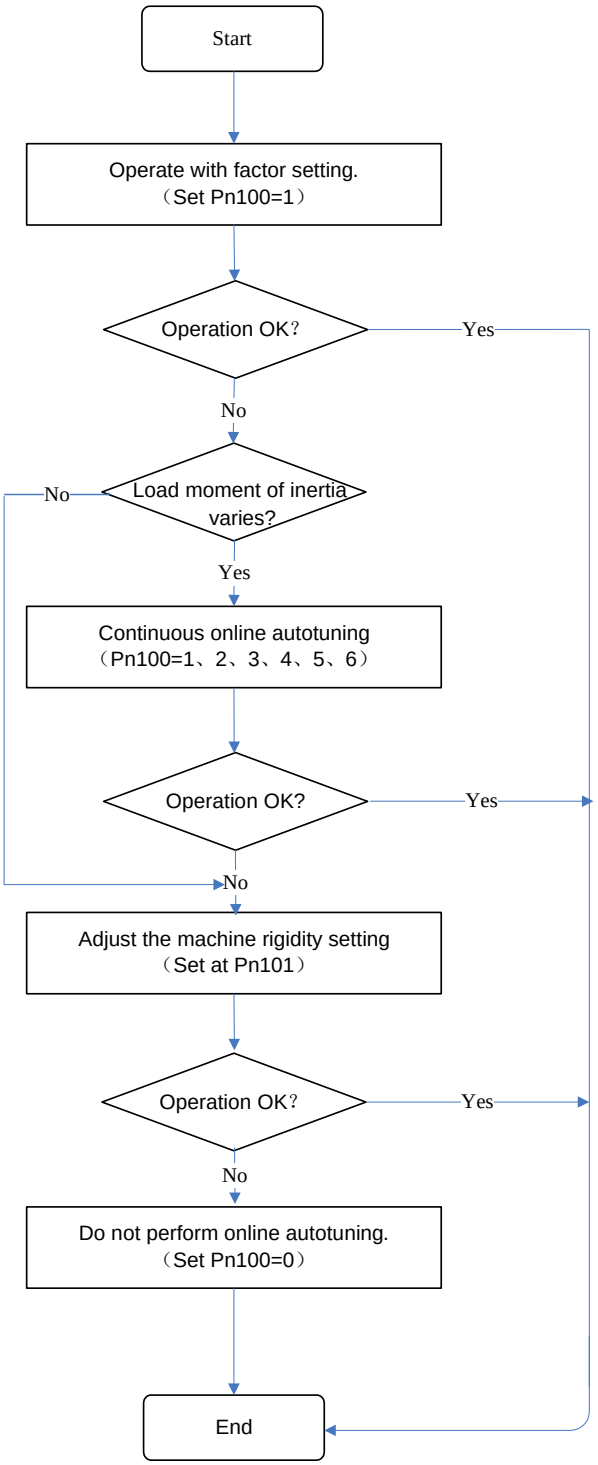
  

If the condition meets one of the above cases or the desired operation cannot be achieved by the online autotuning, set the value in Pn106 (Load inertia percentage) and performthe adjustment manually.

## 5.9.2 Online Autotuning Procedure

**CAUTION**

- Do not perform extreme adjustment or setting changes causing unstable servo operation.Failure to observe this warning may result in injury and damages to the machine.
- Adjust the gains slowly while confirming motor operation.



## 5.9.3 Setting Online Autotuning

| Parameter No. | Description | Unit | Range | Default | When Enabled |
|---|---|---|---|---|---|
| Pn100 | Online autotuning setting | – | 0000 to0006 | 0000 | After restart |
| Pn101 | Machine rigidity setting | – | 0 to 15 | 5 | Immediately |
| Pn128 | Speed gain acceleration relationship during online autotuning.<br>If the setting is greater, the servo gain will increase. | – | 0 to 3 | 3 | Immediately |

## 5.9.4 Machine Rigidity Setting for Online Autotuning

There are 16 machine rigidity settings for online autotuning, When the machine rigidity setting is selected, the servo gains (speed loop gain, speed loop integral time constant, position loop gain) are determined automatically. The factory setting for the machine rigidity setting is 5.

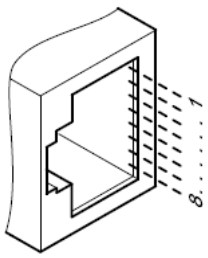
| Machine Rigidity Setting | Position Loop Gain [$s^{-1}$] Pn104 | Speed Loop Gain [Hz] Pn102=Pn104*( Pn128+1) | Speed Loop Integral Time Constant [0.25ms] Pn103 |
|---|---|---|---|
| 0 | 10 | 40 | 800 |
| 1 | 15 | 60 | 600 |
| 2 | 20 | 80 | 450 |
| 3 | 25 | 100 | 400 |
| 4 | 30 | 120 | 300 |
| 5 | 40 | 160 | 200 |
| 6 | 65 | 260 | 140 |
| 7 | 80 | 320 | 110 |
| 8 | 100 | 400 | 90 |
| 9 | 120 | 480 | 80 |
| 10 | 140 | 560 | 70 |
| 11 | 160 | 640 | 60 |
| 12 | 180 | 720 | 55 |
| 13 | 210 | 840 | 50 |
| 14 | 250 | 1000 | 40 |
| 15 | 300 | 1200 | 30 |

# Chapter 6 CANopen Communication

## 6.1 Wiring and Connection

The terminal layout of CN3 and CN4 is as follwing:

| Diagram | Pin | Name | Function |
|---|---|---|---|
|  | 1 | – | Reserved |
| | 2 | – | |
| | 3 | 485+ | RS-485 communication terminal |
| | 4 | ISO_GND | Isolated GND |
| | 5 | ISO_GND | |
| | 6 | 485- | RS-485 communication terminal |
| | 7 | CANH | CAN communication terminal |
| | 8 | CANL | CAN communication terminal |

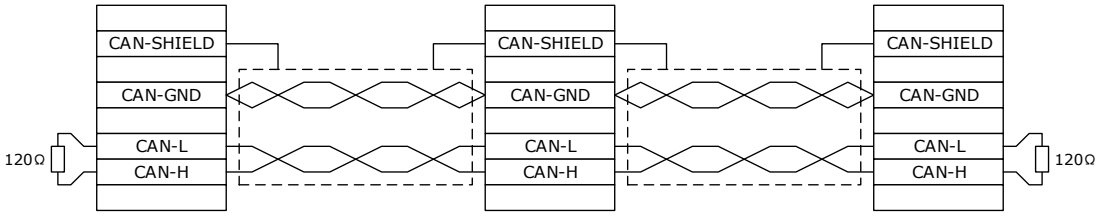
[NOTE] Do not short pin 1 and pin 2.
If you want to connecte more than 16 slave-stations, please contact ESTUN.

CN3 is always the input terminal of communication cable and CN4 is always the output terminal of communication cable. (If connection to another communication node is necessary, the cable will connect CN4 to next communication node. If not, a terminal resistor could be applied at CN4). When multiple ED3M devices are connected, it is forbidden to connect the CN3 terminals of different drives directly.

For example, a network is composed of one PLC, three ProNet drives called A, B and C. The cabling network is as below:

PLC → CN3 of drive A, CN4 of drive A → CN3 of drive B, CN4 of drive B → CN3 of drive C, CN4 of drive C → 120Ω resistor.

The two ends of the CAN cable have to be terminated by a resistor of 120Ω (1%, 1/4W) as below.




Please select the bus cable with double twisted pair cables and shielding layer, one pair for connecting CAN-L and CAN-H, another pair for grounding.

## 6.2 Messages Description

CAL supplies all network management service and message transferring protocol with defining the content of object or type of object for communication. It defines how instead of what, which is the strength of CANopen.

CANopen is developed based on CAL. It applies CAL protocol subsets for communication and service and creates a solution to DCS. CANopen could freely extend the node function to simplicity or complex while the network nodes are accessible and available to each other.

The key concept of CANopen is object dictionary. This way of object description is also applied to other fieldbus system like Profibus and Interbus-S. CANopen communication could access to all the parameter of drivers through object dictionary. Please notice object dictionary is not one part of CAL, instead of which it is realized in CANopen.

CANopen communication defines several types of objects as below.

| Abbreviation | Full Name | Description |
|---|---|---|
| SDO | Service Data Object | Used for normal parameterization of the servo controller |
| PDO | Process Data Object | Fast exchange of process data (e.g. velocity actual value) possible. |
| SYNC | Synchronization Message | Synchronization of several CAN nodes |
| EMCY | Emergency Message | Used to transmit error messages of the servo controller. |
| NMT | Network Management | Used for network services. For example usercan act on all controllers at the same time via this object type. |
| Heartbeat | Error Control Protocol | Used for observing all nodes by cyclic messages. |

CAN employs data frames for transferring data between the host (controller) and the nodes on the bus. The following figure presents the structure of the data frame.

| Head | Arbitration field | | Control field | Data field | Check field | Responsion field | End |
|---|---|---|---|---|---|---|---|
| | COB-ID | RTR | | | | | |
| 1 bit | 11 bits or 29 bits | 1 bit | 6 bits | 0 to 8 bits | 16 bits | 2 bits | 7 bits |

Our drivers doesn't support remote frame currently. The detail of COB-ID is as below.

| Function code | | | | Node ID | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

## 6.2.1 CAN identifier list

| Object | COB-ID bit10~7 (binary) | COB-ID (hex) | Index in OD |
|---|---|---|---|
| NMT | 0000 | $000_h$ | – |
| SYNC | 0001 | $080_h$ | $1005_h$, $1006_h$, $1007_h$ |
| TIME STAMP | 0010 | $100_h$ | $1012_h$, $1013_h$ |
| EMCY | 0001 | $081_h$ ~ $0FF_h$ | $1024_h$, $1015_h$ |
| PDO1 (send) | 0011 | $181_h$ ~ $1FF_h$ | $1800_h$ |
| PDO1 (receive) | 0100 | $201_h$ ~ $27F_h$ | $1400_h$ |
| PDO2 (send) | 0101 | $281_h$ ~ $2FF_h$ | $1801_h$ |
| PDO2 (receive) | 0110 | $301_h$ ~ $37F_h$ | $1401_h$ |
| PDO3 (send) | 0111 | $381_h$ ~ $3FF_h$ | $1802_h$ |
| PDO3 (receive) | 1000 | $401_h$ ~ $47F_h$ | $1402_h$ |
| PDO4 (send) | 1001 | $481_h$ ~ $4FF_h$ | $1803_h$ |

| Object | COB-ID bit10~7 (binary) | COB-ID (hex) | Index in OD |
|---|---|---|---|
| PDO4 (receive) | 1010 | $501_h$ ~ $57F_h$ | $1403_h$ |
| SDO (send) | 1011 | $581_h$ ~ $5FF_h$ | $1200_h$ |
| SDO (receive) | 1100 | $601_h$ ~ $67F_h$ | $1200_h$ |
| Heartbeat | 1110 | $701_h$ ~ $77F_h$ | $1016_h$, $1017_h$ |

[NOTE] PDO/SDO's send/receive is observed by (slave) CAN.
Our drive's CANopen protocol currently supports 4 transimit PDO and 4 receive PDO.

## 6.2.2 SDO

SDO is used to visit the object dictionary of a device. Visitor is called client. The CANopen device whose object dictionary is visited and required to supply the asked service is called server. CANopen messages from a client and servo all contain 8 bits (Not all of them are meaningful). A request from a client must be confirmed by a server

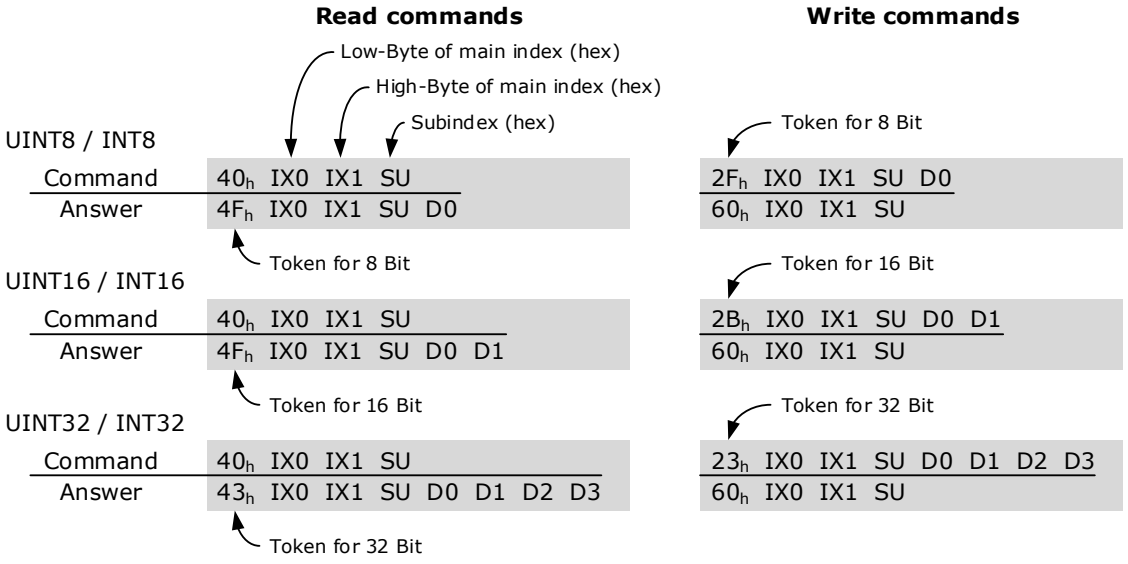
There are 2 method of conveying SDO:

- Expedited transfer: contains 4 bytes at maximum
- Segmented transfer: contains more than 4 bytes

Basic structure of SDO:

| Byte0 | Byte1~2 | Byte3 | Byte4~7 |
|---|---|---|---|
| SDO | Object index | Object sub-index | Data |

SDO read/write command structure:

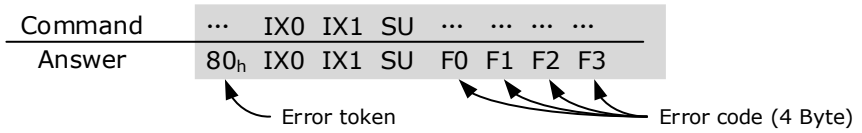


For example:

| | | |
|---|---|---|
| UINT8 / INT8 | Reading of Obj. $6061\_00_h$<br>Returning data: $01_h$ | Writing of Obj. $1401\_02_h$<br>Data: $EF_h$ |
| Command | $40_h$ $61_h$ $60_h$ $00_h$ | $2F_h$ $01_h$ $14_h$ $02_h$ $EF_h$ |
| Answer | $4F_h$ $61_h$ $60_h$ $00_h$ $01_h$ | $60_h$ $01_h$ $14_h$ $02_h$ |
| UINT16 / INT16 | Reading of Obj. $6041\_00_h$<br>Returning data: $1234_h$ | Writing of Obj. $6040\_00_h$<br>Data: $03E8_h$ |
| Command | $40_h$ $41_h$ $60_h$ $00_h$ | $2B_h$ $40_h$ $60_h$ $00_h$ $E8_h$ $03_h$ |
| Answer | $4B_h$ $41_h$ $60_h$ $00_h$ $34_h$ $12_h$ | $60_h$ $40_h$ $60_h$ $00_h$ |
| UINT32 / INT32 | Reading of Obj. $6093\_01_h$<br>Returning data: $12345678_h$ | Writing of Obj. $6093\_01_h$<br>Data: $12345678_h$ |
| Command | $40_h$ $93_h$ $60_h$ $01_h$ | $23_h$ $93_h$ $60_h$ $01_h$ $78_h$ $56_h$ $34_h$ $12_h$ |
| Answer | $43_h$ $93_h$ $60_h$ $01_h$ $78_h$ $56_h$ $34_h$ $12_h$ | $60_h$ $93_h$ $60_h$ $01_h$ |

SDO-error messages:




| Error code | | | | Description |
|---|---|---|---|---|
| F3 | F2 | F1 | F0 | |
| 05 | 03 | 00 | $00_h$ | Toggle bit not alternated |
| 05 | 04 | 00 | $01_h$ | Client/server command specifier not valid or unknown |
| 06 | 01 | 00 | $00_h$ | Unspported access to an object |
| 06 | 01 | 00 | $01_h$ | Attempt to read a write only object |
| 06 | 01 | 00 | $02_h$ | Attempt to write a read only object |
| 06 | 02 | 00 | $00_h$ | Object does not exist in the object dictionary |
| 06 | 04 | 00 | $41_h$ | Object cannot be mapped to the PDO |
| 06 | 04 | 00 | $42_h$ | The number and length of the objects to be mapped would exceed PDO length |
| 06 | 04 | 00 | $47_h$ | General internal incompatibilty in the device |
| 06 | 07 | 00 | $10_h$ | Data type does not match, length of service parameter does not match |
| 06 | 07 | 00 | $12_h$ | Data type does not match, length of service parameter too high |
| 06 | 07 | 00 | $13_h$ | Data type does not match, length of service parameter too low |
| 06 | 09 | 00 | $11_h$ | Sub-index does not exist |
| 06 | 04 | 00 | $43_h$ | General parameter incompatibility |
| 06 | 06 | 00 | $00_h$ | Access failed due to an hardware error |
| 06 | 09 | 00 | $30_h$ | Value range of parameter exceeded |
| 06 | 09 | 00 | $31_h$ | Value of parameter written too high |
| 06 | 09 | 00 | $32_h$ | Value of parameter written too low |
| 06 | 09 | 00 | $36_h$ | Maximum value is less than minumum value |
| 08 | 00 | 00 | $20_h$ | Data cannot be transferred or stored to the application |
| 08 | 00 | 00 | $21_h$ | Data cannot be transferred or stored to the application because of local control |

| Error code | | | | Description |
|---|---|---|---|---|
| F3 | F2 | F1 | F0 | |
| 08 | 00 | 00 | $22_h$ | Data cannot be transferred or stored to the application beacause of the present device state |
| 08 | 00 | 00 | $23_h$ | No Object Dictionary is present |

## 6.2.3 PDO

PDO is applied to transferring real time data which will be conveyed from a producer to one or multiple clients. Data transferring will be limited to 1 to 8 bytes. There is no hand-shake restriction in PDO communication, which means data has been redefined, so clients could process the received data for vary short time. PDO content will be only defined by its CAN ID, assuming producers and clients know PDO content from its CAN ID.

### Objects Description

Two objects in object dictionary are used for each PDO.

- PDO communication parameter: It contains COB-ID, transferring type, restriction time and cycle of timer used by PDO.
- PDO mapping parameter: It contains a list of objects in the object dictionary. These objects are mapped into PDO, includes their data length in bits. Producers and clients must know this mapping to explain the content of PDO.

The content of PDO's message is predefined or configured when the network initializes. Mapping application object into PDO is described in object dictionary. If a device (producer and client) support dynamic mapping, SDO could be used to configure PDO's mapping parameter. Our servo drive supports dynamic PDO mapping. There are 2 rules for PDO mapping to follow.

- Each PDO could be mapped into 4 objects.
- The length of each PDO will be no more than 64 bits.

### Mapping Process

1. Set the sub-index of PDO coordinated mapping parameter (1600 **h, 1601 h,** 1A00 **h** or 1A01 **h**) as o.
2. Revise the sub-index from 1 to 4 of PDO coordinated mapping parameter (1600 **h, 1601 h,** 1A00 **h** or 1A01 **h**).
3. Set the sub-index 0 of PDO coordinated mapping parameter(1600 **h** ,1601 **h,**1A00 **h** or 1A01 **h** ) as legal number( number of PDO's mapping objects)
4. PDO mapping was completed.

### Transmit Type

- Synchronous (Synchronization by receiving SYNC object)

  Cycle: Transmit triggered after every 1 to 240 SYNC messages.
- Asynchronous

  Transmit triggered by special object event regulated in sub-object protocol.

### Transmit Type List

| Value | Description | PDO |
|---|---|---|
| 0 | Reserved | – |
| 1~240 | **SYNC:** It represents the number of SYNC objects between 2 PDOs. | TPDO/RPDO |
| 240~253 | Reserved | – |

| Value | Description | PDO |
|---|---|---|
| 254 | **Asynchronous:** If the content of PDO has changed, PDO transmit will be triggered. | TPDO |
| 255 | **Asynchronous:** The content of PDO will be periodically updated and transmitted. | TPDO/RPDO |

One PDO could set a frozen time which is the shortest interval time between 2 continuous PDO. It could prevent the bus from being occupied by amount of data with high priority. Frozen time is defined by 16 bit unsigned integer number and its unit is 100us.

One PDO could set a timing period. When the regulated time is violated, a PDO transmit could be triggered without a trigger bit. Object timing period is defined as 16 bit unsigned integer and its unit is 1ms.

## Mapping Case

Map the 3 objects to PDO1 (transmit). PDO1 (transmit) is required to be asynchronous periodic type with period time as much as 10ms and frozen time as much as 2ms.

| Object | Index – Sub-index | Description |
|---|---|---|
| Statusword | $6041_{h}$ – $00_{h}$ | Status word |
| modes_of_operation_display | $6061_{h}$ – $00_{h}$ | Practical operational mode |
| Position_Actual_Value | $6064_{h}$ – $00_{h}$ | Practical position |

1. Clear number_of_mapped_objects
   number_of_mapped_objects(1A00 $_{h}$: 00 $_{h}$)= 0
2. Set the parameter for mapping objects
   Index =6041 h Subin. = 00h Length = 10 h ⇒ 1st_mapped_object(1A00 h: 01 h)= 60410010 h
   Index =6061 h Subin. = 00h Length = 08 h ⇒ 2st_mapped_object(1A00 h: 02 h)= 60610008 h
   Index =60FD h Subin. = 00h Length = 20 h ⇒ 3st_mapped_object(1A00 h: 03 h) = 60FD0020 h
3. Set number_of_mapped_objects
   number_of_mapped_objects(1A00 $_{h}$: 00 $_{h}$)= 3
4. Set PDO communication parameters
   PDO1 (transmit) is asynchronous periodical type ⇒ transmission_type (1800 h: 02 h)= FF h
   Frozen time 2ms(20×100us) ⇒ inhibit_time (10A0 h: 03 h)= 14 h
   Period time: 10ms(10×1ms) ⇒ event_time (1800 h: 05 h)= 0A h
5. PDO mapping was completed.

## PDO Parameter

ED3M drive contains 4 transmit PDOs and 4 receive PDOs. The detailed communication parameter and mapping parameter of the first transmit/receive PDO is as below and those of the rest 3 transmit/receive PDO are the same as the first PDO.

| Index | 1800 $_{h}$ |
|---|---|
| Name | transmit_pdo_parameter_tpdo1 |
| Object Code | RECORD |
| No. of Elements | 4 |

| Sub-Index | 01 $_{h}$ |
|---|---|
| Description | cob_id_used_by_pdo_tpdo1 |

| Sub-Index | 01 h |
|---|---|
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 181 h...1FF h, Bit 31 may be set |
| Default Value | 181 h |

| Sub-Index | 02 h |
|---|---|
| Description | transmission_type_tpdo1 |
| Data Type | UINT8 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 1...240,254,255 |
| Default Value | 255 |

| Sub-Index | 03 h |
|---|---|
| Description | inhibit_time_tpdo1 |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | 100μs |
| Value Range | – |
| Default Value | 100 |

| Sub-Index | 05 h |
|---|---|
| Description | event_time_tpdo1 |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | 1ms |
| Value Range | – |
| Default Value | 10 |

| Index | 1A00 h |
|---|---|
| Name | transmit_pdo_mapping_tpdo1 |
| Object Code | RECORD |
| No. of Elements | 2 |

| Sub-Index | 00 h |
|---|---|
| Description | number_of_mapped_objects_tpdo1 |
| Data Type | UINT8 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 0...4 |
| Default Value | 2 |

| Sub-Index | 01 h |
|---|---|
| Description | first_mapped_object_tpdo1 |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | See the table |

| Sub-Index | 02 h |
|---|---|
| Description | second_mapped_object_tpdo1 |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | See the table |

| Sub-Index | 03 h |
|---|---|
| Description | third_mapped_object_tpdo1 |
| Data Type | UINT32 |

| Sub-Index | 03 h |
|---|---|
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | See the table |

| Sub-Index | 04 h |
|---|---|
| Description | fourth_mapped_object_tpdo1 |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | See the table |

- T-PDO1

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1800 h _00 h | number of entries | UINT8 | RO | 04 h |
| 1800 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000181 h |
| 1800 h _02 h | transmission type | UINT8 | RW | FF h |
| 1800 h _03 h | inhibit time (100 μs) | UINT16 | RW | 64 h |
| 1800 h _05 h | event time (1ms) | UINT16 | RW | 0A h |
| 1A00 h _00 h | number of mapped objects | UINT8 | RW | 02 h |
| 1A00 h _01 h | first mapped object | UINT32 | RW | 60410010 h |
| 1A00 h _02 h | second mapped object | UINT32 | RW | 60640020 h |
| 1A00 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1A00 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- T-PDO2

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1801 h _00 h | number of entries | UINT8 | RO | 04 h |
| 1801 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000281 h |
| 1801 h _02 h | transmission type | UINT8 | RW | FF h |
| 1801 h _03 h | inhibit time (100 μs) | UINT16 | RW | 64 h |
| 1801 h _05 h | event time (1ms) | UINT16 | RW | 0A h |

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1A01 h _00 h | number of mapped objects | UINT8 | RW | 02 h |
| 1A01 h _01 h | first mapped object | UINT32 | RW | 60640020 h |
| 1A01 h _02 h | second mapped object | UINT32 | RW | 60610010 h |
| 1A01 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1A01 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- T-PDO3

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1802 h _00 h | number of entries | UINT8 | RO | 04 h |
| 1802 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000381 h |
| 1802 h _02 h | transmission type | UINT8 | RW | FF h |
| 1802 h _03 h | inhibit time (100 μs) | UINT16 | RW | 64 h |
| 1802 h _05 h | event time (1ms) | UINT16 | RW | 0A h |
| 1A02 h _00 h | number of mapped objects | UINT8 | RW | 00 h |
| 1A02 h _01 h | first mapped object | UINT32 | RW | 0 h |
| 1A02 h _02 h | second mapped object | UINT32 | RW | 0 h |
| 1A02 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1A02 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- T-PDO4

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1803 h _00 h | number of entries | UINT8 | RO | 04 h |
| 1803 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000481 h |
| 1803 h _02 h | transmission type | UINT8 | RW | FF h |
| 1803 h _03 h | inhibit time (100 μs) | UINT16 | RW | 64 h |
| 1803 h _05 h | event time (1ms) | UINT16 | RW | 0A h |
| 1A03 h _00 h | number of mapped objects | UINT8 | RW | 00 h |
| 1A03 h _01 h | first mapped object | UINT32 | RW | 0 h |
| 1A03 h _02 h | second mapped object | UINT32 | RW | 0 h |
| 1A03 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1A03 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

If transmit tye is 254 (if PDO content has changed,trigger will be sent by PDO),using the following object can shield parts of PDO changers.Only when the un-shield bit has changed， PDO is occur.If wants shielding any bit, the corresponding bit of object write to 0.

- tpdo_1_transmit_mask

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 2000 h _00 h | number of entries | UINT8 | RO | 02 h |
| 2000 h _01 h | tpdo_1_transmit_mask_low | UINT32 | RW | FFFFFFFF h |
| 2000 h _02 h | tpdo_1_transmit_mask_high | UINT32 | RW | FFFFFFFF h |

- tpdo_2_transmit_mask

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 2001 h _00 h | number of entries | UINT8 | RO | 02 h |
| 2001 h _01 h | tpdo_2_transmit_mask_low | UINT32 | RW | FFFFFFFF h |
| 2001 h _02 h | tpdo_2_transmit_mask_high | UINT32 | RW | FFFFFFFF h |

- tpdo_3_transmit_mask

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 2002 h _00 h | number of entries | UINT8 | RO | 02 h |
| 2002 h _01 h | tpdo_1_transmit_mask_low | UINT32 | RW | FFFFFFFF h |
| 2002 h _02 h | tpdo_1_transmit_mask_high | UINT32 | RW | FFFFFFFF h |

- tpdo_4_transmit_mask

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 2003 h _00 h | number of entries | UINT8 | RO | 02 h |
| 2003 h _01 h | tpdo_2_transmit_mask_low | UINT32 | RW | FFFFFFFF h |
| 2003 h _02 h | tpdo_2_transmit_mask_high | UINT32 | RW | FFFFFFFF h |

- R-PDO1

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1400 h _00 h | number of entries | UINT8 | RO | 02 h |
| 1400 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000201 h |
| 1400 h _02 h | transmission type | UINT8 | RW | FF h |
| 1600 h _00 h | number of mapped objects | UINT8 | RW | 02 h |
| 1600 h _01 h | first mapped object | UINT32 | RW | 60400010 h |
| 1600 h _02 h | second mapped object | UINT32 | RW | 60FF0020 h |
| 1600 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1600 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- R-PDO2

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1401 h _00 h | number of entries | UINT8 | RO | 02 h |
| 1401 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000301 h |

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1401 h _02 h | transmission type | UINT8 | RW | FF h |
| 1601 h _00 h | number of mapped objects | UINT8 | RW | 02 h |
| 1601 h _01 h | first mapped object | UINT32 | RW | 60FF0020 h |
| 1601 h _02 h | second mapped object | UINT32 | RW | 60600010 h |
| 1601 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1601 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- R-PDO3

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1402 h _00 h | number of entries | UINT8 | RO | 02 h |
| 1402 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000401 h |
| 1402 h _02 h | transmission type | UINT8 | RW | FF h |
| 1602 h _00 h | number of mapped objects | UINT8 | RW | 00 h |
| 1602 h _01 h | first mapped object | UINT32 | RW | 0 h |
| 1602 h _02 h | second mapped object | UINT32 | RW | 0 h |
| 1602 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1602 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

- R-PDO4

| Index | Comment | Type | Acc. | Default Value |
|---|---|---|---|---|
| 1403 h _00 h | number of entries | UINT8 | RO | 02 h |
| 1403 h _01 h | COB-ID used by PDO | UINT32 | RW | 00000501 h |
| 1403 h _02 h | transmission type | UINT8 | RW | FF h |
| 1603 h _00 h | number of mapped objects | UINT8 | RW | 00 h |
| 1603 h _01 h | first mapped object | UINT32 | RW | 0 h |
| 1603 h _02 h | second mapped object | UINT32 | RW | 0 h |
| 1603 h _03 h | third mapped object | UINT32 | RW | 00 h |
| 1603 h _04 h | fourth mapped object | UINT32 | RW | 00 h |

### 6.2.4 SYNC Message

Synchronization object is used for controlling data synchronize transmit. For example: starting synchronously several axises. The transmition of synchronous message is based on Producer-Customer model. All the nodes of synchronous PDO can receive (at the same time) the message as customer and synchronize other node.

CANopen suggests a COB-ID with highest priority to ensure that synchronized signal could be transmitted properly. Without transferring data, SYNC message could be as short as possible.

The identifier the servo controller receives SYNC messages are fixed to 080h. The identifier can be read via the object **cob_id_sync**.

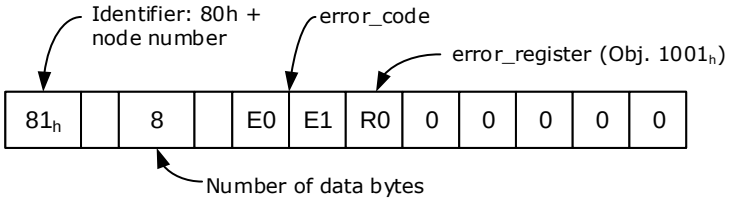
  

| Index | 1005 h |
|---|---|
| Name | cob_id_sync |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 80000080 h, 00000080 h |
| Default Value | 00000080 h |

## 6.2.5 Emergency Message

When an alarm occurs to drive, CANopen will initiate an Emergency message to inform the current drive type and error code to clients. Error code displayed on panel can be read on low byte of 603Fh object.

The structure of Emergency message:




**Alarm Codes**

| error_code (Hex) | Description |
|---|---|
| 2310 | Over current |
| 3100 | Instantaneous power failure |
| 3110 | Over voltage |
| 3120 | Under voltage |
| 5080 | RAM exception |
| 5210 | AD sampling error |
| 5420 | Regenerative resistor error |
| 5421 | Regenerative resistor exception |
| 5581 | Parameter checksum exception |
| 5582 | electric gear error |
| 5583 | Motor type or drive type error |
| 6100 | Illegal error code |
| 6120 | PDO mapping error |
| 6300 | CAN communication error(Address or communication baud rate error) |
| 7303 | serial encoder error |

| error_code (Hex) | Description |
|---|---|
| 7305 | Incremental encoder error |
| 7380 | Resolver error |
| 8100 | CAN communication exception |
| 8110 | CAN bus overflow |
| 8120 | PASSIVE CAN bus turn to PASSIVE |
| 8130 | Heartbeat error |
| 8140 | CAN BUS OFF |
| 8200 | Length of CAN messages error |
| 8210 | Length of receiving PDO error |
| 8311 | Overload alarm |
| 8480 | Over speed alarm |
| 8681 | Forward run prohibited POT |
| 8682 | Reverse run prohibited NOT |

## Relevant Parameters

| Index | 1003 h |
|---|---|
| Name | pre_defined_error_field |
| Object Code | ARRAY |
| No. of Elements | 4 |
| Data Type | UINT32 |

| Sub-Index | 01 h |
|---|---|
| Description | standard_error_field_0 |
| Access | RO |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | – |

| Sub-Index | 02 h |
|---|---|
| Description | standard_error_field_1 |
| Access | RO |
| PDO Mapping | NO |
| Units | – |

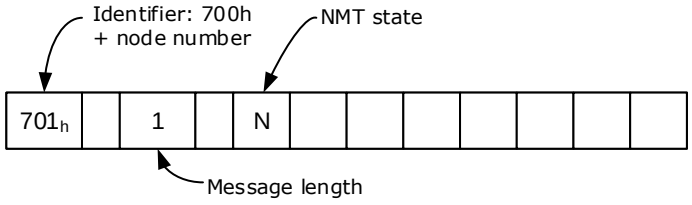
| Sub-Index | 02 h |
|---|---|
| Value Range | – |
| Default Value | – |

| Sub-Index | 03 h |
|---|---|
| Description | standard_error_field_2 |
| Access | RO |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | – |

| Sub-Index | 04 h |
|---|---|
| Description | standard_error_field_3 |
| Access | RO |
| PDO Mapping | NO |
| Units | – |
| Value Range | – |
| Default Value | – |

## 6.2.6 HEARTBEAT Message

**Structure**




**Relevant Parameter**

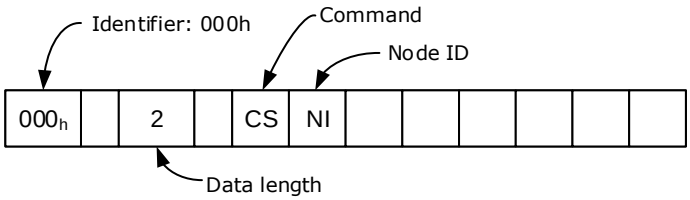
| Index | 1017 h |
|---|---|
| Name | producer_heartbeat_time |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | ms |

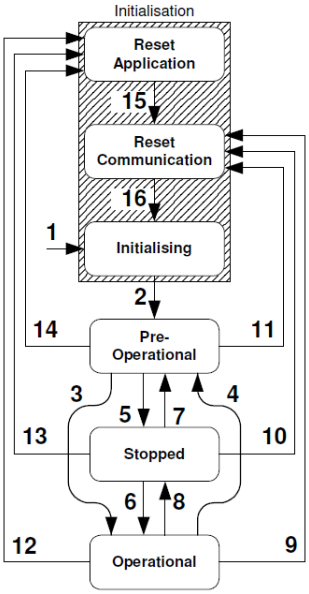
| Index | 1017 h |
|---|---|
| Value Range | 0 - 65535 |
| Default Value | 1000 |

## 6.2.7 Network management (NMT service)

**Structure**




**NMT-State machine**




| CS | Meaning | Transition | Target state |
|---|---|---|---|
| 01h | Start Remote Node | 3,6 | Operational |
| 02h | Stop Remote Node | 5,8 | Stopped |
| 80h | Enter Pre-Operational | 4,7 | Pre-Operational |
| 81h | Reset Application | 12,13,14 | Reset Application |
| 82h | Reset Communication | 9,10,11 | Reset Communication |

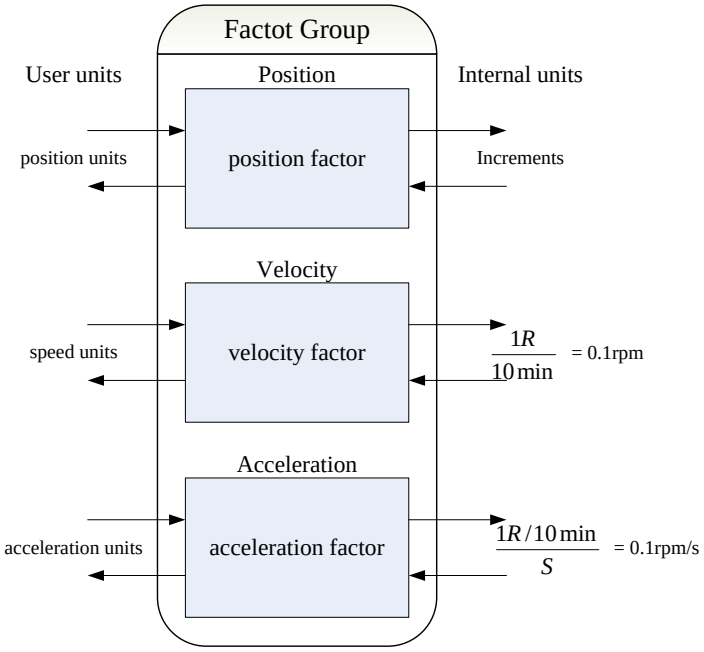
| Name | Meaning | SDO | PDO | NMT |
|---|---|---|---|---|
| Reset Application | No communication. All CAN objects are set to their reset values (application parameter set). | – | – | – |
| Reset Communication | No communication. The CAN controller will be re-initialised. | – | – | – |

| Name | Meaning | SDO | PDO | NMT |
|---|---|---|---|---|
| Initiallsing | State after Hardware Reset. Reset of the CAN node, sending of the Bootup message. | – | – | – |
| Pre-Operational | Communication via SDOs possible. PDOs inactive (No sending/receiving) | × | – | × |
| Operational | Communication via SDOs possible. PDOs active (sending/receiving) | × | × | × |
| Stopped | No communication except heartbeat + NMT | – | – | × |

# 6.3 Conversion Factors (Factor Group)

Servo controllers will be used in a huge number of applications: As direct drive, with gear or for linear drives. To allow an easy parameterization for all kinds of applications, the servo controller can be parameterized in such a way that all values like the demand velocity refer to the driven side of the plant. The necessary calculation is done by the servo controller.




The default setting of the Factor Group is as follows:

| Object | Name | Unit | Description |
|---|---|---|---|
| Length | position units | Increments | Increments per revolution * |
| Velocity | speed units | 1R /10min | 0.1rpm |
| Acceleration | Acceleration units | 1R/10min/s | 0.1rpm/s |
| Jerk | jerk units | pulse/(s*100μs*100μs) | Range:1-20, more smaller,more smooth |

* : Common incremental encoder: 10000P/R
Resolver: 65536P/R
17 bit incremental encoder: 131072P/R
17 bit absolute encoder: 131072P/R
20 bit absolute encoder: 1048576P/R

### 6.3.1 Relevant parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6093 h | ARRAY | position factor | UINT32 | RW |
| 6094 h | ARRAY | velocity factor | UINT32 | RW |
| 6097 h | ARRAY | acceleration factor | UINT32 | RW |

### 6.3.2 Position factor

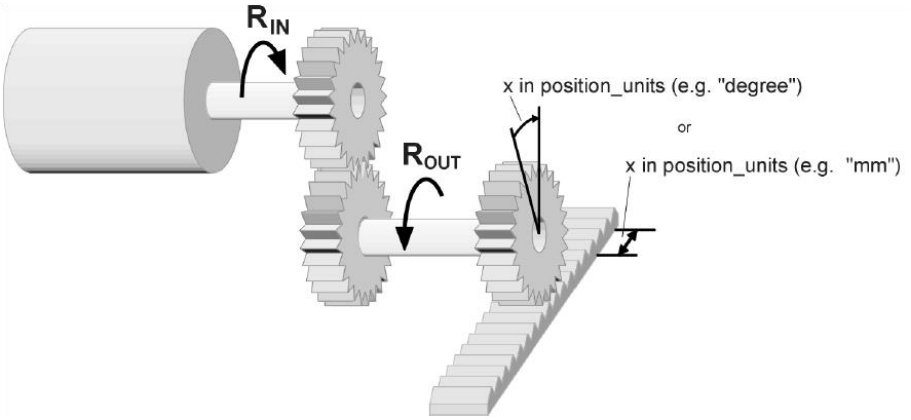
The object **position factor** converts all values of length of the application from **Position units** into the internal unit **increments** (*encoder resolution* equals 1 Revolution). It consists of numerator and divisor:

| Index | 6093 h |
|---|---|
| Name | position factor |
| Object Code | ARRAY |
| No. of Elements | 2 |
| Data Type | UINT32 |

| Sub-Index | 01 h |
|---|---|
| Description | numerator |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |
| Default Value | When power on, this value will be initiated to parameter Pn201 |

| Sub-Index | 02 h |
|---|---|
| Description | division |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |
| Default Value | When power on, this value will be initiated to parameter Pn202 |




To calculate the **position factor** the following values are necessary:

- gear_ratio
  Ratio between revolutions on the driving side (RIN) and revolutions on the driven side (ROUT).
- feed_constant
  Ratio between revolutions on the driven side (ROUT) and equivalent motion in **position_units** (e.g. 1 rev = 360°)

The calculation of the position_factor is done with the following equation:

$$\text{position factor} = \frac{numerator}{division} = \frac{gear\_ratio * encoder\_resolution}{feed\_constant}$$

| **Encoder type** | **Encoder_resolution(Unit: Inc)** |
|---|---|
| Incremental encoder | 10000 |
| Resolver | 65535 |
| 17-bit encoder | 131072 |
| 20-bit encoder | 1048576 |

## 6.3.3 Velocity factor

The object **velocity factor** converts all speed values of the application from **speed_units** into the internal unit **revolutions 0.1rpm**. It consists of numerator and divisor

| Index | 6094 h |
|---|---|
| Name | velocity factor |
| Object Code | ARRAY |
| No. of Elements | 2 |
| Data Type | UINT32 |

| Sub-Index | 01 h |
|---|---|
| Description | numerator |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |

| Sub-Index | 01 h |
|---|---|
| Default Value | 1 |

| Sub-Index | 02 h |
|---|---|
| Description | division |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |
| Default Value | 1 |

In principle the calculation of the **velocity factor** is composed of two parts: A conversion factor from internal units of length into **position_units** and a conversion factor from internal time units into user defined time units (e.g. from seconds to minutes). The first part equals the calculation of the **position_factor.** For the second part another factor is necessary for the calculation:

- time_factor_v
  Ratio between internal and user defined time units. (z.B. **1** ***min*** = **1/10** ***10 min***)
- gear_ratio
  Ratio between revolutions on the driving side (RIN) and revolutions on the driven side (ROUT).
- feed_constant
  Ratio between revolutions on the driven side (ROUT) and equivalent motion in position_units (e.g. 1 R = 360°)

The calculation of the velocity factor is done with the following equation:

$$\text{velocity factor} = \frac{numerator}{division} = \frac{gear\_ratio * time\_factor\_v}{feed\_constant}$$

## 6.3.4 Acceleration factor

The object **acceleration_factor** converts all acceleration values of the application from **acceleration_units** into the internal unit (0.1rpm) . It consists of numerator and divisor:

| Index | 6097 h |
|---|---|
| Name | acceleration factor |
| Object Code | ARRAY |
| No. of Elements | 2 |
| Data Type | UINT32 |

| Sub-Index | 01 h |
|---|---|
| Description | numerator |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |

| Sub-Index | 01 h |
|---|---|
| Default Value | 1 |

| Sub-Index | 02 h |
|---|---|
| Description | division |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |
| Default Value | 1 |

The calculation of the **acceleration_factor** is also composed of two parts: A conversion factor from internal units of length into **position_units** and a conversion factor from internal time units squared into user defined time units squared (e.g. from seconds2 to minutes2). The first part equals the calculation of the **position_factor.** For the second part another factor is necessary for the calculation.

- time_factor_a

  Ratio between internal time units squared and user defined time units squared
  (z.B.: 1min2 = 1min*min = 60s*1min =60/10 10min/s)
- gear_ratio

  Ratio between revolutions on the driving side (RIN) and revolutions on the driven side (ROUT).
- feed_constant

  Ratio between revolutions on the driven side (ROUT) and equivalent motion in position_units (e.g. 1 R = 360°)

The calculation of the acceleration_factor is done with the following equation:

$$\text{acceleration factor} = \frac{numerator}{division} = \frac{gear\_ratio * time\_factor\_v}{feed\_constant}$$

# 6.4 Position Control Function

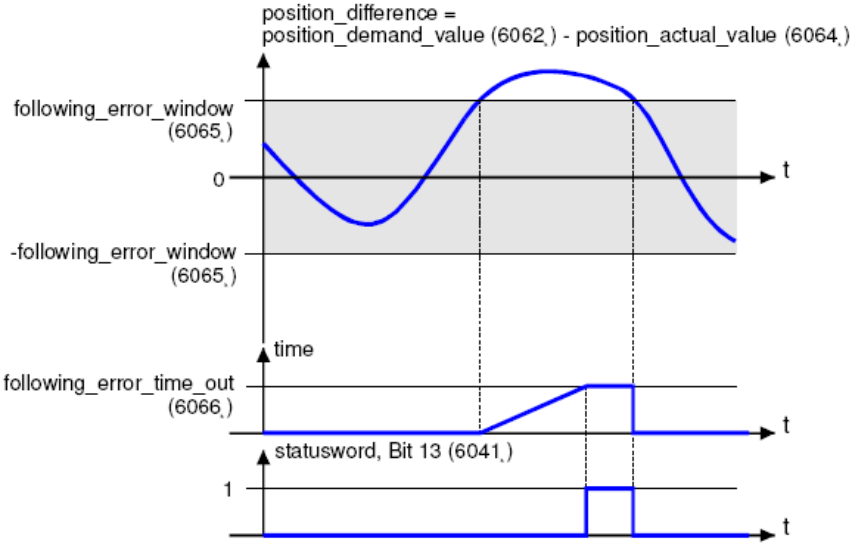
This chapter describes all parameters which are required for the position controller. The desired position value (**position_demand_value**) of the trajectory generator is the input of the position controller. Besides this the actual position value (**position_actual_value**) is supplied by the angle encoder (resolver, incremental encoder, etc.). The behaviour of the position controller can be influenced by parameters.

It is possible to limit the output quantity (**control_effort**) in order to keep the position control system stable. The output quantity is supplied to the speed controller as desired speed value. In the **Factor Group** all input and output quantities are converted from the application-specific units to the respective internal units of the controller

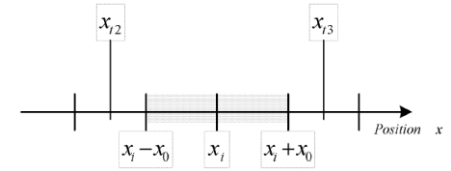
The following subfunctions are defined in this chapter:

## 6.4.1 Following error)




The deviation of the actual position value (**position_actual_value**) from the desired position value (**position_demand_value)** is named trailing error. If for a certain period of time this trailing error is bigger than specified in the trailing error window (**following_error_window**) bit 13 (**following_error**) of the object **statusword** will be set to 1.




The permissible time can be defined via the object **following_error_time_out**.. Figure above shows how the window function is defined for the message "following error".The range between xi-x0 and xi+x0 is defined symmetrically around the desired position (**position_demand_value**) xi. For example the positions xt2 and xt3 are outside this window (**following_error_window**). If the drive leaves this window and does not return to the window within the time defined in the object **following_error_time_out** then bit 13 (**following_error**) in the **statusword** will be set to 1.

## 6.4.2 Position Reached

This function offers the chance to define a position window around the target position (**target_position**). If the actual position of the drive is within this range for a certain period of time – the **position_window_time** – bit 10 (**target_reached**) will be set to 1 in the statusword.

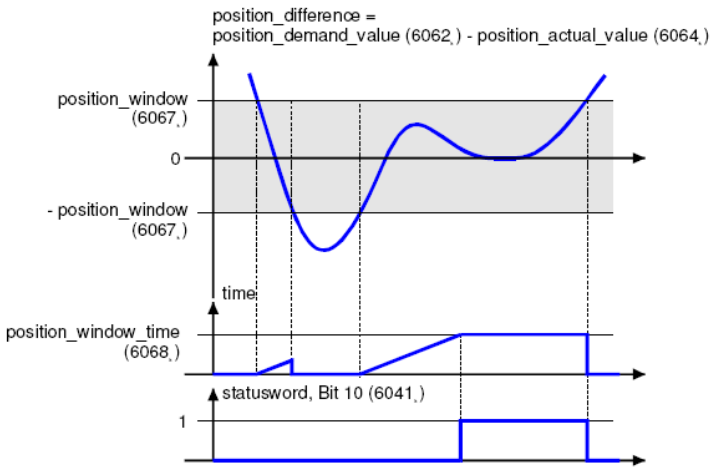
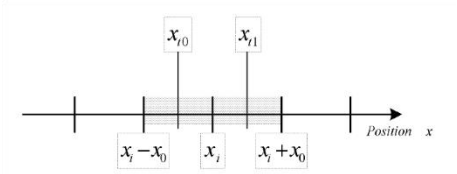


Figure below shows how the window function is defined for the message ”position reached”. The position range between xi-x0 and xi+x0 is defined symmetrically around the target position (**target_position**) xi. For example the positions xt0 and xt1 are inside this position window (**position_window**). If the drive is within this window a timer is started. If this timer reaches the time defined in the object **position_window_time** and the drive uninterruptedly was within the valid range between xi-x0 and xi+x0, bit 10 (**target_reached**) will be set in the **statusword**. As far as the drive leaves the permissible range, bit 10 is cleared and the timer is set to zero.




## 6.4.3 Relevant Parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6062 h | VAR | position_demand_value | INT32 | RO |
| 6063 h | VAR | position_actual_value* | INT32 | RO |
| 6064 h | VAR | position_actual_value | INT32 | RO |
| 6065 h | VAR | following_error_window | UINT32 | RW |
| 6066 h | VAR | following_error_time_out | UINT16 | RW |
| 6067 h | VAR | position_window | UINT32 | RW |
| 6068 h | VAR | position_time | UINT16 | RW |
| 60FA h | VAR | control_effort | INT32 | RO |

| Index | 6062 h |
|---|---|
| Name | position_demand_value |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RO |
| PDO Mapping | YES |
| Units | position units |
| Value Range | – |
| Default Value | – |

| Index | 6064 h |
|---|---|
| Name | position_ actual _value |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RO |
| PDO Mapping | YES |

| Index | 6064 h |
|---|---|
| Units | position units |
| Value Range | – |
| Default Value | – |

| Index | 6065 h |
|---|---|
| Name | following_error_window |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | position units |
| Value Range | 0 – 7FFFFFFF h |
| Default Value | 30000 |

| Index | 6066 h |
|---|---|
| Name | following_error_time_out |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | ms |
| Value Range | 0 – 65535 |
| Default Value | 200 |

| Index | 60FA h |
|---|---|
| Name | control_effort |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RO |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | – |

| Index | 6067 h |
|---|---|
| Name | position_window |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | position units |
| Value Range | – |
| Default Value | 10 |

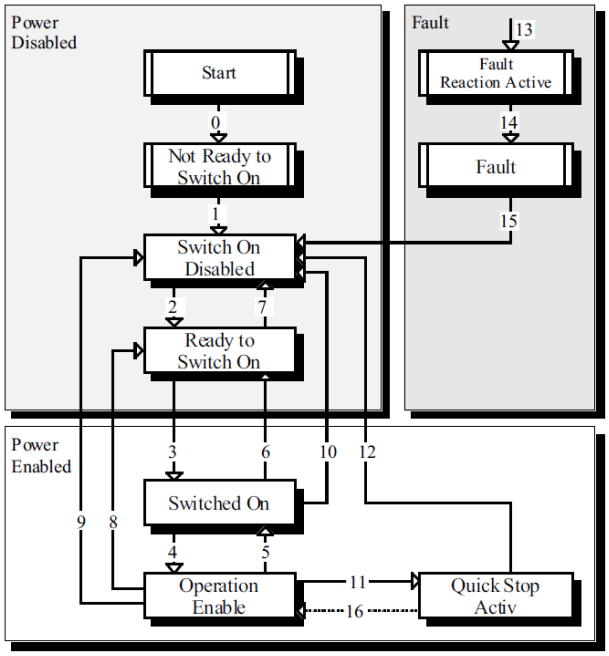
| Index | 6068 h |
|---|---|
| Name | position_time |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | ms |
| Value Range | 0 – 65535 |
| Default Value | 50 |

## 6.5 State machine

Using CANopen the complete control of the servo is done by two objects. Via the **controlword** the host is able to control the servo, as the status of the servo can be read out of the **statusword**. The following items will be used in this chapter:

| Term | Description |
|---|---|
| State | The servo controller is in different states dependent on for instance if the power stage is alive or if an error has occurred. States defined under CANopen will be explained in this chapter. |
| State Transition | Just as the states it is defined as well how to move from one state to another (e.g. to reset an error). These state transitions will be either executed by the host by setting bits in the **controlword** or by the servo controller itself, if an error occurs for instance. |
| Command | To initiate a state transition defined bit combinations have to be set in the **controlword**. Such bit combination are called command. Example: **Enable Operation**. |
| State diagram | All the states and all state transitions together form the so called state diagram: A survey of all states and the possible transitions between two states. |




The state diagram can be divided into three main parts: "Power Disabled" means the power stage is switched off and "Power Enabled" the power stage is active. The area "Fault" contains all states necessary to handle errors of the controller. The most important states have been highlighted in the Figure: After switching on the servo controller initializes itself and reaches the state **SWITCH_ON_DISABLED** after all. In this state CAN communication is possible and the servo controller can be parameterized (e.g. the mode of operation can be set to "velocity control"). The power stage remains switched off and the motor shaft is freely rotatable. Through the state transitions 2, 3 and 4 – principally like the controller enable under CANopen - the state **OPERATION_ENABLE** will be reached. In this state the power stage is live and the servo controller controls the motor according to the parameterized mode of operation. Therefore previously ensure that the servo controller has been parameterized correctly and the according demand value is zero. The state transition 9 complies with disabling the power stage, i.e. the motor is freely rotatable.

| Status | Description |
|---|---|
| Not Ready to Switch On | The servo controller executes its self-test. The CAN communication is not working |
| Switch On Disabled | The self-test has been completed. The CAN communication is activated.. |
| Ready to Switch On | Servo driver is waiting for the state of Switch and servo motor is not at power stage |
| Switched On | The power stage is alive. |
| Operation Enable | The motor is under voltage and is controlled according to operational mode |
| Quick Stop Active | Servo driver will be stopped through its fixed way, |
| Fault Reaction Active | Servo driver tests error and will be stopped through its fixed way, with motor's power stage alive |
| Fault | An error has occurred. The power stage has been switched off. |

## 6.6 Relevant Parameters of Device Control

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6040 h | VAR | controlword | UINT16 | RW |

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6041 h | VAR | statusword | UINT16 | RO |
| 605A h | VAR | quick_stop_option_code | INT16 | RW |
| 605B h | VAR | shutdown_option_code | INT16 | RW |
| 605C h | VAR | disabled_operation_option_code | INT16 | RW |
| 605D h | VAR | halt_option_code | INT16 | RW |
| 605E h | VAR | fault_reaction_option_code | INT16 | RW |

## 6.6.1 Controlword

| Index | 6040 h |
|---|---|
| Name | controlword |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | -- |
| Value Range | -- |
| Default Value | 0 |

Controlword bit description is as below:

| 15 11 | 10 9 | 8 | 7 | 6 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Manufacturer specific | Reserved | halt | Fault reset | Operation mode specific | Enable operation | Quick stop | Enable voltage | Switch on |

**Bit0 ~ 3 and Bit7**

Transmit of status machine is triggered by 5 bits coordinated control code as below

| Command | Bit of the controlword | | | | | |
|---|---|---|---|---|---|---|
| | **Fault reset** | **Enable operation** | **Quick stop** | **Enable voltage** | **Switch on** | **Transitions** |
| Shutdown | 0 | × | 1 | 1 | 0 | 2,6,8 |
| Switch on | 0 | 0 | 1 | 1 | 1 | 3* |
| Switch on | 0 | 1 | 1 | 1 | 1 | 3** |
| Disable voltage | 0 | × | × | 0 | × | 7,9,10,12 |
| Quick stop | 0 | × | 0 | 1 | × | 7,9,10,11 |
| Disable operation | 0 | 0 | 1 | 1 | 1 | 5 |
| Enable operation | 0 | 1 | 1 | 1 | 1 | 4,16 |

| Command | Bit of the controlword | | | | | |
|---|---|---|---|---|---|---|
| | **Fault reset** | **Enable operation** | **Quick stop** | **Enable voltage** | **Switch on** | **Transitions** |
| Fault reset |  | × | × | × | × | 15 |

NOTE: X means this bit could be ignored.

**Bit4, 5, 6, 8**

The definition of this 4 bit is different in different control mode.

| Bit | Control Mode | | |
|---|---|---|---|
| | **profile position mode** | **profile velocity mode** | **homing mode** |
| 4 | Newsetpoint | Reserved | Start homingoperation |
| 5 | Changesetimmediately | Reserved | Reserved |
| 6 | abs/rel | Reserved | Reserved |
| 8 | Halt | Halt | Halt |

**Other bits**

All reserved.

## 6.6.2 Statusword

| Index | 6041 H |
|---|---|
| Name | Statusword |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RO |
| PDO Mapping | YES |
| Units | – |
| Value Range | – |
| Default Value | – |

Explanation of statusword bit is as below:

| Bit | Description |
|---|---|
| 0 | Ready to switch on |
| 1 | Switched on |
| 2 | Operation enabled |
| 3 | Fault |
| 4 | Voltage enabled |
| 5 | Quick stop |

| Bit | Description |
|---|---|
| 6 | Switch on disabled |
| 7 | Warning |
| 8 | Reserved |
| 9 | Remote |
| 10 | Target reached |
| 11 | Internal limit active |
| 12~13 | Operation mode specific |
| 14~15 | Reserved |

### Bit0 ~ 3, Bit5, and Bit6

The combination of this bit indicates the status of drives.

| Value (binary) | State |
|---|---|
| xxxx xxxx x0xx 0000 | Not ready to switch on |
| xxxx xxxx x1xx 0000 | Switch on disabled |
| xxxx xxxx x01x 0001 | Ready to switch on |
| xxxx xxxx x01x 0011 | Switched on |
| xxxx xxxx x01x 0111 | Operation enabled |
| xxxx xxxx x00x 0111 | Quick stop active |
| xxxx xxxx x0xx 1111 | Fault reaction active |
| xxxx xxxx x0xx 1000 | Fault |

### Bit4: Voltage enabled

Main power is on when this bit is 1.

### Bit5: Quick stop

Driver will follow setting (605A $_h$: quick_stop_option_code) to halt when this bit is 0.

### Bit7: Warning

Driver detects alarm when this bit is 1.

### Bit9: Warning

Servo can deal with Controlword when this bit is 1 and CANOPEN is enabled.

### Bit10: Target reached

In different control modes the meaning of this bit is different.

- In profile position mode, when set position is reached, this bit is set. When Halt is booted, speed is reduced to 0 and this bit will be set. When new position is set, this bit will be cleared.

- In profile Velocity Mode, when the speed reaches the targeted speed, this bit will be set. When Halt is booted and speed is reduced to 0, this bit is set.

**Bit11: Internal limit active**

When this bit is 1, it indicates that internal torque has surpassed the set value,or reached the max.forward/reverse run.It can be confirmed by reading object 60FDh (digital inputs) .

**Bit12, 13**

These 2 bits mean different in different control mode.

| Bit | Control mode | | |
|---|---|---|---|
| | **profile position mode** | **profile velocity mode** | **homing mode** |
| 12 | Set-point acknowledge | Speed | Homing attained |
| 13 | Following error | Max slippage error | Homing error |

**Other bits**

All reserved.

## 6.6.3 Shutdown_option_code

The object **shutdown_option_code** determines the behaviour if the state transition 8 (from OPERATION ENABLE to READY TO SWITCH ON) will be executed.

| Index | 605B h |
|---|---|
| Name | shutdown_option_code |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 0,1 |
| Default Value | 0 |

| Value | Description |
|---|---|
| 0 | Power stage will be switched off. Motor is freely rotatable. |
| 1 | Switch off the power stage after the motor stops deceleration. |

## 6.6.4 Disable_operation_option_code

The object **disable_operation_option_code** determines the behaviour if the state transition 5 (from **OPERATION ENABLE** to **SWITCHED ON)** will be executed.

| Index | 605C h |
|---|---|
| Name | disable_operation_option_code |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 0,1 |
| Default Value | 0 |

| Value | Description |
|---|---|
| 0 | Power stage will be switched off. Motor is freely rotatable. |
| 1 | Switch off the power stage after the motor stops deceleration. |

## 6.6.5 Quick_stop_option_code

The object **quick_stop_option_code** determines the behaviour if a **Quick Stop** will be executed.

| Index | 605A h |
|---|---|
| Name | quick_stop_option_code |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 0,1,2,5,6 |
| Default Value | 2 |

| Value | Description |
|---|---|
| 0 | Power stage will be switched off. Motor is freely rotatable. |
| 1 | Switch off the power stage after the motor stops deceleration. |
| 2 | Power stage will be shut down after the motor decelerates to still urgently. |
| 5 | QuickStop is alive after the motor decelerates to still. |
| 6 | QuickStop is alive after the motor decelerates urgently to still. |

## 6.6.6 Halt_option_code

Halt_option_code determines how to stop when bit.8 (halt) of controlword is set to 1.

| Index | 605D h |
|---|---|
| Name | halt_option_code |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 1,2 |
| Default Value | 0 |

| Value | Description |
|---|---|
| 1 | The motor decelerates to still. |
| 2 | The motor decelerates urgently to still |

## 6.6.7 Fault_reaction_option_code

When an error is occurred, fault_reation_option_code determines how to stop.

| Index | 605E h |
|---|---|
| Name | fault_reaction_option_code |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | – |
| Value Range | 0 |
| Default Value | 0 |

| Value | Description |
|---|---|
| 0 | Power stage will be switched off. Motor is freely rotatable. |

# Chapter 7 CANopen Control Mode

EM3A currently supports 5 control modes in CANopen DSP402:

- HOMING MODE
- PROFILE VELOCITY MODE
- PROFILE TORQUE MODE
- PROFILE POSITION MODE
- INTERPOLATED POSITION MODE

## 7.1 Relevant Parameter of Control Mode

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6060 h | VAR | modes_of_operation | INT8 | RW |
| 6061 h | VAR | modes_of_operation_display | INT8 | RO |

### 7.1.1 Modes_of_operation

Drive control mode will be determined by parameters in modes_of_operation.

| Index | 6060 h |
|---|---|
| Name | modes_of_operation |
| Object Code | VAR |
| Data Type | INT8 |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | 1,3,4,6,7 |
| Default Value | 1 |

| Value | Description |
|---|---|
| 1 | PROFILE POSITION MODE |
| 3 | PROFILE VELOCITY MODE |
| 4 | PROFILE TORQUE MODE |
| 6 | HOMING MODE |
| 7 | INTERPOLATION MODE |

### 7.1.2 Modes_of_operation_display

Drive current control mode could be read from parameters in modes_of_operation_display.

| Index | 6061 h |
|---|---|
| Name | modes_of_operation_display |
| Object Code | VAR |
| Data Type | INT8 |
| Access | RO |
| PDO Mapping | YES |
| Units | – |
| Value Range | 1,3,4,6,7 |
| Default Value | 1 |

NOTE: The current control mode could be only known from parameters in modes_of_operation_display.

## 7.2 Homing Mode

ED3M servo drive currently supports multiple homing mode and users could choose the suitable homing mode. For example, if an incremental encoder is applied in servomotor, then homing mode of Zero impulse could be chosen and if serial encoder or resolver is applied in servomotor then Zero impulse homing mode couldn't be selected.

The user can determine the velocity, acceleration, and the kind of homing operation. After the servo controller has found its reference the zero position can be moved to the desired point via the object home_offset (607C h) .

### 7.2.1 Control Word

| 15 ~ 9 | 8 | 7 ~ 5 | 4 | 3 ~ 0 |
|---|---|---|---|---|
| * | Halt | * | home_ operation _ start | * |

*: referred to previous chapters

| Name | Value | Description |
|---|---|---|
| Homingoperationstart | 0 | Homing mode inactive |
| | 0 → 1 | Start homing mode |
| | 1 | Homing mode active |
| | 1 → 0 | Interrupt homing mode |
| Halt | 0 | Execute the instruction of bit 4 |
| | 1 | Stop axle with homing acceleration |

### 7.2.2 Status Word

| 15 ~ 14 | 13 | 12 | 11 | 10 | 9 ~ 0 |
|---|---|---|---|---|---|
| * | homing_error | homing_attained | * | target_reached | * |

*: referred to previous chapters

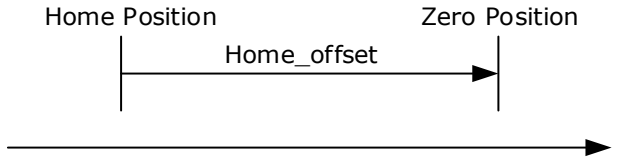
| Name | Value | Description |
|---|---|---|
| Target reached | 0 | Halt = 0: Home position not reached<br>Halt = 1: Axle decelerates |
| | 1 | Halt = 0: Home position reached<br>Halt = 1: Axle has velocity 0 |
| Homing attained | 0 | Homing mode not yet completed |
| | 1 | Homing mode carried out successfully |
| Homing error | 0 | No homing error |
| | 1 | Homing error occurred;<br>Homing mode carried out not successfully;<br>The error cause is found by reading the error code |

## 7.2.3 Relevant parameter

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 607C $_h$ | VAR | home_offset | INT32 | RW |
| 6098 $_h$ | VAR | homing_method | INT8 | RW |
| 6099 $_h$ | ARRAY | homing_speeds | UINT32 | RW |
| 609A $_h$ | VAR | homing_acceleration | INT32 | RW |

**home_offset**

The object **home_offset** determines the displacement of the zero position to the limit resp. reference switch position.




| Index | 607C $_h$ |
|---|---|
| Name | home_offset |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | position units |
| Value Range | – |
| Default Value | 0 |

## homing_method

The negative and positive limit switch, the reference switch and the (periodic) zero impulse of the angle encoder.

| Index | 6098 h |
|---|---|
| Name | homing_method |
| Object Code | VAR |
| Data Type | INT8 |
| Access | RW |
| PDO Mapping | YES |
| Units | -- |
| Value Range | 1-14*，17-22，23-30*，33-35 |
| Default Value | 1 |

NOTE: * means only some servo models support this home mode.

Homing method value description is as following:

| Value | Direction | Target | Reference Point | DS402 |
|---|---|---|---|---|
| 1 | Negative | NOT | Zero impulse | 1 |
| 2 | Positive | POT | Zero impulse | 2 |
| 3 | Negative | Reference switch | Zero impulse | 3 |
| 4 | Positive | Reference switch | Zero impulse | 4 |
| 5 | Negative | Reference switch | Zero impulse | 5 |
| 6 | Positive | Reference switch | Zero impulse | 6 |
| 7 | Positive | Reference switch | Zero impulse | 7 |
| 8 | Positive | Reference switch | Zero impulse | 8 |
| 9 | Positive | Reference switch | Zero impulse | 9 |
| 10 | Positive | Reference switch | Zero impulse | 10 |
| 11 | Negative | Reference switch | Zero impulse | 11 |
| 12 | Negative | Reference switch | Zero impulse | 12 |
| 13 | Negative | Reference switch | Zero impulse | 13 |
| 14 | Negative | Reference switch | Zero impulse | 14 |
| 17 | Negative | NOT | NOT | 17 |
| 18 | Positive | POT | POT | 18 |
| 19 | Negative | Reference switch | Reference switch | 19 |
| 20 | Positive | Reference switch | Reference switch | 20 |
| 21 | Negative | Reference switch | Reference switch | 21 |
| 22 | Positive | Reference switch | Reference switch | 22 |
| 23 | Positive | Reference switch | Reference switch | 23 |
| 24 | Positive | Reference switch | Reference switch | 24 |

| Value | Direction | Target | Reference Point | DS402 |
|---|---|---|---|---|
| 25 | Positive | Reference switch | Reference switch | 25 |
| 26 | Positive | Reference switch | Reference switch | 26 |
| 27 | Negative | Reference switch | Reference switch | 27 |
| 28 | Negative | Reference switch | Reference switch | 28 |
| 29 | Negative | Reference switch | Reference switch | 29 |
| 30 | Negative | Reference switch | Reference switch | 30 |
| 33 | Negative | Current position | Zero impulse | 33 |
| 34 | Positive | Current position | Zero impulse | 34 |
| 35 | – | Current position | Current position | 35 |
| -4 | Positive | Target torque | Zero impulse | Reserved |
| -3 | Negative | Target torque | Zero impulse | Reserved |
| -2 | Positive | Target torque | Target torque | Reserved |
| -1 | Negative | Target torque | Target torque | Reserved |

## homing_speeds

There are two kinds of speeds required to find reference point, speed during search for switch and speed during search for zero.

| Index | 6099 h |
|---|---|
| Name | homing_speeds |
| Object Code | ARRAY |
| No. of Elements | 2 |
| Data Type | INT32 |

| Sub-Index | 01 h |
|---|---|
| Name | speed_during_search_for_switch |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 5000 |

| Sub-Index | 02 h |
|---|---|
| Name | speed_during_search_for_zero |

| Sub-Index | 02 h |
|---|---|
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 100 |

## Pn207 (stopper torque)

It is used for homing method -4, -3, -2, -1.When the drive hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, movement in the opposite direction or makes the current position for the origin.

| Index | $3049_h$ |
|---|---|
| Name | Pn207 (stopper torque) |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | 1% rated torque |
| Value Range | 0-200 |
| Default Value | 20 |

## Pn208 (blocking time)

It is used for homing method -4, -3, -2, -1.When the drive hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, movement in the opposite direction or makes the current position for the origin.

| Index | $304A_h$ |
|---|---|
| Name | Pn208 (Blocking time) |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | NO |
| Units | 0.125ms |
| Value Range | 0-10000 |
| Default Value | 100 |

**homing_acceleration**

The objects **homing_acceleration** determine the acceleration which is used for all acceleration and deceleration operations during the search for reference.

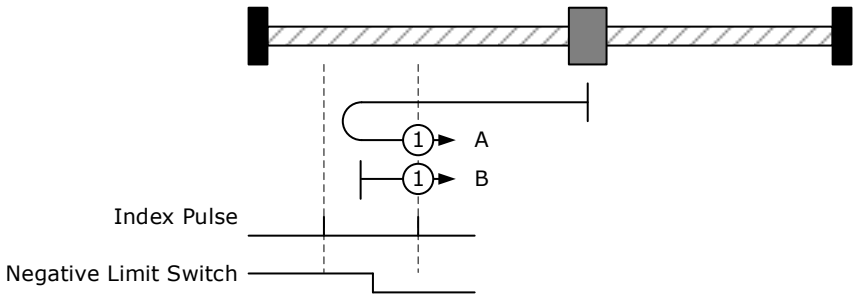
| Index | 609A $_{h}$ |
|---|---|
| Name | homing_acceleration |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | acceleration units |
| Value Range | – |
| Default Value | 100000 |

## 7.2.4 Homing Sequences

**Method 1: Using egative limit switch and zero impulse evaluation**

A: When homing mode is enabled, If negative limit switch N-OT=0, the drive first moves relatively quick into the negative direction until it reaches the negative limit switch. This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns, and stops until reaches the falling edge.

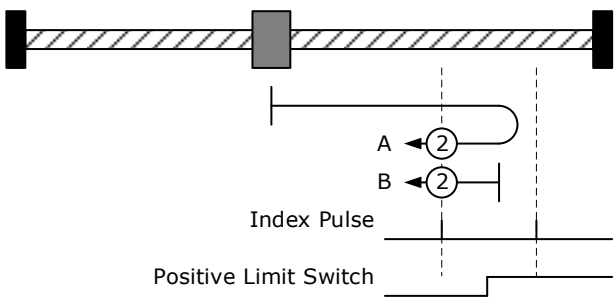
B: When homing mode is enabled, If negative limit switch N-OT=1, the drive first moves slowly into the positve direction until reaches the falling edge.




**Method 2: Using positive limit switch and zero impulse evaluation**

A: When homing mode is enabled, If positive limit switch P-OT=0, the drive first moves relatively quick into the positive direction until it reaches the positive limit switch. This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns, and stops until reaches the falling edge.

B: When homing mode is enabled, If positive limit switch P-OT=1, the drive first moves slowly into the negative direction until reaches the falling edge.



## Methods 3 and 4: Using positive reference switch and zero impulse evaluation

- Method 3

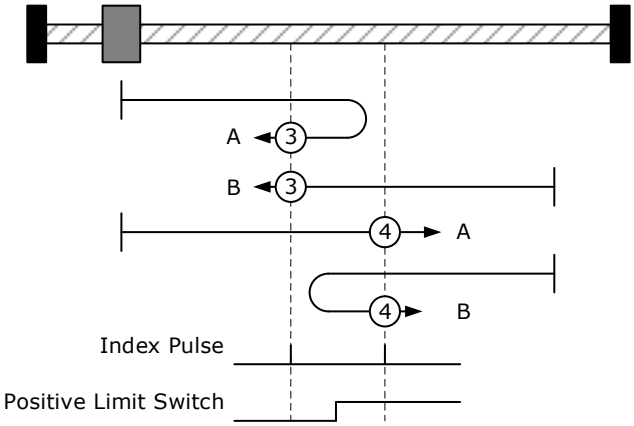
  A: When homing mode is enabled, If positive reference switch H-S=0, the drive first moves relatively quick into the positive direction until it reaches the positive reference switch. This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns, and stops until reaches the falling edge.

  B: When homing mode is enabled, If positive reference switch H-S =1, the drive first moves slowly into the negative direction until reaches the falling edge.

- Method 4

  A: When homing mode is enabled, If positive reference switch H-S =0, the drive first moves slowly into the positive direction until reaches the rising edge.

  B: When homing mode is enabled, If positive reference switch H-S=1, the drive first moves relatively quick into the negative direction until it reaches the positive reference switch. This is displayed in the diagram by the falling edge. Afterwards the drive slowly returns, and stops until reaches the rising edge.




## Methods 5 and 6: Using negative reference switch and zero impulse evaluation

- Method 5

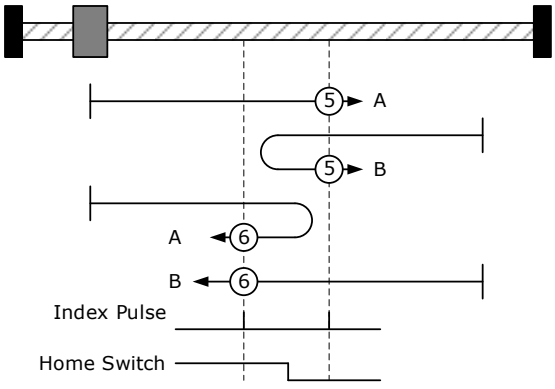
  A: When homing mode is enabled, If negative reference switch H-S =1, the drive first moves slowly into the positive direction until reaches the zero impulse evaluation. This is displayed in the diagram by the falling edge of H-S.

  B: B: When homing mode is enabled, If negative reference switch H-S=0, the drive first moves relatively quick into the negative direction until it reaches the negative reference switch This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns, and stops until reaches the zero impulse evaluation. This is displayed in the diagram by the falling edge of H-S.

- Method 6

  A: When homing mode is enabled, If negative reference switch H-S=1, the drive first moves relatively quick into the positive direction until it reaches the negative reference switch. This is displayed in the diagram by the falling edge. Afterwards the drive slowly returns, and stops until reaches the zero impulse evaluation. This is displayed in the diagram by the rising edge of H-S.

  B: When homing mode is enabled, If negative reference switch H-S =0, the drive first moves slowly into the positive direction, and stops until reaches the zero impulse evaluation. This is displayed in the diagram by the rising edge of H-S.



## Methods 7 ~ 14 Using reference switch , limit switch and zero impulse evaluation

Methods 7~14 use the reference switch which is only active over parts of the distance.

- Use positive limit switch P-OT.
  If this method 7~10 is used the drive first moves relatively quick into the positive direction
  - Method 7

    A: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,not reaches positive limit switch ,until it reaches the reference switch H-S. This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns, and stops until reaches the falling edge.

    B: When homing mode is enabled, If reference switch H-S =1, the drive first moves slowly into the negative direction until reaches the falling edge.

    C: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction , and reaches positive limit switch .The drive moves quickly into the negative direction.When reachinig the rising edge of H-S ,the drive moves slowly , and moves into the negative direction until reaches the falling edge of H-S.
  - Method 8

    A: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,not reaches positive limit switch , Afterwards the drive moves slowly into positive derection when reaches the rising edge of H-S, and stops until reaches the zero impulse evaluation.

    B: When homing mode is enabled, If reference switch H-S =1, the drive first moves slowly into the negative direction until reaches the falling edge of H-S. Then moves slowly into the positive direction, stops until reaches the zero impulse evaluation. This is displayed in the diagram by the H-S rising edge.

    C: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,reaches positive limit switch ,Afterwards the drive moves quickly into the negative derection,until reaches the rising edge of H-S. The drive slows down, and moves into into the negative derection.Reaches the falling edge of H-S,the drive returns into positive derection,until reaches the zero impulse evaluation.This is displayed in the diagram by the H-S rising edge.
  - Method 9

    A: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,not reaches positive limit switch. Afterwards the drive moves slowly into positive derection when reaches the rising edge of H-S. The drive slows down to stop until reaches the falling edge of H-S.Then drive returns slowly,and stops until reaches the zero impulse evaluation.This is displayed in the diagram by the H-S rising edge.

    B: When homing mode is enabled, If reference switch H-S =1, the drive first moves slowly into the postive direction until reaches the falling edge of H-S. Then moves slowly into the negative direction, stops until reaches the zero impulse evaluation. This is displayed in the diagram by the H-S rising edge.

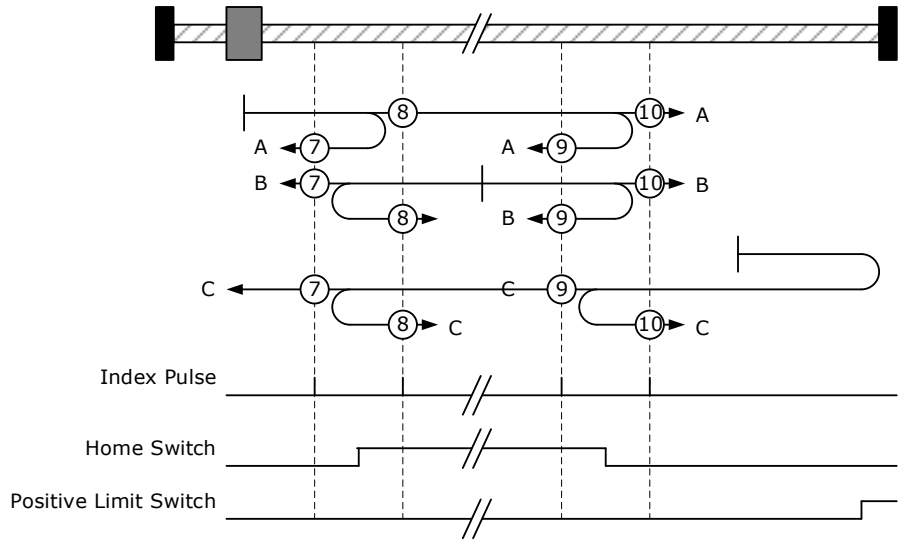
    C: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,reaches positive limit switch ,Afterwards the drive moves quickly into the negative derection,until reaches the rising edge of H-S. The drive slows down, and moves into into the negative derection, and stops until reaches the zero impulse evaluation.
  - Method 10

    A: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction, not reaches positive limit switch. Afterwards the drive moves slowly into positive derection when reaches the rising edge of H-S.If reaches the falling edge of H-S,the drive moves slowly into positive derection until reaches the zero impulse evaluation.

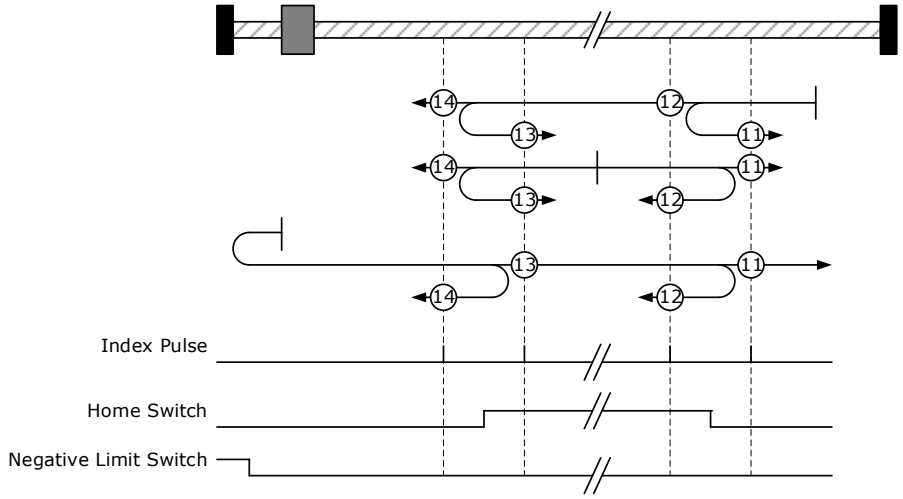
    B: When homing mode is enabled, If reference switch H-S =1, the drive first moves slowly into the positive direction until reaches the zero impulse evaluation. This is displayed in the diagram by the H-S falling edge.

    C: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the positive direction ,reaches positive limit switch ,Afterwards the drive moves quickly into the negative derection,until reaches the rising edge of H-S. The drive slows down to stop.Then the drive returns slowly to the positive derection, stops until reaches the zero impulse evaluation. This is displayed in the diagram by the H-S falling edge.

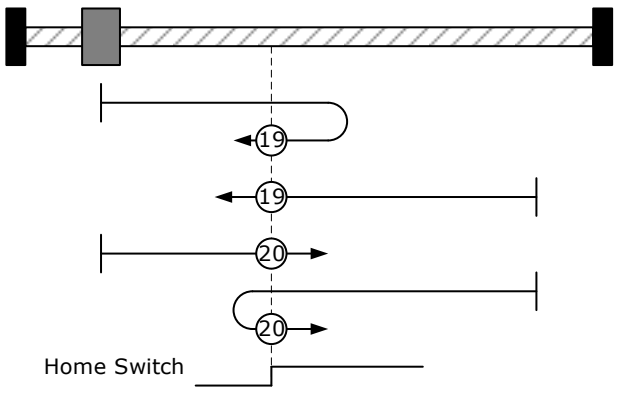



- Use negative limit switch
  If this method 11~14 is almost same as method 7~10, the drive first moves relatively quick into the negative direction.




## Method 17~20, 23~30: Homing operation to the negative limit switch

If this method is used the drive first moves relatively quick into the negative direction, until it reaches the negative limit switch. This is displayed in the diagram by the rising edge. Afterwards the drive slowly returns and searches for the exact position of the limit switch. The zero position refers to the descending edge from the negative limit switch.



## Methods 21, 22 Using reference switch

- Method 21

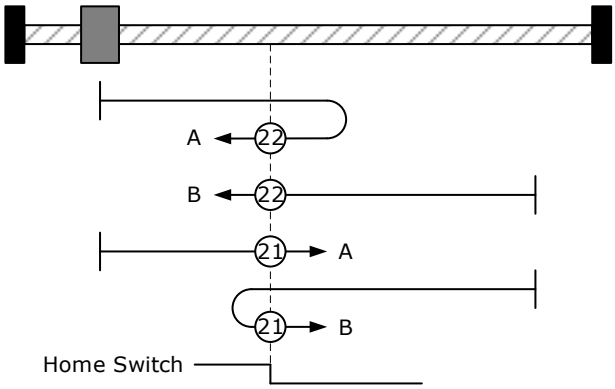
  A: When homing mode is enabled, If reference switch H-S =1, the drive first moves slowly into the positive direction until reaches the falling edge of H-S.

  B: When homing mode is enabled, If reference switch H-S=0, the drive first moves relatively quick into the negative direction until it reaches the reference switch. This is displayed in the diagram by the rising edge. Then the drive returns slowly to the positive derection, stops until reaches the falling edge of the H-S.

- Method 22

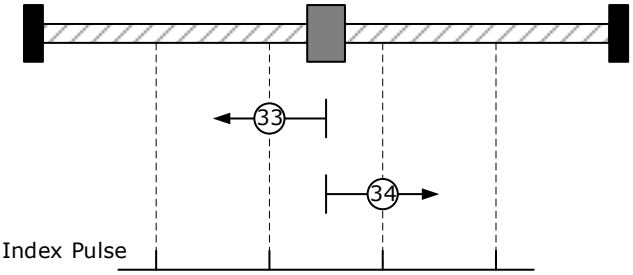
  A: When homing mode is enabled, If reference switch H-S=1, the drive first moves relatively quick into the positive direction until it reaches the reference switch. This is displayed in the diagram by the falling edge. Afterwards the drive slowly returns, and stops until reaches the rising edge of the H-S.

  B: When homing mode is enabled, If reference switch H-S =0, the drive first moves slowly into the negative direction until reaches the rising edge of the H-S.

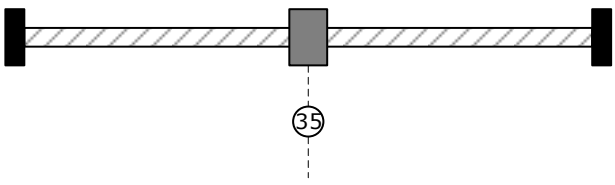



## Methods 33, 34 Using zero impulse evaluation

- Method 33: The drive moves slowly into the negative direction， stops until reaches the zero impulse evaluation.
- Method 34: The drive moves slowly into the positive direction， stops until reaches the zero impulse evaluation.



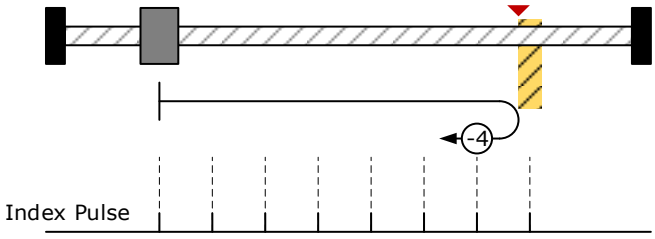

## Method 35: Set current position as the homing point




## Method -4 Movement in positive direction, hitting an end and reversing to travel，the target homing position is the first C pulse

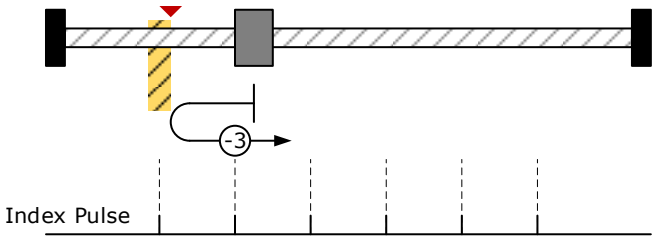
In this method, the motor moves in positive direction. When it hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, movement in the opposite direction, and the target homing position is the first C pulse.



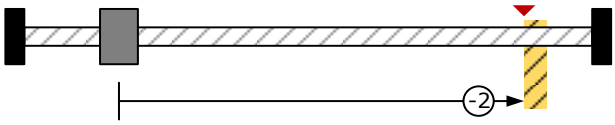

**Method -3 Movement in negative direction, hitting an end and reversing to travel，the target homing position is the first C pulse**

In this method, the motor moves in negative direction. When it hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, movement in the opposite direction, and the target homing position is the first C pulse.



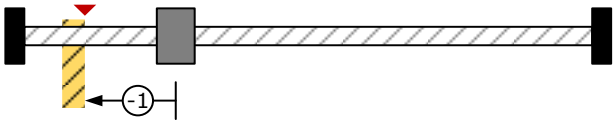

**Method -2:Movement in positive direction, hitting an end, makes the current position for the origin.**

In this method, the motor moves in positive direction. When the drive hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, and makes the current position for the origin.




**Method -1: Movement in negative direction, hitting an end, makes the current position for the origin.**

In this method, the motor moves in negative direction. When the drive hits an end so that the torque set in Pn207 is reached for the blocking time set in Pn208, and makes the current position for the origin.

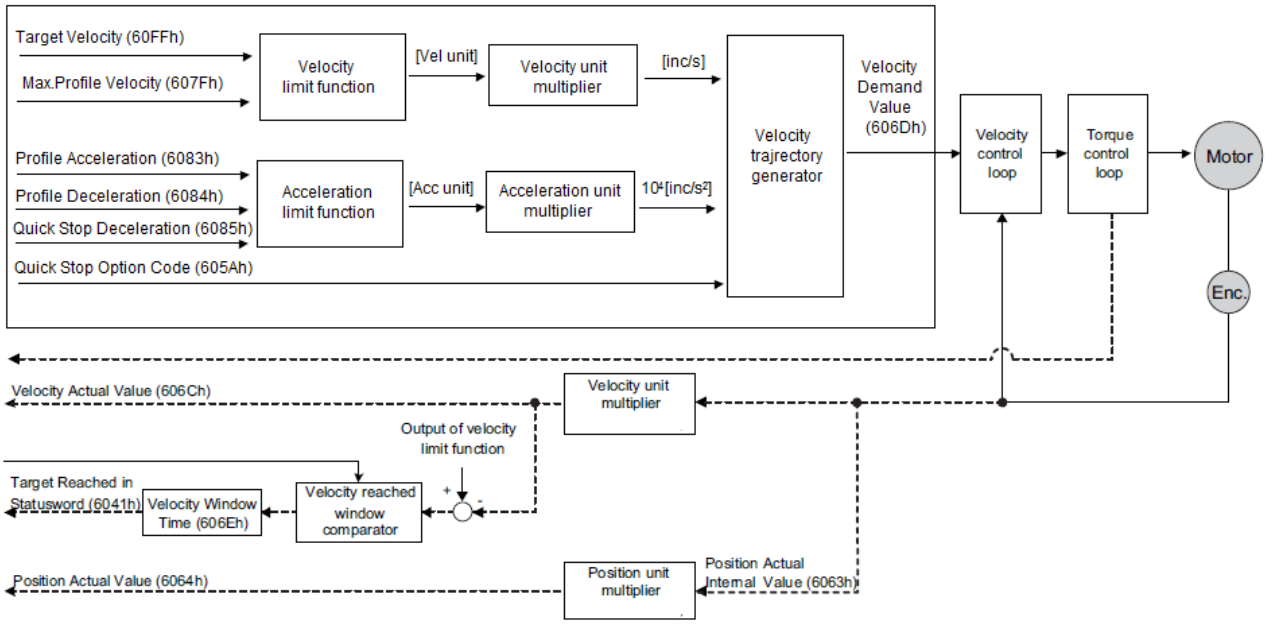



Notes: When starting homing on homing methed about input signal, the rotation direction of servo motor is associated with the initial status of the input signal.Changing the inicial status by inverse input on set Pn516/Pn517 if it is necessary.When using reference switch homing,I/O should be set as C:HmRef by Pn509/Pn510.

# 7.3 Profile Velocity Mode

## 7.3.1 Flow Diagram




## 7.3.2 Control word

| 15 ~ 9 | 8 | 7 ~ 4 | 3 ~ 0 |
|---|---|---|---|
| * | Halt | * | * |

*: referred to previous chapters

| Name | Value | Description |
|---|---|---|
| Halt | 0 | Execute the motion |
| | 1 | Stop axle |

## 7.3.3 Status word

| 15 ~ 14 | 13 | 12 | 11 | 10 | 9 ~ 0 |
|---|---|---|---|---|---|
| * | MaxSlippageError | Speed | * | Target reached | * |

*: referred to previous chapters

| Name | Value | Description |
|---|---|---|
| Target reached | 0 | Halt = 0: Target velocity not reached<br>Halt = 1: Axle decelerates |
| | 1 | Halt = 0: Target velocity reached<br>Halt = 1: Axle has velocity 0 |
| Speed | 0 | Speed is not equal 0 |
| | 1 | Speed is equal 0 |

| Name | Value | Description |
|---|---|---|
| Max slippage error | 0 | Maximum slippage not reached |
| | 1 | Maximum slippage reached |

## 7.3.4 Relevant Parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6069 h | VAR | velocity_sensor_actual_value | INT32 | RO |
| 606B h | VAR | velocity_demand_value | INT32 | RO |
| 606C h | VAR | velocity_actual_value | INT32 | RO |
| 606D h | VAR | velocity_window | UINT16 | RW |
| 606E h | VAR | velocity_window_time | UINT16 | RW |
| 606F h | VAR | velocity_threshold | UINT16 | RW |
| 6070 h | VAR | velocity_threshold_time | UINT16 | RW |
| 607F h | VAR | Max profile velocity | UINT32 | RW |
| 60FF h | VAR | target_velocity | INT32 | RW |

### velocity_sensor_actual_value

The speed encoder is read via the object **velocity_sensor_actual_value**. The value is normalised in internal units. The velocity demand value can be read via this object.

| Index | 6069 h |
|---|---|
| Name | velocity_sensor_actual_value |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | 0.1rmps (1R/10min) |
| Value Range | – |
| Default Value | – |

### velocity_demand_value

The velocity demand value can be read via this object. The unit of this object is the unit of user's speed unit. The velocity demand value can be read via this object.

| Index | 606B h |
|---|---|
| Name | velocity_demand_value |
| Object Code | VAR |
| Data Type | INT32 |

| Index | 606B h |
|---|---|
| Access | RO |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | – |

## velocity_actual_value

The actual velocity value can be read via the object **velocity_actual_value**. The velocity demand value can be read via this object.

| Index | 606C h |
|---|---|
| Name | velocity_actual_value |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RO |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | – |

## velocity_window

With the object **velocity_window** a tolerance window for the velocity actual value will be defined for comparing the **velocity_actual_value (**606C h**)** with the target velocity (**target_velocity** object **60FFh**). If the difference is smaller than the velocity window (606D h) for a longer time than specified by the object **velocity_window_time (**606E h**)** bit 10 (**target_reached)** will be set in the object **statusword**.

| Index | 606D h |
|---|---|
| Name | velocity_window |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 20 R/10min |

## velocity_window_time

The object **velocity_window_time** serves besides the object **606Dh**: **velocity_window** to adjust the window comparator.

| Index | 606E h |
|---|---|
| Name | velocity_window_time |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | ms |
| Value Range | – |
| Default Value | 0 |

## velocity_threshold

The object **velocity_threshold** determines the velocity underneath the axis is regarded as stationary. As soon as the velocity_actual_value exceeds the velocity_threshold longer than the velocity_threshold_time bit 12 is cleared in the statusword.

| Index | 606F h |
|---|---|
| Name | velocity_threshold |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 50 |

## velocity_threshold_time

The object **velocity_threshold** determines the velocity below the axis is regarded as stationary. Its unit is ms.As soon as the **velocity_actual_value** exceeds the **velocity_threshold** longer than the **velocity_threshold_time** bit 12 is cleared in the **statusword**.

| Index | 6070 h |
|---|---|
| Name | velocity_threshold_time |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |

| Index | 6070 h |
|---|---|
| Units | ms |
| Value Range | – |
| Default Value | 0 |

## Max profile velocity

The object max profile velocity is the speed that the motor can not exceed. Its unit is the unit of customer's speed.

| Index | 607F h |
|---|---|
| Name | Max profile velocity |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 0 |

## target_velocity

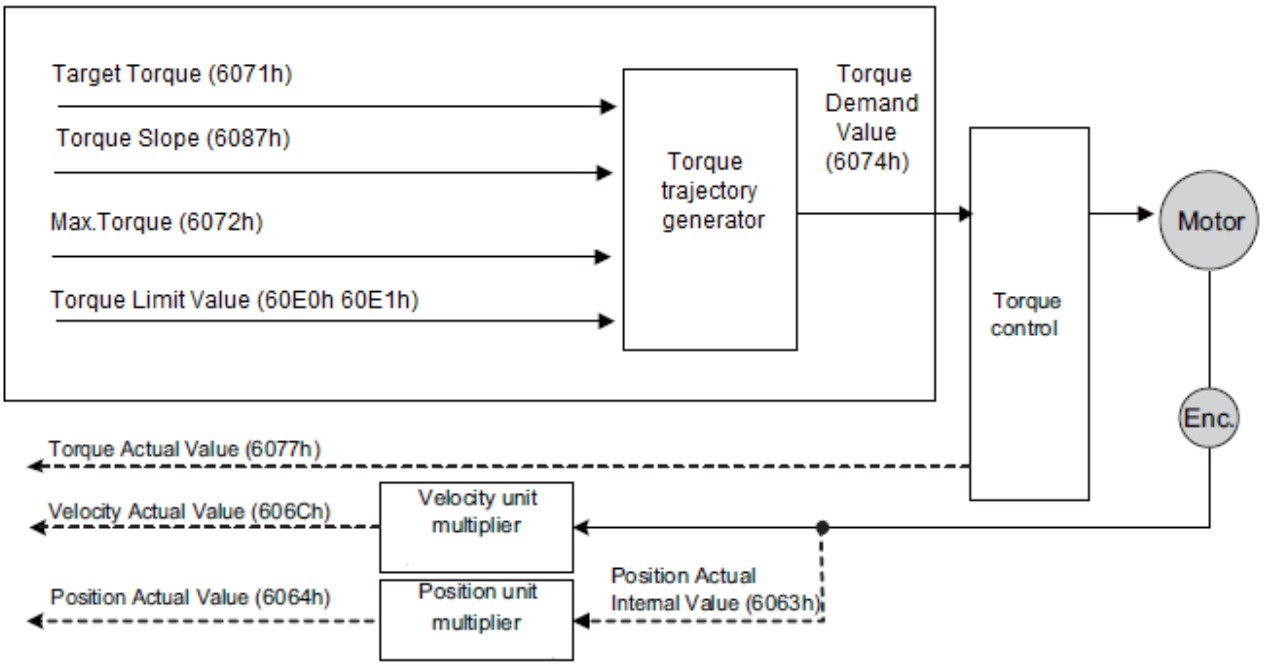
The object **target_velocity** is the setpoint for the ramp generator.

| Index | 60FF h |
|---|---|
| Name | target_velocity |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 0 |

# 7.4 Profile Torque Mode

## 7.4.1 Flow Diagram




## 7.4.2 Control Word

| 15 ~ 9 | 8 | 7 ~ 4 | 3 ~ 0 |
|---|---|---|---|
| * | Halt | * | * |

*: referred to previous chapters

| bit | Value | Definition |
|---|---|---|
| 8 | 0 | The motion shall be executed 8 or continued |
| | 1 | Axis shall be stopped according to the halt option code (605Dh) |

## 7.4.3 Status Word

| 15 ~ 14 | 13 | 12 | 11 | 10 | 9 ~ 0 |
|---|---|---|---|---|---|
| * | * | * | * | Target reached | * |

*: referred to previous chapters

| bit | Value | Definition |
|---|---|---|
| 10 | 0 | Target torque not reached |
| | 1 | Target torque reached |

## 7.4.4 Relevant Parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 6071 h | VAR | target_torque | INT16 | RW |
| 6072 h | VAR | Max torque | UINT16 | RW |
| 6074 h | VAR | torque_demand | INT16 | RO |
| 6077 h | VAR | torque_actual_value | INT16 | RO |
| 6087 h | VAR | torque_slope | UINT32 | RW |

**target_torque**

The torque command can be sent via target_torque. Unit: 0.1% rated torque.

| Index | 6071 h |
|---|---|
| Name | target_torque |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | 0.1% rated torque |
| Value Range | – |
| Default Value | – |

**Max torque**

The object max torque is the torque that the motor can not exceed. Unit: 0.1% rated torque.

| Index | 6072 h |
|---|---|
| Name | Max torque |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | 0.1% rated torque |
| Value Range | -- |
| Default Value | 0 |

**torque_demand**

The output of the torque command generator.The driver generates the command according to the value of Target_Torque and Torque_Slope.

| Index | 6074 h |
|---|---|
| Name | torque_demand |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RO |
| PDO Mapping | YES |
| Units | 0.1% rated torque |
| Value Range | – |
| Default Value | – |

## torque_actual_value

The torque output can be read via torque_actual_value. Unit: 0.1% rated torque.

| Index | 6077 h |
|---|---|
| Name | torque_actual_value |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RO |
| PDO Mapping | YES |
| Units | 0.1% rated torque |
| Value Range | – |
| Default Value | – |

## torque_slope

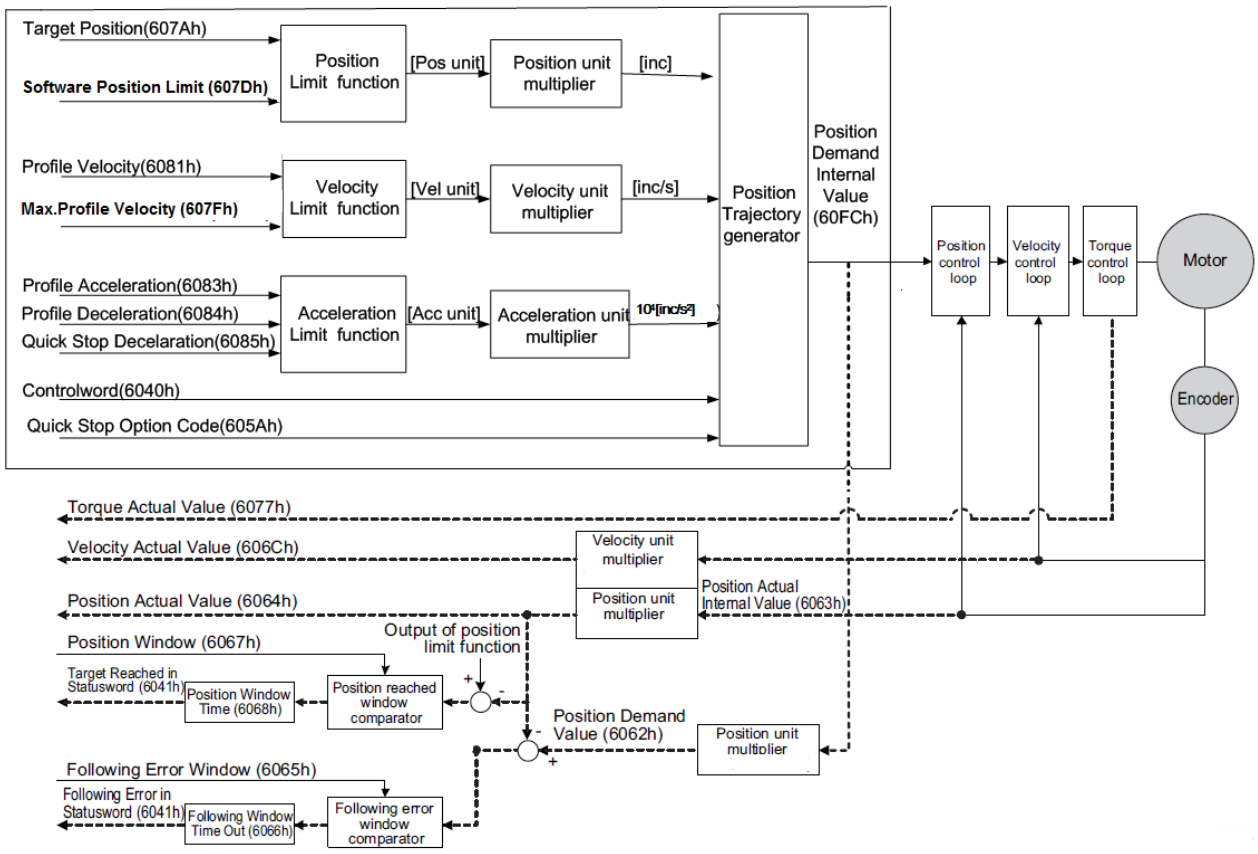
The speed of variational torque command can be set via torque_slope. Unit: 0.1% rated torque/S.

| Index | 6087 h |
|---|---|
| Name | torque_slope |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | 0.1% rated torque/S |
| Value Range | – |
| Default Value | – |

# 7.5 Profile Position Mode

## 7.5.1 Flow diagram




## 7.5.2 Control Word

| 15 ~ 9 | 8 | 7 | 6 | 5 | 4 | 3 ~ 0 |
|---|---|---|---|---|---|---|
| * | Halt | * | abs / rel | change set immediately | New set-point | * |

*: referred to previous chapter

| Name | Value | Description |
|---|---|---|
| New set-point | 0 | Does not assume target position |
| | 1 | Assume target position |
| Change set immediately | 0 | Finish the actual positioning and then start the next positioning |
| | 1 | Interrupt the actual positioning and start the next positioning |
| abs/rel | 0 | Target position is an absolute value |
| | 1 | Target position is a relative value |
| Halt | 0 | Execute positioning |
| | 1 | Stop axle with profile deceleration (if not supported with profile acceleration) |

## 7.5.3 Status word

| 15 ~ 14 | 13 | 12 | 11 | 10 | 9 ~ 0 |
|---|---|---|---|---|---|
| * | Following error | Set_point acknowledge | * | Target reached | * |

*: referred to previous chapter

| Name | Value | Description |
|---|---|---|
| Target reached | 0 | Halt = 0: Target position not reached<br>Halt = 1: Axle decelerates |
| | 1 | Halt = 0: Target position reached<br>Halt = 1: Velocity of axle is 0 |
| Set-point acknowledge | 0 | Trajectory generator has not assumed the positioning values (yet) |
| | 1 | Trajectory generator has assumed the positioning values |
| Following error | 0 | No following error |
| | 1 | Following error |

## 7.5.4 Relevant Parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 607A h | VAR | target_position | INT32 | RW |
| 6081 h | VAR | profile_velocity | UINT32 | RW |
| 6082 h | VAR | end_velocity | UINT32 | RW |
| 6083 h | VAR | profile_acceleration | UINT32 | RW |
| 6084 h | VAR | profile_deceleration | UINT32 | RW |
| 6085 h | VAR | quick_stop_deceleration | UINT32 | RW |
| 6086 h | VAR | motion_profile_type | INT16 | RW |
| 60A4-01 h | VAR | Profile_jerk1 | UINT32 | RW |

**target_position**

The object **target_position** determines the destination the servo controller moves to. The target position (**target_position**) is interpreted either as an absolute or relative position. This depends on bit 6 (**relative**) of the object **control word**.

| Index | 607A h |
|---|---|
| Name | target_ position |
| Object Code | VAR |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | position units |

| Index | 607A h |
|---|---|
| Value Range | – |
| Default Value | 0 |

## profile_velocity

The object **profile_velocity** specifies the speed that usually is reached during a positioning motion at the end of the acceleration ramp. The object **profile_velocity** is specified in **speed_units**.

| Index | 6081 h |
|---|---|
| Name | profile_velocity |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 0 |

## end_velocity

The object **end_velocity** defines the speed at the target position (**target_position**). Usually this object has to be set to zero so that the controller stops when it reaches the target position. For gapless sequences of positionings a value unequal zero can be set.

| Index | 6082 h |
|---|---|
| Name | end_velocity |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | speed units |
| Value Range | – |
| Default Value | 0 |

## profile_acceleration

The object **profile_acceleration** determines the maximum acceleration used during a positioning motion. It is specified in user specific acceleration units (**acceleration_units)**.

| Index | 6083 h |
|---|---|
| Name | profile_acceleration |

| Index | 6083 h |
|---|---|
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | acceleration units |
| Value Range | – |
| Default Value | 100000 R/10min/s |

## profile_deceleration

The object **profile_deceleration** specifies the maximum deceleration used during a positioning motion. This object is specified in the same units as the object **profile_acceleration**.

| Index | 6084 h |
|---|---|
| Name | profile_deceleration |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | acceleration units |
| Value Range | – |
| Default Value | 100000 R/10min/s |

## quick_stop_deceleration

The object **quick_stop_deceleration** determines the deceleration if a Quick Stop will be executed.

| Index | 6085 h |
|---|---|
| Name | quick_stop_deceleration |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | acceleration units |
| Value Range | – |
| Default Value | 200000 R/10min/s |

### motion_profile_type

The object **motion_profile_type** is used to select the kind of speed profile. At present only a linear trpezia profile (set as 0) and a stable S linear jerk profile are available (set as 2).

| Index | 6086 h |
|---|---|
| Name | motion_profile_type |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | YES |
| Units | – |
| Value Range | 0or2 |
| Default Value | 0 |

### profile_jerk1

Profile_jerk1 is used to set the jerk of speed profile. The value is more smaller,the speed changing is more smooth.

| Index | 60A4 -01h |
|---|---|
| Name | profile_jerk1 |
| Object Code | VAR |
| Data Type | UINT32 |
| Access | RW |
| PDO Mapping | YES |
| Units | jerk units |
| Value Range | 1-20 |
| Default Value | 5pulse/(s*100μs*100μs) |

## 7.5.5 Function description

When the speed profile is trapezia (motion_profile_type=0),two different ways to apply target positions to the servo controller are supported.

### Single-Step Set

After reaching the **target_position** the servo controller signals this status to the host by the bit **target_reached** (Bit 10 of **controlword**) and then receives a new setpoint. The servo controller stops at the **target_position** before starting a move to the next setpoint.

When moving to a setpoint, the servo controller signals a new setpoint by the bit **target_reached** (Bit 4 of **controlword**).Than the servo drive will move to the new setpoint.

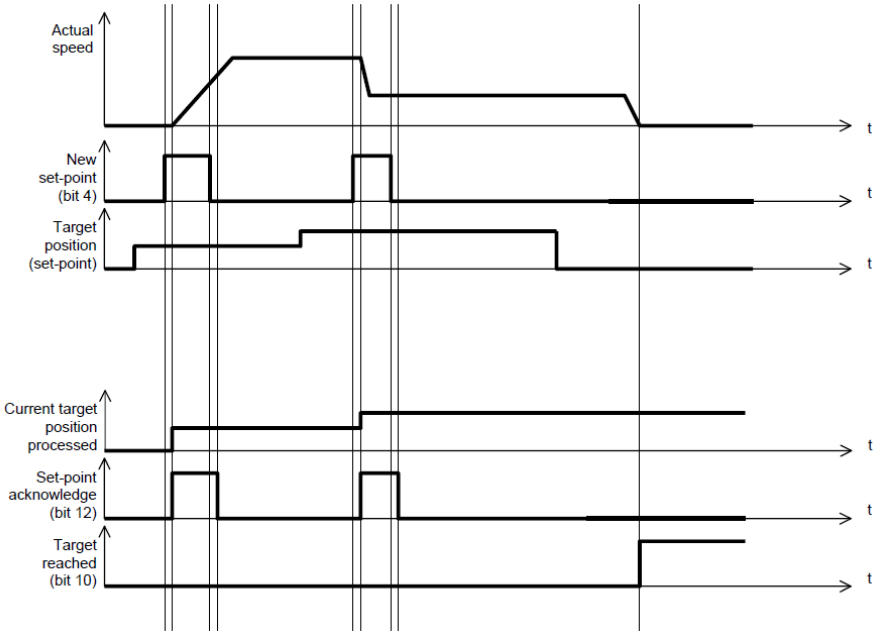
## Multi-Step Set

After reaching the **target_position** the servo controller signals this status to the host by the bit **target_reached** (Bit 10 of **controlword**) and then receives a new setpoint. The servo controller stops at the **target_position** before starting a move to the next setpoint.

These Two methods are controlled by the bit4 and **bit5** in the object **controlword** and **set_point_acknowledge** in the object **statusword**. These bits are in a request-response relationship. So it is possible to prepare one positioning job while another job is still running.

## Single-Step Setting Procedure

1. At first set NMT as Operational and control mode parameter (6061h) as 1.
2. At first the positioning data (**target_position:** 607A $_{h}$ **, profile_velocity, end_velocity** and **profile_acceleration**) are transferred to the servo controller.
3. The host can start the positioning motion by setting the bit4 (**new_set_point)** in the **controlword** as 1, bit5 (change_set_immediately) as 1 and bit6 as absolute or referential type according to target position type (absolute or referential).
4. This will be acknowledged by the servo controller by setting the bit **set_point_acknowledge** in the **statusword** when the positioning data has been copied into the internal buffer. Motion could be started now.
5. When the target is reached, drive will be acknowledged by bit 10 (target_reached) in status word. And then it will run gapless according to program or accept a new target position.

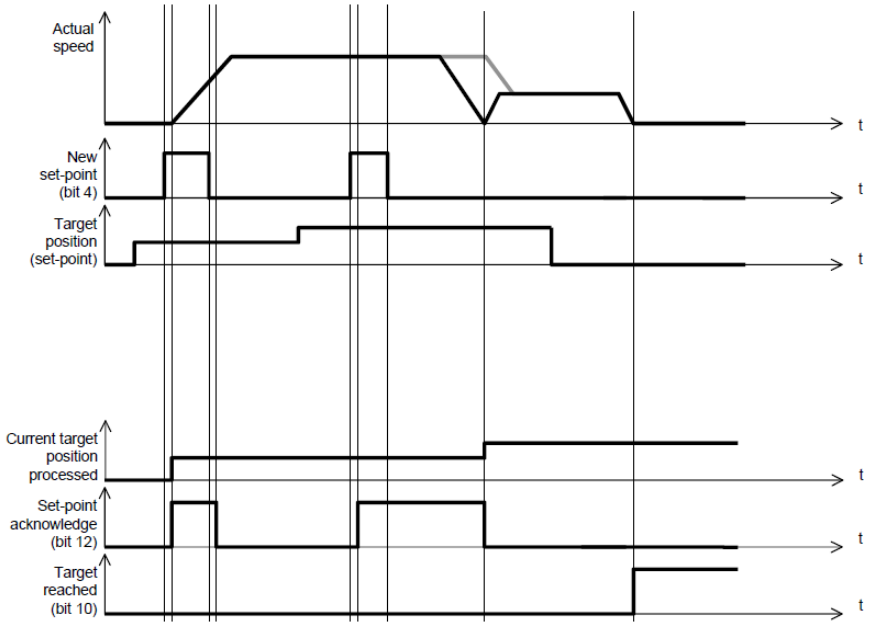



## Multi-Step Setting Procedure

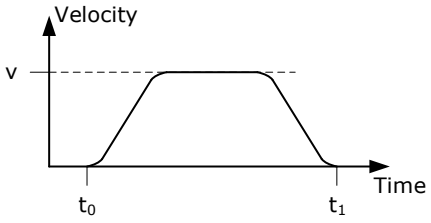
1. At first set NMT as Operational and control mode parameter (6061h) as 1.
2. At first the positioning data (**target_position:** 607A$_{h}$**, profile_velocity, end_velocity** and **profile_acceleration**) are transferred to the servo controller.
3. The host can start the positioning motion by setting the bit4 (**new_set_point)** in the **controlword** as 1, bit5 (change_set_immediately) as 0 and bit6 as absolute or referential type according to target position type (absolute or referential).
4. This will be acknowledged by the servo controller by setting the bit **set_point_acknowledge** in the **statusword** when the positioning data has been copied into the internal buffer. Motion could be started now.
5. Second positioning data (**target_position:** 607A$_{h}$**, profile_velocity, end_velocity** and **profile_acceleration**) are transferred to the servo controller.

6. The host can start the positioning motion by setting the bit4 (**new_set_point)** in the **controlword** as 1, bit5 (change_set_immediately) as 0 and bit6 as absolute or referential type according to target position type (absolute or referential).
7. When the 1 target is reached driver will move forward to second target position. When the second target position is reached drive will be acknowledged by bit10 (target reached) in status word. And then it will be executed by program or accept another new target position.



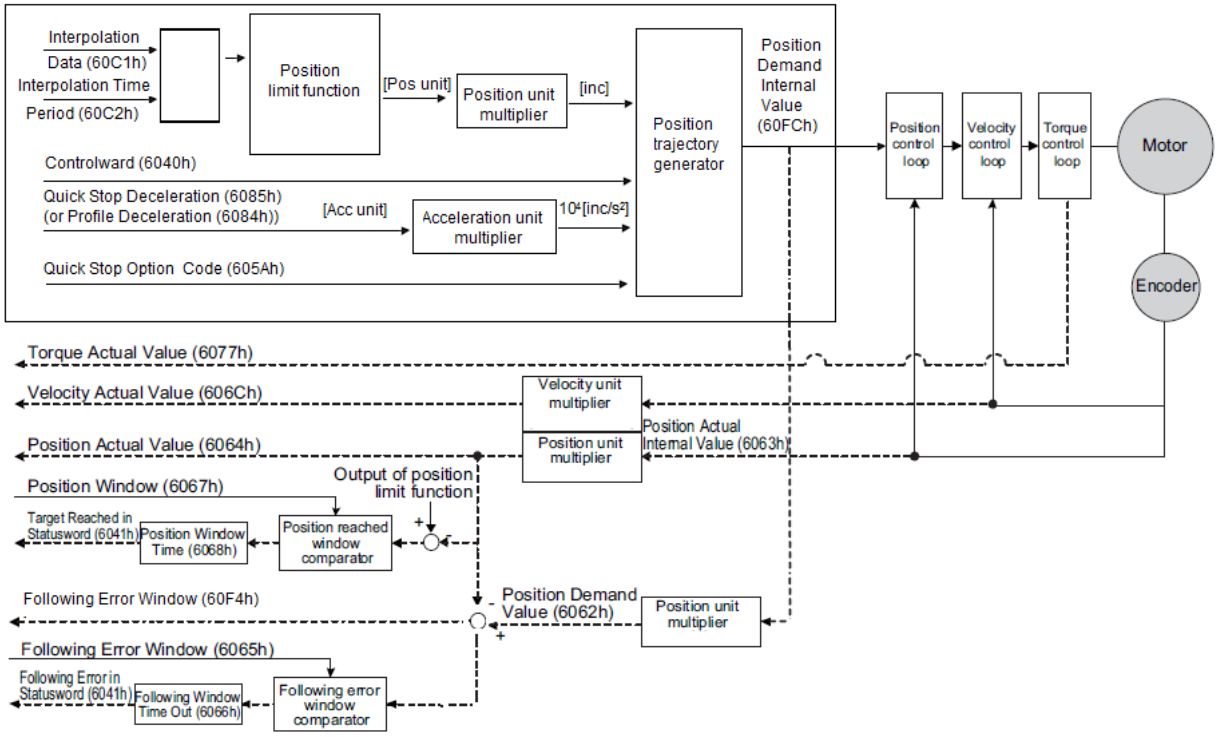

When the speed profile is S (motion_profile_type=2), only set of setpoints is available. 6083 h (profile_acceleration) limits max. acceleration. 6081h (profile_velocity) limits max.speed. 60A4-01 h (VAR Profile_jerk1) limits the jerk.now only symmetrical S linear is available.



# 7.6 Interplation Position Mode

## 7.6.1 Flow Diagram




## 7.6.2 Control Word

| 15 ~ 9 | 8 | 7 | 6 | 5 | 4 | 3 ~ 0 |
|---|---|---|---|---|---|---|
| * | Halt | * | * | * | Enable ip mode | * |

*: referred to previous chapter

| Name | Value | Description |
|---|---|---|
| Enable ip mode | 0 | Interpolated position mode inactive |
| | 1 | Interpolated position mode active |
| Halt | 0 | Execute the instruction of bit 4 |
| | 1 | Stop axle |

## 7.6.3 Status word

| 15 ~ 14 | 13 | 12 | 11 | 10 | 9 ~ 0 |
|---|---|---|---|---|---|
| * | * | ip mode active | * | Target reached | * |

*: referred to previous chapter

| Name | Value | Description |
|---|---|---|
| Target reached | 0 | Halt = 0: Target position not (yet) reached<br>Halt = 1: Axle decelerates |

| Name | Value | Description |
|---|---|---|
| | 1 | Halt = 0: Target position reached<br>Halt = 1: Velocity of axle is 0 |
| ip mode active | 0 | Interpolated position mode inactive |
| | 1 | Interpolated position mode active |

## 7.6.4 Relevant Parameters

| Index | Object | Name | Type | Attr. |
|---|---|---|---|---|
| 60C0 h | VAR | Interpolation sub mode select | INT16 | RW |
| 60C1 h | ARRAY | Interpolation data record | INT32 | RW |
| 60C2 h | RECORD | Interpolation time period | | RW |

### Interpolation sub mode select

Interpolation sub mode select is used to select the method of interpolation under IP control. Pronet servo drive only offers linear interpolation.

| Index | 60C0h |
|---|---|
| Name | Interpolation sub mode select |
| Object Code | VAR |
| Data Type | INT16 |
| Access | RW |
| PDO Mapping | NO |
| Value Range | 0 |
| Default Value | 0 |
| Comment | 0: Linear interpolation |

### Interpolation data record

Interpolation data record is used to reserve interpolation potion data. Our servo drive's interpolation command only uses the first data whose subindex is 1.

| Index | 60C1h |
|---|---|
| Subindex | 0 |
| Object Code | ARRAY |
| Data Type | INT32 |
| Access | RO |
| PDO Mapping | YES |
| Value Range | INT8 |
| Default Value | 2 |

| Index | 60C1h |
|---|---|
| Comment | number of entries |

| Index | 60C1h |
|---|---|
| Subindex | 1 |
| Object Code | ARRAY |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | INT32 |
| Default Value | 0 |
| Comment | the first parameter of ip function |

| Index | 60C1h |
|---|---|
| Subindex | 2 |
| Object Code | ARRAY |
| Data Type | INT32 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | INT32 |
| Default Value | 0 |
| Comment | The second parameter of ip function |

## Interpolation time period

Interpolation time period is used to reserve the time data of interpolation position.

| Index | 60C2h |
|---|---|
| Object Code | RECORD |
| Data Type | Interpolation time period record (0080h) |
| Category | Conditional: mandatory if ip, csp, csv or cst mode is supported |

| Index | 60C2h |
|---|---|
| Subindex | 0 |
| Object Code | RECORD |
| Data Type | UINT8 |
| Access | C |

| Index | 60C2h |
|---|---|
| PDO Mapping | NO |
| Value Range | 02 |
| Default Value | 02 |
| Comment | Highest sub-index supported |

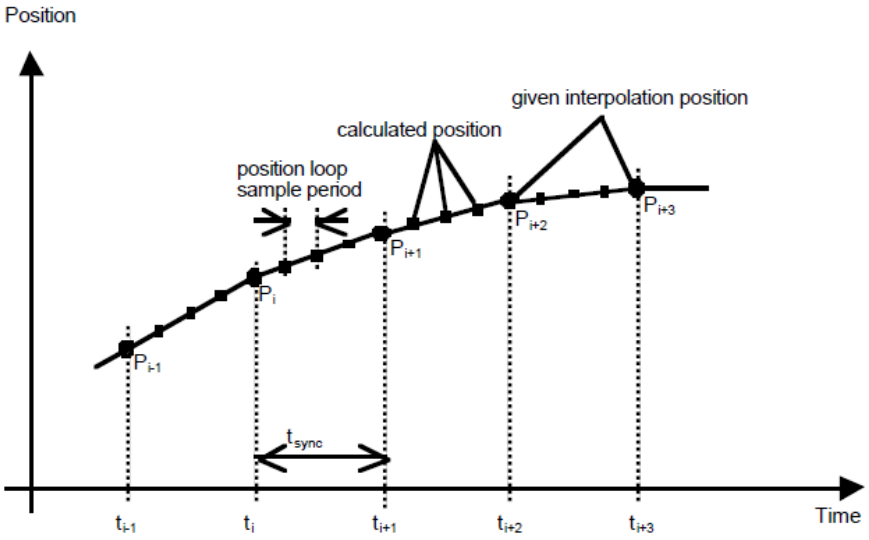
| Index | 60C2h |
|---|---|
| Subindex | 01 |
| Object Code | RECORD |
| Data Type | UINT8 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | UINT8 |
| Default Value | 01 |
| Comment | Interpolation time period value |

| Index | 60C2h |
|---|---|
| Subindex | 02 |
| Object Code | RECORD |
| Data Type | INT8 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | -128 to +63 |
| Default Value | -3 |
| Comment | Interpolation time index |

## 7.6.5 Function description

### Interpolation principle in IP mode




Pi: interpolation position set by the host

$t_{sync}$: sync period

**NOTE**

- In our servo drive, there is no buffer for position data so in IP control, all the position data needs to be updated by the controller. To achieve synchronization, controllers need to send the updated position at first and then use SYNC signal to make all the servo drive receive the synchronization information. After receiving the synchronization information, servo drive will synchronize its internal clock. Please notice that the sync period should be not bigger than interpolation cycle period in order to keep the updating of interpolation data.
- In IP mode, the host should at first set the servo's PDO receiving method into sync mode (Use SYNC frame to receive and send synchronization information). Because SYNC is broad casted, every servo drive will only update PDO data after receiving this signal.
- Before SYNC is sent, we need host to send position data Xi and control word to the servo drive.
- When there is data delay, servo drive will use the last sync date to do interpolation.
- After one sync period, if there is no further data updating, interpolation cycle overtime alarm (A 69) will happen. And then servo drive will stop.

### Recommended RPDO configuration

- When you use only one RPDO,
  - Control word(index:6040h,subindex:0h)
  - 32bit position reference (index:60C1h,subindex:01h)
- When you use two RPDO
  - Control word(index:6040h,subindex:0h)
  - 32bit position reference (index:60C1h,subindex:01h)

### Configuration process

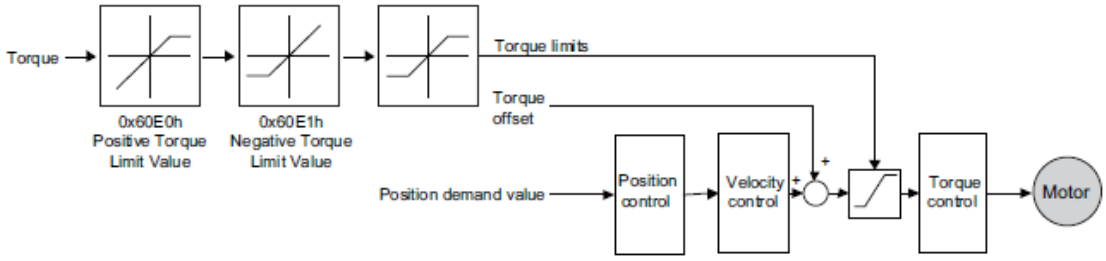
1. Configure PDO. (RPDO1 is configured as index: 6040h, subindex: 0h, RPDO2 is configured as index 60c1h, subindex: 1h)
2. Set interpolation cycle time 2105h and 60C2, the unit is micro send (us). Please notice that both values need to be configured. For example, if the cycle time is 2ms, you need to set 2105h as 2000 and 60c2:01 as 2, 60c2:02 as -3.
3. Set sync cycle time (1006h), the unit is micro send (us)

4. Set PDO as Sync mode (Set the object dictionary (index: 1400h, subindex: 02h) as 1. Set object dictionary (index: 1401h, subindex: 02h) as 1). If sending PDO needs to be in sync mode as well, we need to set object dictionary (index: 1800h, subindex: 02h) as 1 and (index: 6060h, subindex: 0h) as 1 as well.
5. NMT starts node.

## 7.7 Torque limit Function

In CANopen bus mode, torque limit function is realized by 0x60E0 snd 0x60E1 as below.




**PosTorLimit(0x60E0)**

PosTorLimit (0x60E0): positive torque limit, unit: 0.1% rated torque

| Index | 60E0h |
|---|---|
| Name | PosTorLimit |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | 0-3000 |
| Default Value | 3000 |

**NegTorLimit(0x60E1)**

NegTorLimit(0x60E1): negative torque limit,unit: 0.1% rated torque

| Index | 60E1h |
|---|---|
| Name | NegTorLimit |
| Object Code | VAR |
| Data Type | UINT16 |
| Access | RW |
| PDO Mapping | YES |
| Value Range | 0-3000 |
| Default Value | 3000 |

# Chapter 8 CANopen Configuration Example

The entire test below is based on three conditions:

- Communication has been established correctly.
- The address of the servo drive is 1.
- All the message data is hexadecimal.

## 8.1 SDO configuration

SDO operation is to read and write parameters (0601h → host sends 0581h →slave sends)

Address: 3022h (Pn118) . Write 1000. And then read this parameter.
Activate the downloading process: 2B，3022，00，03E8

- Message: 601(ID) 2B 22 30 00 E8 03 00 00

  The servo drive should respond 60，3022，00，00，00，00，00
- Message: 581(ID) 60 22 30 00 00 00 00 00

  Activate the uploading: 40，3022，00，0000
- Message: 601(ID) 40 22 30 00 00 00 00 00

  The servo drive needs to respond: 43，3022，00，03E8
- Message: 581(ID) 43 22 30 00 E8 03 00 00

## 8.2 PDO Configuration

RPDO mapping example: To configure two RPDO, one of which is 6040h and the other are 607A and 6081h). The slaves respond of message 581h (ID) is omitted.

- RPDO 1 Mapping

```
601 (ID)2F 00 16 00 00 00 00 00 //RPDO1 stop
First RPDO 201
601 (ID)23 00 16 01 10 00 40 60 //6040h
601(ID) 2F 00 16 00 01 00 00 00 // RPDO1 enable

601(ID) 2F 01 16 00 00 00 00 00 //RPDO2 stop
```

- RPDO 2 Mapping

```
601(ID) 23 01 16 01 20 00 7A 60 //607Ah and 6081h
601 (ID)23 01 16 02 20 00 81 60
601(ID) 2F 01 16 00 02 00 00 00// RPDO2 enable
```

And then set the transmit PDO as SYNC or Timing method. The default setting is Time method.

After configuring the PDO, if you need to activate the configuration, you need to reset the communication.

```
NMT is OPERATIONAL: 00 01 01// (the first “01” is the start node instruction,the
second “01” is the number of the node)
```

## 8.3 Profile Positon Mode

**Step 1** At first, please mapping and configure PDO according to the example above and activate the communication.

**Step 2** And then, please set the control mode.

```
message: 601(ID) 2F 60 60 00 01 00 00 00//set 6060h as 1 (position contrl is PP)
```

**Step 3** And then, set status machine as Operation Enable

```
message: 601(ID) 2B 40 60 00 06 00 00 00//set 6040h as 6,switch to "ready to switch
on"
message: 601(ID) 2B 40 60 00 07 00 00 00 //set 6040h as 7, switch to "switch on"
message: 601(ID) 2B 40 60 00 0F 00 00 00 //set 6040h as F, switch to "operation
enable" and servo-on;
```

This switching flow is based on successful switching received command and original state is "switch on disabled".

**Step 4** And then, send data by PDO

Let servo motor rotate for 5 revolutions (Set PDO1 as 6040(status word), PDO2 as 607A (position pulse number) and 6081(velocity, unit as much as 0.1rpm)

**Step 5** Send RPDO2 The data is as below

```
message: 301 (ID) 50 C3 00 00 2C 01 00 00(50000,300)// 50 C3 00 00 is position data,
that is, 50000 pulses; 2C 01 00 00 is speed, that is, 30rpm;
```

**Step 6** Send RPDO1 as below

```
message: 201(ID) 0F 00 //; Clear the bit4 of 6040 as 0.
message: 201(ID) 1F 00 // Clear the bit4 of 6040 as 1 and servo motor is operating
under absolute position; Motor runs.
message: 201(ID) 0F 00 //Clear the bit4 of 6040.
message: 201(ID) 5F 00 // Clear the bit4 of 6040 as 1. The servo motor runs under
incremental position.
message: 201(ID) 0F 00 //Clear bit4 of 6040 as 0.
```

- The servo drive is using ↑of 6040's bit 4 to accept new position order. So after every single operation, the bit needs to be cleared. Host needs to check bit12 of status word 6040 in the servo drive to decide whether or not to give new data to servo systems. When status word 6041 in the servo drives 0, it means the servo drive is ready for new data and order. If the value is 1, the order won't be executed even if there is data for the servo drive to receive.
- In absolute approach, continuous position updating is required.
- If you want to change the operating distance, you need to send RPDO2 again.

**----End**

# 8.4 Two-axis interplate position mode

**Step 1** At first, mapping and configure PDO.

```
// receive 2 PDO by default: RPDO1: 60C1h--01h
// Send two PDO by default: TPDO1: 6041h  TPDO2: 6064h/606Ch
// pulse, Velocity 0.1rpm
```

**Step 2** Configure 1 RPDO

```
RPDO MAPPing
message: 601(ID)  2F 00 16 00 00 00 00 00   //RPDO1 stop
message: 601(ID)  23 00 16 01 20 01 C1 60  //60C1h,sub01
message: 601(ID)  2F 00 16 00 01 00 00 00// RPDO1 enable
```

**Step 3** Configure 2 TPDO，TPDO1: 6041h TPDO2: 6064h/606Ch

```
TPDO MAPPing
message: 601(ID)  2F 00 1A 00 00 00 00 00  //TPDO1 stop
```

```
message: 601(ID)  23 00 1A 01 10 00 41 60  //6041h
message: 601(ID)  2F 00 1A 00 01 00 00 00 // TPDO1 enable

message: 601(ID)  2F 01 1A 00 00 00 00 00   //RPDO2 stop
message: 601(ID)  23 01 1A 01 20 00 64 60  //6064h  and   606Ch
message: 601(ID)  23 01 1A 02 20 00 6C 60  //
message: 601(ID)  2F 01 1A 00 02 00 00 00// TPDO2 enable
```

**Step 4** Set Sync time.

```
message: 601(ID)  2F C2 60 01 10 00 00 00  //60C2h-01----->1ms, set according to the
actual needs
message: 601(ID)  2F C2 60 02 FD 00 00 00  //
```

**Step 5** Configure the PDO receiving and sending are both activated by one Sync frame.

- Set 1400h

```
message: 601(ID)  2F 00 14 02 01 00 00 00  //1400-02---->SYNC
```

- Set 1800h

```
message: 601(ID)  2F 00 18 02 01 00 00 00  //1800-02---->SYNC
```

- Set 1801h

```
message: 601(ID)  2F 01 18 02 01 00 00 00  //1801-02---->SYNC
```

**Step 6** Set control mode

```
message: 601(ID)  2F 60 60 00 07 00 00 00//Set 6060h as7 ( IP position control)
```

**Step 7** And then, set the status machine

```
message: 601(ID)  2B 40 60 00 06 00 00 00// Set 6040h as 6
message: 601(ID)  2B 40 60 00 07 00 00 00  // Set 6040h as 7
message: 601(ID)  2B 40 60 00 0F 00 00 00  // Set 6040h as F to servo on

message: 601(ID)  2B 40 60 00 IF 00 00 00  // Set 6040h as1F to IP_ACTIVE status;
```

**Step 8** Activate the communicaiton

```
message: 00(ID) 01 01
```

**Step 9** Setting the second axis is the same with the first axis.

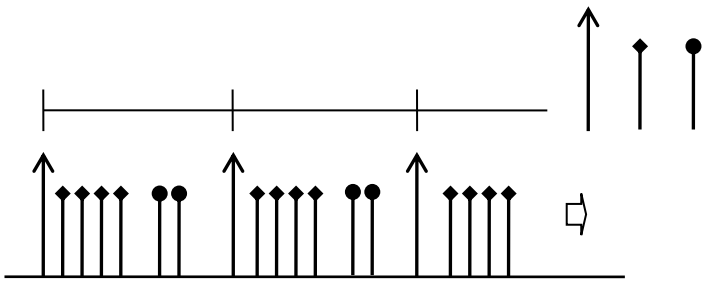
**Step 10** Host sends according to the setting of test SYNC cycle(1ms):

```
message: 201(ID)  10 00 00 00 //16 P/R, interpolation potion data of the first axis
message: 202(ID)  20 00 00 00 //32 P/R, interpolation potion data of the second axis
message: 80(ID) // cycle to send SYNC according to interpolation cycle
```

**Step 11** And then,drive returns

```
message: 181(ID)  xx xx 00 00 //status word of the first axis
message: 182(ID)  xx xx xx xx     //position and speed of the first axis
message: 281(ID)  xx xx 00 00 //status word of the second axis
message: 282(ID)  xx xx xx xx     //position and speed of the second axis
```

Sequence diagram is as following



**----End**

# 8.5 Homing

**Step 1** Set the control mode as homing control.

```
message: 601(ID) 2F 60 60 00 06 00 00 00// Set the control mode as homing control.

message: 601(ID) 2F 98 60 00 04 00 00 00//Use the fourth way to set the homing mode.
```

**Step 2** Set the status machine

```
message: 601(ID) 2B 40 60 00 06 00 00 00
message: 601(ID) 2B 40 60 00 07 00 00 00
message: 601(ID) 2B 40 60 00 0F 00 00 00 //Servo On
```

**Step 3** Send data through PDO. (Set PDO1 as 6040(status word). Set PDO2 as 607A (Position pulse number) and 6081. (Speed, unit 0.1rpm)
Set the homing method as 10rpm.

```
message: 601(ID) 23 99 60 02 64 00 00 00
```

**Step 4** Homing is started.

```
message: 201 (ID) 1F 00
```

Cancel homing

```
Message: 201(ID) 0F 00
```

**----End**

# Chapter 9 MODBUS Communication

## 9.1 RS-485 Wiring

ED3M model Servodrives provide the MODBUS communication function with RS-485 interface, which can be used to easily set parameters or to perform monitoring operations and so on.

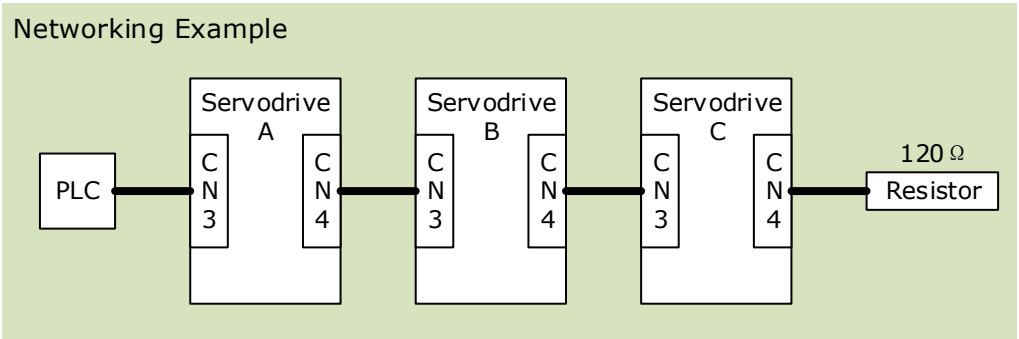
The definitions of the Servodrive communication connector terminals (CN3 and CN4) are as follows.

| Terminal No | Name | Description |
|---|---|---|
| 1 | – | Reserved |
| 2 | – | Reserved |
| 3 | 485+ | Terminal for RS-485 (+) |
| 4 | ISO_GND | Isolated ground |
| 5 | ISO_GND | Isolated ground |
| 6 | 485- | Terminal for RS-485 (-) |
| 7 | CANH | Terminal for CAN (high-bits) |
| 8 | CANL | Terminal for CAN (low-bits) |

NOTE

- The cable length shall be less than 100 meters in the environment of less interference. However, if the transmission speed needs to be 9600 bps or more, please use the cable within 15meters.
- Up to 31 Servodrives can be connected at the same time by using RS485, and a 120Ω of resistor needs to be connected to each terminal end. For connecting more devices into the network, please use a repeater to expand the number of connections.
- Always regard CN3 of Servodrive as the input terminal, and regard CN4 of Servodrive as the output terminal. That is, connect CN4 with CN3 of another Servodrive for expanding one slave station, and add a balance resistor to CN4 of the last Servodrive.
  It is strictly forbidden to connect CN3 of any two Servodrives.

For example, a network consists of one PLC and three of Servodrives A, B and C, the connection of them is shown in following figure.



# 9.2 Relevant Parameters for MODBUS

| No. | Description | When Enabled | Related Control | | |
|---|---|---|---|---|---|
| Pn700.0 | Communication rate selection:<br>[0] 4800bps<br>[1] 9600bps<br>[2] 19200bps<br>[3] 38400bps<br>[4] 57600bps<br>[5] 115200bps | After restart | P | S | T |
| Pn700.1 | Communication mode selection<br>[0] 7, N, 2 (Modbus,ASCII)<br>[1] 7, E, 1 (Modbus,ASCII)<br>[2] 7, O, 1 (Modbus,ASCII)<br>[3] 8, N, 2 (Modbus,ASCII)<br>[4] 8, E, 1 (Modbus,ASCII)<br>[5] 8, O, 1 (Modbus,ASCII)<br>[6] 8, N, 2 (Modbus,RTU)<br>[7] 8, E, 1 (Modbus,RTU)<br>[8] 8, O, 1 (Modbus,RTU) | After restart | P | S | T |
| Pn700.2 | Reserved | – | – | – | – |
| Pn700.3 | Reserved | – | – | – | – |
| Pn701 | Axis address in MODBUS | After restart | P | S | T |

# 9.3 MODBUS Protocol

There are two modes for MODBUS communication: ASCII (American Standard Code for information interchange) mode and RTU (Remote Terminal Unit) mode.

## 9.3.1 Code Meaning

**ASCII Mode**

In Modbus ASCII, each data byte is split into the two bytes, representing the two ASCII characters in the Hexadecimal value. For example, the ASCII value '64' is represented as **3634** in hex, in which, ASCII value '6' is represented as $36_{(H)}$, and '4' is represented as $34_{(H)}$.

The range of data bytes in Modbus ASCII represent only the 16 hexadecimal characters. Therefore, every data byte in Modbus ASCII must be one of these 16.

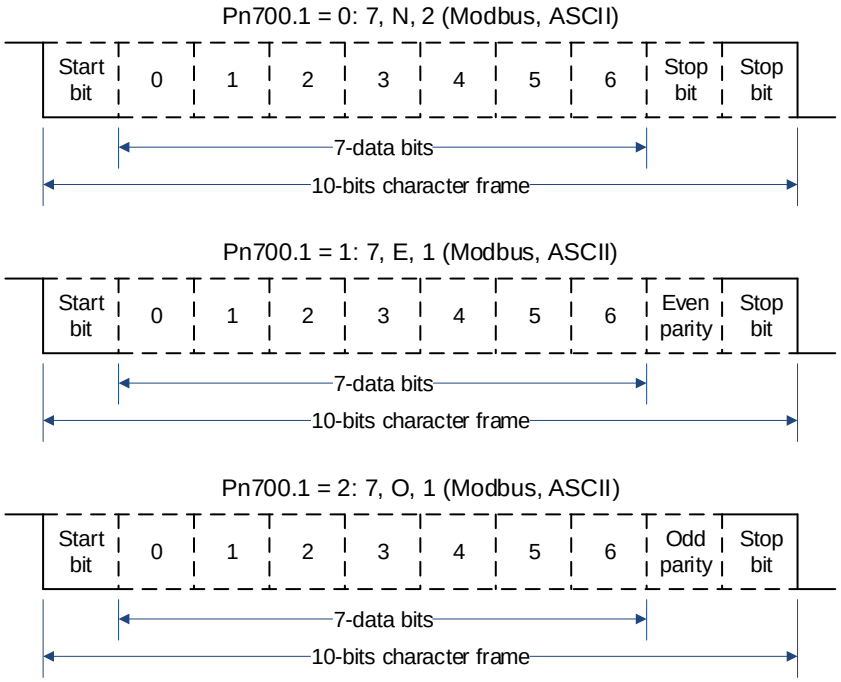
| ASCII | '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' |
|---|---|---|---|---|---|---|---|---|
| Hex | $30_{(H)}$ | $31_{(H)}$ | $32_{(H)}$ | $33_{(H)}$ | $34_{(H)}$ | $35_{(H)}$ | $36_{(H)}$ | $37_{(H)}$ |
| ASCII | '8' | '9' | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' |
| Hex | $38_{(H)}$ | $39_{(H)}$ | $41_{(H)}$ | $42_{(H)}$ | $43_{(H)}$ | $44_{(H)}$ | $45_{(H)}$ | $46_{(H)}$ |

**RTU Mode**

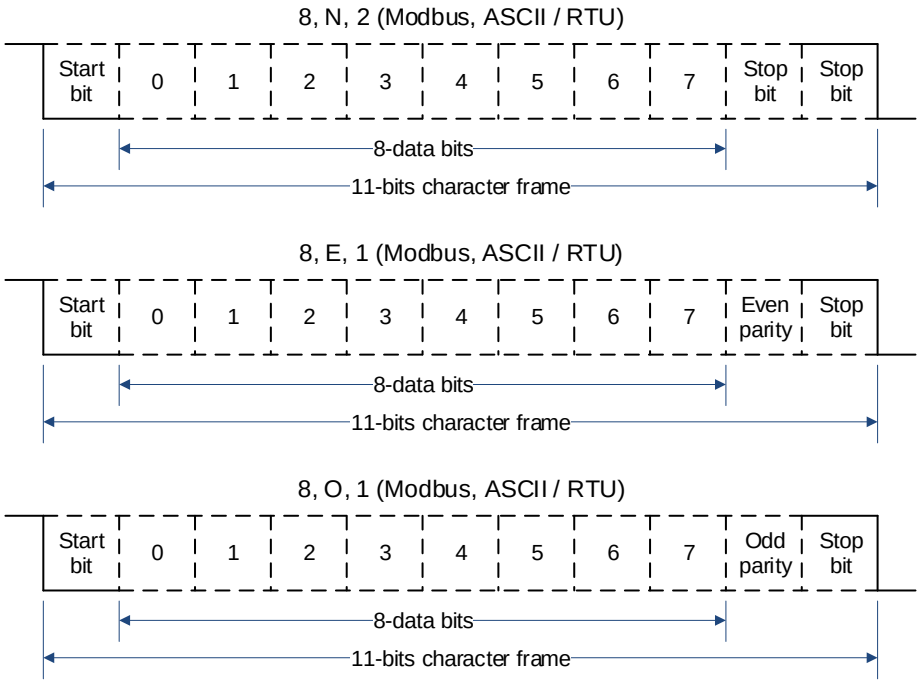
In Modbus RTU, each data byte (8 bit) consists of two 4 bits hexadecimal value. For example, the decimal data 100 can be represented as $64_{(H)}$ in RTU mode.

- Data Structure: 10 bits character form (for 7 bits data)




- Data Structure: 11 bits character form (for 8 bits data)



## 9.3.2 Data Format

### ASCII Mode

| | |
|---|---|
| **STX** | Starting character, which is from ‘: ’ to ($3A_H$) |
| **ADR** | Communication address: 2 ASCII characters. |
| **CMD** | Command code: 2 ASCII characters. |
| **DATA(n-1)** | Data content: n word = 2n bytes = 4n ASCII characters, n is not greater than 12 |
| **……** | |
| **DATA(0)** | |
| **LRC** | Check code: 2 ASCII characters. |
| **End 1** | End code 1: ($0D_H$) (CR) |
| **End 0** | End code 1: ($0A_H$) (LF) |

### RTU Mode

| | |
|---|---|
| **STX** | A static period for transferring 4 bytes at least. |
| **ADR** | Communication address: 1 byte |
| **CMD** | Command code: 1 byte |
| **DATA(n-1)** | Data content: n word = 2n bytes, n is not greater than 12 |
| **……** | |
| **DATA(0)** | |
| **CRC** | Check code: 1 byte |
| **End 1** | A static period for transferring 4 bytes at least. |

### Format Description

- STX (communication starting)
  - ASCII mode: character ‘: ’
  - RTU mode: A static period for transferring 4 bytes at least.
- ADR (communication address)

  A valid communication address is the range from 1 to 254.

  For example, communicate with the Servodrive which address is 32 (20 in hex):
  - In ASCII mode: ADR=‘2’, ‘0’ → ‘2’=$32_H$, ‘0’=$30_H$
  - In RTU mode: ADR=$20_H$
- CMD (command code) and DATA

  The format of the data depends on the command code. The common command code is shown as following:

  Command code: $03_H$, read N words, and N is not greater than 20.

  For example, read 2 words starting from $0070_H$ from the servo drive which address is 01 H.

| **ASCII Mode** | | | |
|---|---|---|---|
| **Request information** | | **Response information** | |
| STX | ": " | STX | ": " |
| ADR | '0' | ADR | '0' |
| | '1' | | '1' |
| CMD | '0' | CMD | '0' |
| | '3' | | '3' |
| Data started address | '0' | Number of Date (calculated by byte) | '0' |
| | '0' | | '4' |
| | '7' | Data content of started address 0200 H | '0' |
| | '0' | | '0' |
| Number of Date (calculated by word) | '0' | | '0' |
| | '0' | | '0' |
| | '0' | Data content of address 0201 H | '0' |
| | '2' | | '0' |
| LRC Check | '8' | | '0' |
| | 'A' | | '0' |
| End 1 | (0D H)(CR) | LRC Check | 'F' |
| End 0 | (0A H)(LF) | | '8' |
| | | End 1 | (0D H )(CR) |
| | | End 0 | (0A H )(LF) |

| **RTU Mode** | | | |
|---|---|---|---|
| **Request information** | | **Response information** | |
| ADR | 01 H | ADR | 01 H |
| CMD | 03 H | CMD | 03 H |
| Data started address | 00 H (high-bits) | Number of Date (calculated by byte) | 04 H |
| | 70 H (low-bits) | Data content of started address 0200 H | 00 H (high-bits) |
| Number of Date (calculated by word) | 00 H | | 00 H (low-bits) |
| | 02 H | Data content of address 0201 H | 00 H (high-bits) |
| CRC Check (Low) | C5 H (low-bits) | | 00 H (low-bits) |
| CRC Check (High) | D0 H (high-bits) | CRC Check (Low) | FA H (low-bits) |
| | | CRC Check (High) | 33 H (high-bits) |

For another example, write 1 (0001 H) into $01_H$ servo address $0070_H$. In which, the command code is $06_H$, and write 1 word into.

| **ASCII Mode** | | | |
|---|---|---|---|
| **Request information** | | **Response information** | |
| STX | ": " | STX | ": " |
| ADR | '0' | ADR | '0' |
| | '1' | | '1' |
| CMD | '0' | CMD | '0' |
| | '6' | | '6' |
| Data started address | '0' | Number of Date (calculated by byte) | '0' |
| | '0' | | '4' |
| | '7' | Data content of started address 0200 H | '0' |
| | '0' | | '0' |
| Number of Date (calculated by word) | '0' | | '7' |
| | '0' | | '0' |
| | '0' | Data content of address 0201 H | '0' |
| | '1' | | '0' |
| LRC Check | '8' | | '0' |
| | '8' | | '1' |
| End 1 | (0D H)(CR) | LRC Check | '8' |
| End 0 | (0A H)(LF) | | '8' |
| | | End 1 | (0D H )(CR) |
| | | End 0 | (0A H )(LF) |

| **RTU Mode** | | | |
|---|---|---|---|
| **Request information** | | **Response information** | |
| ADR | 01 H | ADR | 01 H |
| CMD | 06 H | CMD | 06 H |
| Data started address | 00 H (high-bits) | Data started address | 00 H (high-bits) |
| | 70 H (low-bits) | | 70 H (low-bits) |
| Date content | 00 H (high-bits) | Date content | 00 H (high-bits) |
| | 01 H (low-bits) | | 01 H (low-bits) |
| CRC Check (Low) | 49 H (low-bits) | CRC Check (Low) | 49 H (low-bits) |
| CRC Check (High) | D1 H (high-bits) | CRC Check (High) | D1H (high-bits) |

## Error Detection Value Calculation

- LRC check in ASCII mode

ASCII mode uses LRC (Longitudinal Redundancy Check) error detection value. The exceeded parts (e.g. the total value is $128_H$ of hex, then take $28_H$ only) is taken off by the unit of 256 in the total value from ADR to the last information, then calculate and compensate, the final result is LRC error detection value.

For example: read 1 word from $01_H$ servo address $0201_H$

| | |
|---|---|
| STX | ': ' |
| ADR | '0' |
| | '1' |
| CMD | '0' |
| | '3' |
| Data start address | '0' |
| | '2' |
| | '0' |
| | '1' |
| Number of Date (calculated by word) | '0' |
| | '0' |
| | '0' |
| | '1' |
| LRC Check | 'F' |
| | '8' |
| End 1 | ($0D_H$)(CR) |
| End 0 | ($0A_H$)(LF) |

**Add from ADR data to the last data.**

$01_H + 03_H + 02_H + 01_H + 00_H + 01_H = 08_H$

The compensate value is $F8_H$ when 2 is used to compensate $08_H$, so LRC is "F", "8".

- CRC check in RTU mode

RTU mode uses CRC (Cyclical Redundancy Check) error detection value.

The process of CRC error detection value calculation is shown as follows:

Step 1: Load in a 16-bit register of $FFFF_H$, named "CRC" register.

Step 2: Run XOR calculation between the first bit (bit 0) of instruction information and 16-bit CRC register's low-bits (LSB), and the result is saved to CRC register.

Step 3: Check the lowest bit (LSB) of CRC register, if it is 0, CRC register moves one bit to right; if it is 1, CRC register moves one bit to right, then run XOR calculation with $A001_H$;

Step 4: Go to Step 5 till the third step has been executed for 8 times, otherwise return to Step 3.

Step 5: Repeat the steps from Step 2 to Step 4 for the next bit of instruction information, the comment of CRC register is the CRC error detection value while all the bits have been executed by the same way.

## Example

After calculating out the CRC error detection value, the CRC low-bits should be filled first in instruction information, and then fill the high-bits of CRC. Please refer to the following example:

Read 2 words from the $0101_H$ address of $01_H$ servo. The final CRC register content calculated from ADR to the last bit of data is $3794_H$, and then the instruction information is shown as follows.

Please be sure that $94_H$ is transmitted before $37_H$.

| | |
|---|---|
| **ADR** | $01_H$ |
| **CMD** | $03_H$ |
| **Data start address** | $01_H$ (high-bits) |
| | $01_H$ (low-bits) |
| **Date number (calculated by word)** | $00_H$ (hit-bits) |
| | $02_H$ (low-bits) |
| **CRC check (low)** | $94_H$ (low-bits) |
| **CRC check (high)** | $37_H$ (high-bits) |
| **End 1** | Completed the communication |
| **End 0** | |

- ASCII mode
  Communication is ended with (0DH) - [carriage return] and (0AH) - [new line].
- RTU mode
  When the time exceeds the sleep interval by at least 4 bytes transmission time while in the current communication speed, it means the communication is finished.
- Example

  The following example uses C language to generate CRC value. The function needs two parameters.

```
unsigned char * data;
unsigned char length;
// The function will return unsigned integer type CRC value.
unsigned int crc_chk(unsigned char * data,unsigned char length){
   int i,j;
   unsigned int crc_reg=oxFFFF;
   While(length- -){
        crc_ reg ^=*data++;
        for(j=0;j<8;j++){
             If(crc_reg & 0x01){
   crc_reg=( crc_reg >>1)^0xA001;
}else{
   crc reg=crc reg >>1;
}
}
}
return crc_reg;
}
```

## 9.3.3 Communication Exception

Problems that occur during communication are a result of the following:

- Data address is incorrect while reading/writing parameters.
- The data is not within the parameter setting range while writing.
- Data transmission fault or checking code fault when communication is disturbed.

When the first and second communication faults occur, the servo drive is running normally, and will feed back an error frame.

When the third communication fault occurs, transmission data will be recognized as invalid to give up, and no error frame is returned.

The format of error frame:

| Host controller data frame | | | | |
|---|---|---|---|---|
| Start | Slave station address | Command | Data address,content | Check |
| – | – | [command] | – | – |

| Servo drive feeds back error frame | | | | |
|---|---|---|---|---|
| Start | Slave station address | Response code | Error code | Check |
| – | – | [command] + 80 $_H$ | – | – |

Error frame responses code = [command]+$80_H$

Error code = 00H:Normal communication
= 01H: Servodrive cannot identify the required functions
= $02_H$: The required data address does not exist in the servo drive
= 03H: The required data in servo drive is not allowed. (exceeds the range of parameter)
= 04H: Servodrive starts to perform the requirement, but cannot achieve it.

**For example:** Servo drive axis number is $03_H$, write data $06_H$into parameter Pn100 is not allowed , because the range of parameter Pn100is 0 to 6. The servo drive will feedback an error frame, the error code is $03_H$ (Beyond the parameter's maximum value or minimum value).

| Host controller data frame | | | | |
|---|---|---|---|---|
| Start | Slave station address | Command | Data address,content | Check |
| – | $03_H$ | $06_H$ | $0066_H$ $1388_H$ | – |

| Servo drive feedback error frame | | | | |
|---|---|---|---|---|
| Start | Slave station address | Response code | Error code | Check |
| – | $03_H$ | $86_H$ | $03_H$ | – |

Besides, if the data frame sent from host controller slave station address is $00_H$, it determines the data to be broadcast data. The servo drives will not feed back any frames.

## 9.3.4 Data Communication Address in Servodrive

The communication parameter addresses are shown in the following table:

| Address in Servodrive | | | Meaning | Description | Operation |
|---|---|---|---|---|---|
| Axis-A | Axis-B | Axis-C | | | |
| 0000 to 0369 | 2000 to 2369 | 4000 to 4369 | Parameter area | Corresponding parameters in parameter list | Read / Write |
| | | | | | |
| 081E to 0827 | 281E to 2827 | 481E to 4827 | Alarm information memory area | 10 of alarm records | Read only |
| | | | | | |
| 07FD | 27FD | 47FD | Iu zero offset | | Read only |

| Address in Servodrive | | | Meaning | Description | Operation |
|---|---|---|---|---|---|
| Axis-A | Axis-B | Axis-C | | | |
| 07FE | 27FE | 47FE | Iv zero offset | | Read only |
| | | | | | |
| 0806 to 0816 | 2806 to 2816 | 4806 to 4816 | Monitor data (corresponding with displayed data) | | Read only |
| 0806 | 2806 | 4806 | Speed feedback | Unit: rpm | Read only |
| 0809 | 2809 | 4809 | Internal torque reference percentage | Relative rated torque | Read only |
| 080A | 280A | 480A | The low 16 bits of the encoder rotation pulse number | | Read only |
| 080B | 280B | 480B | The high 16 bits of the encoder rotation pulse number | | Read only |
| 080C | 280C | 480C | The low 4 bits of input signal | | Read only |
| 080D | 280D | 480D | The high 4 bits of input signal | | Read only |
| 080E | 280E | 480E | Output signal state | | Read only |
| 080F | 280F | 480F | Reference pulse | | Read only |
| 0810 | 2810 | 4810 | The low-bits of the current position | Unit:1 reference pulse | Read only |
| 0811 | 2811 | 4811 | The high-bits of the current position | Unit: 10000 reference pulse | Read only |
| 0814 | 2814 | 4814 | The low 16 bits of deviation counter | | Read only |
| 0815 | 2815 | 4815 | The high 16 bits of deviation counter | | Read only |
| 0818 | 2818 | 4818 | The low 16 bits of reference pulse | Unit:1 reference pulse | Read only |
| 0819 | 2819 | 4819 | The high 16 bits of reference pulse | Unit:10000 reference pulses | Read only |
| 081C | 281C | 481C | Load inertia percentage | % | Read only |
| 081D | 281D | 481D | Motor overload ratio | % | Read only |
| 081E | 281E | 481E | Current alarm | % | Read only |
| 0850 | 2850 | 4850 | Electronic gear denominator in 16 bits | The dynamic electronic gear ratio set by Modbus will take effect immediately and will be reset after restart. | Read / Write |
| 0851 | 2851 | 4851 | 1st electronic gear numerator in 16 bits | | Read / Write |
| 0852 | 2852 | 4852 | 2nd electronic gear numerator in 16 bits | | Read / Write |
| 0853 | 2853 | 4853 | High-bits of electronic gear denominator in 32 bits | | Read / Write |

| Address in Servodrive | | | Meaning | Description | Operation |
|---|---|---|---|---|---|
| Axis-A | Axis-B | Axis-C | | | |
| 0854 | 2854 | 4854 | Low-bits of electronic gear denominator in 32 bits | | Read / Write |
| 0855 | 2855 | 4855 | High-bits of 1st electronic gear numerator in 32 bits | | Read / Write |
| 0856 | 2856 | 4856 | Low-bits of 1st electronic gear numerator in 32 bits | | Read / Write |
| 0857 | 2857 | 4857 | High-bits of 2nd electronic gear numerator in 32 bits | | Read / Write |
| 0858 | 2858 | 4858 | Low-bits of 2nd electronic gear numerator in 32 bits | | Read / Write |
| 0859 | 2859 | 4859 | Write 1 into this address, updating the electronic gear ratio in 32 bits to the relevant parameters | | Read / Write |
| | | | | | |
| 0900 | 2900 | 4900 | MODBUS communication IO signal | Reset when power off | Read / Write |
| 090E | | | DSP version | Represent the version by number | Read only |
| 090F | | | CPLD version | Represent the version by number | Read only |
| | | | | | |
| 1010 | 3010 | 5010 | Encoder multi-turn data | | Read only |
| 1011 | 3011 | 5011 | Low-bits of single-turn data | | Read only |
| 1012 | 3012 | 5012 | High-bits of single-turn data | | Read only |
| | | | | | |
| 1021 | 3021 | 5021 | Clear historical alarms | 01:Clear | Write |
| 1022 | 3022 | 5022 | Clear current alarms | 01:Clear | Write |
| 1023 | 3023 | 5023 | JOG fucntion | 01: Enabled<br>00: Disabled | Read / Write |
| 1024 | 3024 | 5024 | Forward rotation in JOG | 01: Start<br>00: Stop | Read / Write |
| 1025 | 3025 | 5025 | Reverse rotation in JOG | 01: Start<br>00: Stop | Read / Write |

- Parameter area (axis A is 0000 to 0348H, axis B is 2000 to 2348H, axis C is 4000 to 4348H)

Parameter address is relevant to the parameters in the parameter list.
For example, axis A parameter Pn000 is relevant to communication address $0000_H$; parameter Pn102 is relevant to communication address $0066_H$.

- Alarm information storage area (axis A is 07F1 to 07FAH, axis B is 27F1 to 27FAH, axis C is 47F1 to 47FAH)

| Alarm No. | Description | Communication address |
|---|---|---|
| 0 | Historical alarm 1 (the latest alarm) | $07F1_H$, $27F1_H$, $47F1_H$ |
| 1 to 8 | Historical alarm 2 to 9 | $07F2_H$ to $07F9_H$, $27F2_H$ to $27F9_H$, $47F2_H$ to $47F9_H$ |
| 9 | Historical alarm 10 (the furthest alarm) | $07FA_H$, $27FA_H$, $47FA_H$ |

- Monitor data area (axis A is 0806 to 0816H, axis B is 2806 to 2816H, axis C is 4806 to 4816H)

  The monitor data is corresponding to servo drive panel displays Un000 to Un016.

  For example: the corresponding data of communication address $0807_H$ (speed setting) is $FB16_H$.
  Therefore, the speed setting of axis A is -1258r/m.

- MODBUS communication IO signal

  Use communication to control digital IO signal. This data will not be saved after power off.

  It is operated with Pn512 as the communication input IO signal. That is to say, when the parameters setting in Pn512 enable the IO bit, the IO can be controlled by communication.

- Software version ($090F_H$)

  Use digit to represent servo drive software version. For example, if the read out data is $0100_H$, it means the software version is t-1.00.

# Chapter 10 Specifications and Dimension

## 10.1 Servodrive Specifications

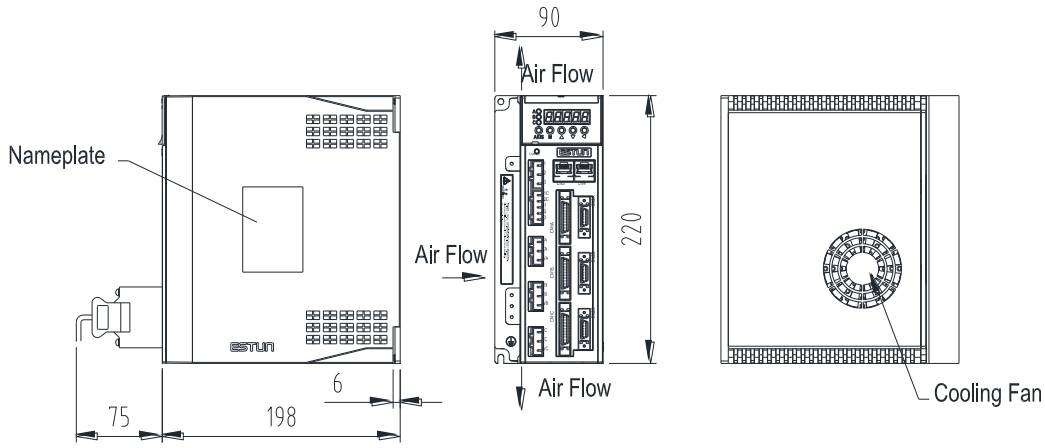
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Servodrive Model | | | ED3M-1010AMA / ED3M-101010AMA | | | | |
| Applicable Servomotor Model | | | EMJ-A5A* | EMJ-01A* | EMJ-04A* | EML-10A* | EMG-10A* |
| | | | EM3A-02A* | EM3A-04A* | EM3A-08A* | EM3A-10A* | – |
| Power Supply | Main Circuit | | Three-phase, 200 VAC to 230VAC, -15% to +10%, 50 Hz or 60Hz | | | | |
| | Control Circuit | | Single-phase, 200 VAC to 230 VAC, -15% to +10%, 50 Hz or 60Hz | | | | |
| Control Mode | | | SVPWM | | | | |
| Feedback | | | Incremental Wire-saving type:2500 P/R<br>Serial Encoder 131072P/R/1048576P/R | | | | |
| Operating Conditions | | Ambient/Storage Temperature | 0 to 55℃/-20 to 85℃ | | | | |
| | | Ambient/Storage Humidity | 90% RH (with no condensation) | | | | |
| | | Vibration/Shock Resistance | Vibration Resistance: 4.9m/s$^2$, Impact Resistance: 19.6m/s$^2$ | | | | |
| Configuration | | | Base-mounted | | | | |
| Speed Control | Speed Selection | Rotation Direction Selection | With /P-CON signal | | | | |
| | | Speed | Speed 1 to 7 | | | | |
| | Function | Soft Start Setting | 0 to 10s (Can be set individually for acceleration and deceleration.) | | | | |
| Position Control | Pulse Reference | Type | Sign + pulse train; CCW + CW pulse train;<br>90°phase difference 2-phase (phase A + phase B) | | | | |
| | | Form | Non-insulated line driver (about + 5V), open collector | | | | |
| | | Frequency | ×1 multiplier: 4Mpps<br>×2 multiplier: 2Mpps<br>×4 multiplier: 1Mpps<br>Open collector: 200Kpps<br>Frequency will begin to decline when the duty ratio error occurs.. | | | | |
| | Position Reference Setting | Position Setting | 16 position nodes can be set. | | | | |
| I/O Signals | Encoder Dividing Pulses Output | | Phase-A, phase-B, phase-C, line driver output<br>Number of dividing pulses: (1 to 16384)/16384 | | | | |
| | Sequence Input | Number of channels | 3×8 channels | | | | |
| | | Function | Signal allocations and positive/negative logic modifications:<br>Servo ON (/S-ON), P control (/P-CON), alarm reset (/ALM-RST), position error clear (/CLR), forward run prohibited (P-OT), reverse run prohibited (N-OT), forward current limit (/P-CL), reverse current limit (/N-CL) and so on. | | | | |
| | Sequence Output | Number of channels | 3×4 channels | | | | |

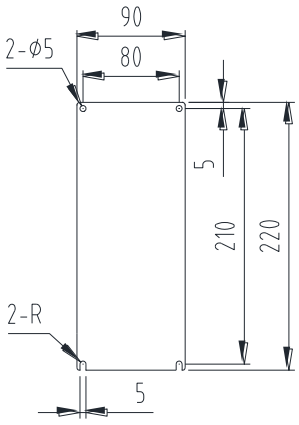
  

| | | | |
|---|---|---|---|
| | | Function | Signal allocations and positive/negative logic modifications: Positioning completion(/COIN), speed coincidence(/V-CMP), servomotor rotation detection(/TGON), servo ready (/S-RDY), torque limit output(/CLT), brake interlock output (/BK), encoder C pulse(/PGC), Over travel/OT) and so on. |
| Internal Functions | Dynamic Brake | | Each axis with dynamic brake function, which operated at main power OFF, servo alarm, servo OFF or overtravel. |
| | Protection Functions | | Overcurrent, overvoltage, low voltage, overload, regeneration error, overspeed, etc. |
| | Utility Function | | Alarm trace back, JOG operation, load inertia detection, etc. |
| | Communication Functiion | | RS-485 communication port, MODBUS protocol, CAN communication port, CANopen protocol; |
| | Display Function | | CHARGE×1, power ×1, Axis LED×3, 7-segment LEDS×5, push button ×5 |

# 10.2 **Servodrive Dimension**

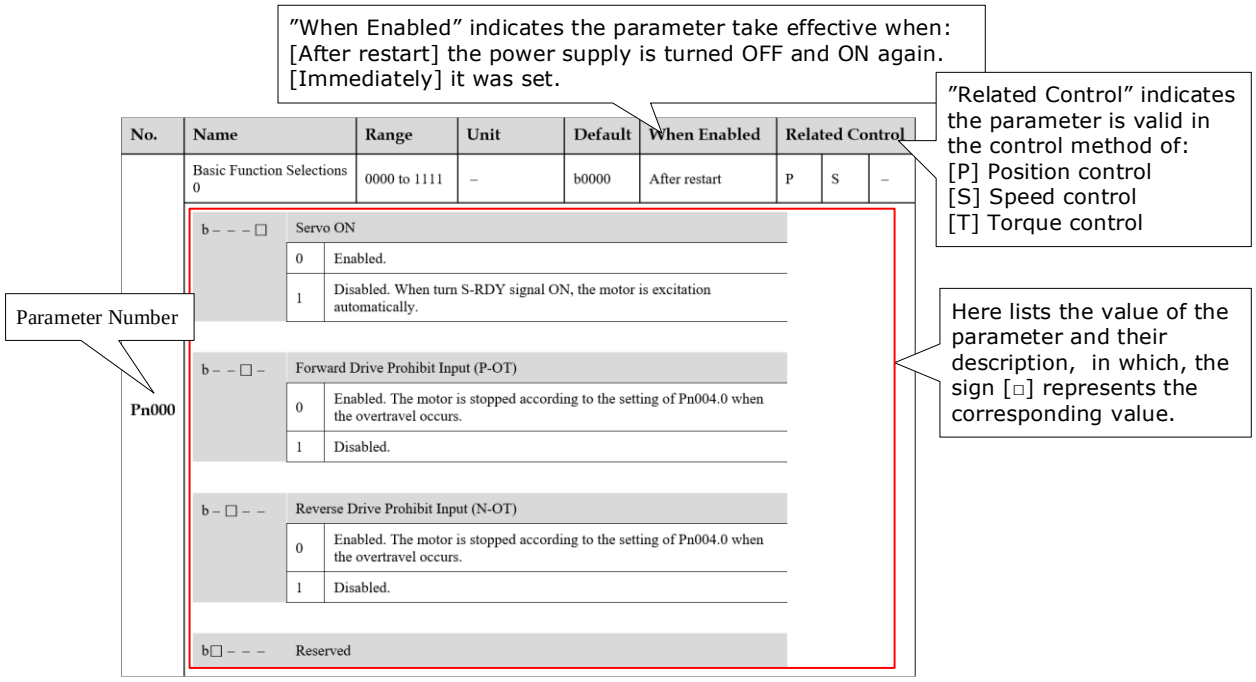
Unit: mm



# Appendix A Parameters List

## A.1 Interpreting the Parameters List



# A.2 List of Servo Parameters

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn000 | Function Selections 0 | 0000 to 1111 | – | 0000 | After restart | P | S | – |

b – – – □ Servo ON

| | |
|---|---|
| 0 | Enabled. |
| 1 | Disabled. When turn S-RDY signal ON, the motor is excitation automatically. |

b – – □ – Forward Drive Prohibit Input (P-OT)

| | |
|---|---|
| 0 | Enabled. The motor is stopped according to the setting of Pn004.0 when the overtravel occurs. |
| 1 | Disabled. |

b – □ – – Reverse Drive Prohibit Input (N-OT)

| | |
|---|---|
| 0 | Enabled. The motor is stopped according to the setting of Pn004.0 when the overtravel occurs. |
| 1 | Disabled. |

b□ – – – Reserved

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Pn001 | Function Selections 1 | 0000 to 1111 | – | 0000 | After restart | P | S | – |

b – – – □ Rotation Direction Selection

| | |
|---|---|
| 0 | Use CCW as the forward direction. |
| 1 | Use CW as the forward direction. |

b – – □ – Reserved

b – □ – – Reserved

b□ – – – 2nd electronic gear functon (only valid for position control)

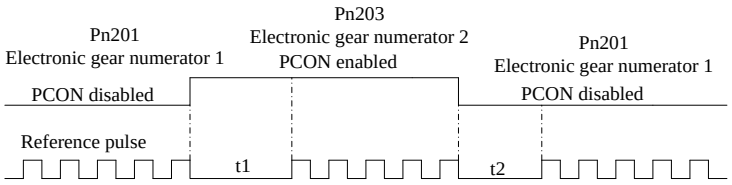
| | |
|---|---|
| 0 | Disbaled, regards /P-CON sginal as P/PI switching. |
| 1 | Enabled, regards /P-CON sginal as 2nd electronic gear switching. \ |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn002** | Function Selections 2 | 0000 to 1111 | – | 0010 | After restart | P | S | – |

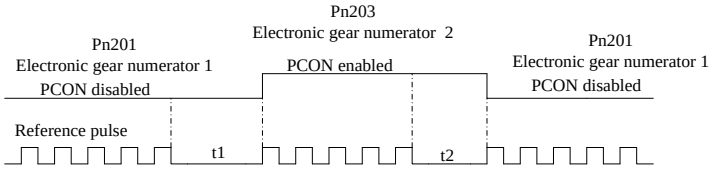
b – – – □ Electronic gear switching mode

0



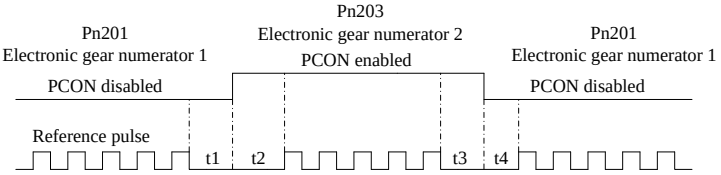

t1，t2>1ms

1



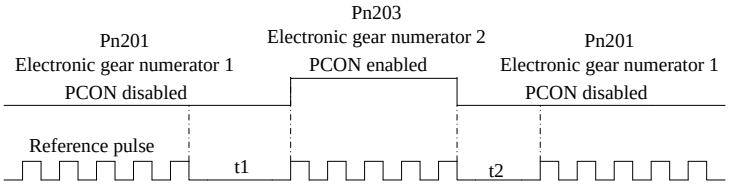

t1，t2>1ms

Time sequence when Pn002.0=0 or 1




t1，t2，t3，t4>1ms

Error time sequence:




t1，t2>1ms

b – – □ – Reserved

b – □ – – Reserved

b□ – – – Reserved

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn003 | Function Selections 3 | 0000 to 1111 | – | 0000 | After restart | P | S | – |

b – – – □ Reserved

b – – □ – Reserved

b – □ – – Low speed compensation

| | |
|---|---|
| 0 | Disabled. |
| 1 | Enabled. To enhance avoid servomotor creeping, but the degree of correction is determined by the setting in Pn219. |

b□ – – – Overload enhancement

| | |
|---|---|
| 0 | Disabled. |
| 1 | Enabled. To enhance the overload capacity when servomotor exceeds the 2 times rated overload. It is used in frequent power ON/OFF occasions. |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn004 | Function Selections 4 | 0000 to 3425 | – | 0000 | After restart | P | S | – |

H – – – □ Stop Method

| | |
|---|---|
| 0 | Stops the servomotor by applying DB and then releases DB. |
| 1 | Coast to a stop. |
| 2 | Stops the servomotor by DB when servo OFF, stops the servomotor by plug braking when overtravel, then places it into coast (power OFF) mode. |
| 3 | Makes the servomotor coast to a stop state when servo OFF, stops the servomotor by plug braking when overtravel, then places it into coast (power OFF) mode. |
| 4 | Stops the servomotor by DB when servo OFF, stops the servomotor by plug braking when overtravel, then places it into zero clamp mode. |
| 5 | Makes the servomotor coast to a stop state when servo OFF, stops the servomotor by plug braking when overtravel, then places it into zero clamp mode. |

H – – □ – Deviation counter clear mode (only valid for position control)

| | |
|---|---|
| 0 | Clear error pulse when S-0FF, not clear error pulse when overtravel. |
| 1 | Not clear error pulse |
| 2 | Clear error pulse When S-OFF or over travel |

H – □ – – Reference pulse form (only valid for position control)

| | |
|---|---|
| 0 | Sign + Pulse |
| 1 | CW + CCW |
| 2 | A + B (×1) |
| 3 | A + B (×2) |
| 4 | A + B (×4) |

H□ – – – Inverses pulse (only valid for position control)

| | |
|---|---|
| 0 | Do not inverse PULS reference and SIGN reference |
| 1 | Do not inverse PULS reference; Inverses SIGN reference. |
| 2 | Inverse PULS reference;Do not inverse SIGN reference |
| 3 | Inverse PULS reference and SIGN reference |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn005 | Function Selections 5 | 0000 to 3351 | – | 0000 | After restart | P | S | – |

H – – – □ Torque feedforward form

| | |
|---|---|
| 0 | Use general torque feedforward |
| 1 | Use high-speed torque feedforward |

H – – □ – Control method

| | |
|---|---|
| 0 | Speed control(Parameter reference)<br>**PCON is invalid.** |
| 1 | Position control(pulse train reference)<br>**PCON: OFF, PI control; ON, P control** |
| 2 | Speed control (contact reference) ↔ speed Control (zero reference)<br>**PCON, PCL, NCL: OFF Switches to position control (zero reference)** |
| 3 | Speed control (contact reference) ↔ position control (pulse train reference)<br>**PCON, PCL, NCL: OFF Switches to position control(pulse train reference)** |
| 4 | Positin control (pulse train reference) ↔position control (INHIBIT)<br>**PCON: OFF, Position control (pulse train reference); ON, position control (INHIBIT)** |
| 5 | Position control (contact reference)<br>**PCON: Used to change step; PCL, NCL: Used to search reference point or start** |

H – □ – – Out-of-tolerance alarm selection (only valid for position control)

| | |
|---|---|
| 0 | Disabled. |
| 1 | Enabled. An alarm occurs when the value of deviation counter exceeds Pn504. |
| 2 | Reserved |
| 3 | Reserved |

H□ – – – Servomotor model selection

| | |
|---|---|
| 0 | EM3A/EMJ |
| 1 | EMG |
| 2 | EML |
| 3 | Reserved |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn006 | Function Selections 6 | 0000 to 2203 | – | 0000 | After restart | P | S | T |

H – – – □ Bus type selection

| | |
|---|---|
| 0 | No bus |
| 1 | Reserved |
| 2 | Reserved |
| 3 | CANopen |

H – – □ – Reserved

H – □ – – Low-frequency vibration suppression switch

| | |
|---|---|
| 0 | Low-frequency vibration suppression function disabled |
| 1 | Speed low-frequency vibration suppression function enabled |
| 2 | Position low-frequency vibration suppression function enabled |

H□ – – – Reference input filter for open collector signal

| | |
|---|---|
| 0 | • Pn840.0=3/4/5, the max value of servo receiving pulse frequency⩽4M<br>• Pn840.0=6, the max value of servo receiving pulse frequency⩽700k |
| 1 | • Pn840.0=3/4/5, the max value of servo receiving pulse frequency⩽650k<br>• Pn840.0=6, the max value of servo receiving pulse frequency⩽200k |
| 2 | • Pn840.0=3/4/5, the max value of servo receiving pulse frequency⩽150k<br>• Pn840.0=6, the max value of servo receiving pulse frequency⩽60k |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn007 | Function Selections 7 | 0000 to 1111 | – | 0000 | After restart | P | S | T |

b – – – □ Witdh of C-pulse

| | |
|---|---|
| 0 | Standard width of C pulse |
| 1 | Wider the width of C pulse |

b – – □ – Reserved

b – □ – – Reserved

b□ – – – Torque filter

| | |
|---|---|
| 0 | Standard torque filter |
| 1 | New type torque filter |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn009 | Function Selections 9 | 0000 to 0100 | – | 0000 | After restart | P | – | – |

b – – – ☐ Reserved

b – – ☐ – Reserved

b – ☐ – – Electronic gear selection

| | |
|---|---|
| 0 | 16-bit electronic gear |
| 1 | 32-bit electronic gear |

b☐ – – – Reserved

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Pn010 | Function Selections 10 | 0000 to 0001 | – | 0001 | After restart | P | S | T |

H – – – ☐ Automatic identification of motor selection

| | |
|---|---|
| 0 | Disabled. |
| 1 | Enabeld.<br>The drive, motor and encoder models are automatically acquired and the corresponding servo parameters are automatically loaded. |

H – – ☐ – Reserved

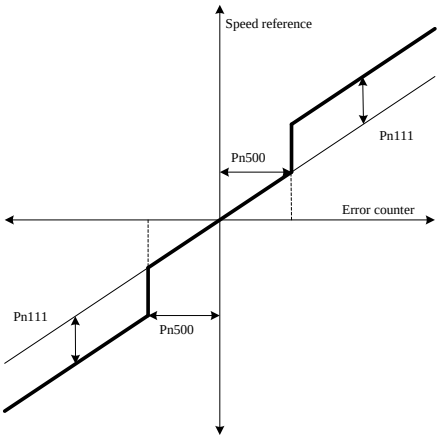
H – ☐ – – Reserved

H☐ – – – Reserved

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn100</td><td>Turning method</td><td>0000 to 0006</td><td>–</td><td>0000</td><td>After restart</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">
H – – – □ Turning Method Selection
<table>
<tr><td>0</td><td>Manual Turning</td></tr>
<tr><td>1</td><td>Perform automatic tuning of conventional load with no-change in inertia</td></tr>
<tr><td>2</td><td>Perform automatic tuning of conventional load with less change in inertia</td></tr>
<tr><td>3</td><td>Perform automatic tuning of conventional load with more change in inertia</td></tr>
<tr><td>4</td><td>Perform automatic tuning of vertical load with no-change in inertia</td></tr>
<tr><td>5</td><td>Perform automatic tuning of vertical load with less change in inertia</td></tr>
<tr><td>6</td><td>Perform automatic tuning of vertical load with more change in inertia</td></tr>
</table>
H – – □ – Reserved<br>
H – □ – – Reserved<br>
H□ – – – Reserved
</td></tr>
<tr><td rowspan="2">Pn101</td><td>Machine rigidity setting</td><td>0 to 15</td><td>–</td><td>5</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">The response speed of servo system is determined by this parameter. Normally, the rigidity should be set a little larger. However, if it is too large, it would suffer mechanical impact. It should be set a little smaller when large vibration is present. This parameter is only valid in autotuning.</td></tr>
<tr><td rowspan="2">Pn102</td><td>Speed loop gain</td><td>1 to 4000</td><td>Hz</td><td>300</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">This parameter determines speed loop gain.</td></tr>
<tr><td rowspan="2">Pn103</td><td>Speed loop integral time constant</td><td>1 to 4096</td><td>0.25ms</td><td>80</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Decreases the value of this parameter to shorten positioning time and enhance speed response.</td></tr>
<tr><td rowspan="2">Pn104</td><td>Position loop gain</td><td>0 to 1000</td><td>1/s</td><td>40</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">This parameter determines position loop gain. Decreases this value to enhance servo rigidity, but vibration will occur if the value is too large.</td></tr>
<tr><td rowspan="2">Pn105</td><td>Torque reference filter time constant</td><td>0 to 250</td><td>0.25ms</td><td>2</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Torque reference filter can eliminate or lighten mechanical vibration, but incorrect setting will result to mechanical vibration.</td></tr>
<tr><td rowspan="2">Pn106</td><td>Load inertia percentage</td><td>0 to 20000</td><td>%</td><td>0</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Setting value = (load inertia/rotor inertia) × 100</td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn107 | 2nd speed loop gain | 1 to 4000 | Hz | 320 | Immediately | P | S | – |
| | The meanings of these parameters are the same as Pn102.<br>This parameters is only needed to set when two types of gain function are enabled. | | | | | | | |
| Pn108 | 2nd speed loop integral time constant | 1 to 4096 | 0.25ms | 40 | Immediately | P | S | – |
| | The meanings of these parameters are the same as Pn103.<br>This parameters is only needed to set when two types of gain function are enabled. | | | | | | | |
| Pn109 | 2nd position loop gain | 0 to 1000 | Rad/s | 40 | Immediately | P | – | – |
| | The meanings of these parameters are the same as Pn104.<br>This parameters is only needed to set when two types of gain function are enabled. | | | | | | | |
| Pn110 | 2nd torque reference filter time constant | 0 to 250 | 0. 25ms | 2 | Immediately | P | S | – |
| | The meanings of these parameters are the same as Pn105.<br>This parameters is only needed to set when two types of gain function are enabled. | | | | | | | |
| Pn111 | Speed bias | 0 to 300 | rpm | 0 | Immediately | P | – | – |
| | This parameter setting can shorten positioning time. However, if it is too large or does not cooperate with Pn111 correctly, vibration will occur.<br>The relationship with speed reference, deviation counter, positioning error is shown in the following chart.<br>  <br> | | | | | | | |
| Pn112 | Feedforward | 0 to 100 | % | 0 | Immediately | P | – | – |
| | It is used to set position feedforward. The response speed is faster and position error is less when this parameter setting is higher. Vibration will occur if the value is set too large. | | | | | | | |
| Pn113 | Feedforward filter | 0 to 640 | 0.25ms | 0 | Immediately | P | – | – |
| | It is used to ease mechanical vibration due to position feedforward. The feedforward lag will be enlarged and result to vibration if the value is set too large. | | | | | | | |

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn114</td><td>Torque feedforward</td><td>0 to 100</td><td>%</td><td>0</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">It is used to set torque feedforward, and enhance response speed.<br>Set the load inertia percentage (Pn106) correctly to enable this function in manual gain adjustment mode.</td></tr>
<tr><td rowspan="2">Pn115</td><td>Torque feedforward filter</td><td>0 to 640</td><td>0.25ms</td><td>0</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">It is used to ease mechanical vibration due to torque feedforward.</td></tr>
<tr><td rowspan="2">Pn116</td><td>P/PI switching condition</td><td>0 to 4</td><td>–</td><td>4</td><td>After restart</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8"><table>
<tr><td>0</td><td>Torque reference percentage</td></tr>
<tr><td>1</td><td>Value of offset counter</td></tr>
<tr><td>2</td><td>Value of acceleration speed setting</td></tr>
<tr><td>3</td><td>Value of speed setting</td></tr>
<tr><td>4</td><td>Fixed PI</td></tr>
</table></td></tr>
<tr><td rowspan="2">Pn117</td><td>Torque switching threshold</td><td>0 to 300</td><td>%</td><td>200</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Threshold of torque to switch PI control to P control.</td></tr>
<tr><td rowspan="2">Pn118</td><td>Offset counter switching threshold</td><td>0 to 10000</td><td>1 Pulse</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">Threshold of deviation counter to switch PI control to P control.</td></tr>
<tr><td rowspan="2">Pn119</td><td>Setting acceleration speed switching threshold</td><td>0 to 3000</td><td>10 rpm/s</td><td>0</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Threshold of acceleration speed to switch PI control to P control.</td></tr>
<tr><td rowspan="2">Pn120</td><td>Setting speed switching threshold</td><td>0 to 10000</td><td>rpm</td><td>0</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Threshold of speed to switch PI control to P control.</td></tr>
<tr><td rowspan="2">Pn121</td><td>Gain switching condition</td><td>0 to 8</td><td>–</td><td>0</td><td>After restart</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8"><table>
<tr><td>0</td><td>Fix to 1st group gain</td></tr>
<tr><td>1</td><td>External switch gain switching (G-SEL)</td></tr>
<tr><td>2</td><td>Torque percentage</td></tr>
<tr><td>3</td><td>Value of offset counter</td></tr>
<tr><td>4</td><td>Value of acceleration speed setting (10rpm)</td></tr>
<tr><td>5</td><td>Value of speed setting</td></tr>
<tr><td>6</td><td>Speed reference input</td></tr>
<tr><td>7</td><td>Actual motor speed</td></tr>
<tr><td>8</td><td>Pn123 + Pn124</td></tr>
</table></td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn122** | Switching delay time | 0 to 20000 | 0.25ms | 0 | Immediately | P | S | – |
| | Delay time of switching gain when switching condition is satisfied. | | | | | | | |
| **Pn123** | Switch threshold level | 0 to 20000 | – | 0 | Immediately | P | S | – |
| | Gain switching trigger level | | | | | | | |
| **Pn124** | Reserved | | | | | | | |
| **Pn125** | Position gain switching time | 0 to 20000 | 0.25ms | 0 | Immediately | P | – | – |
| | This parameter is used to smooth transition if the change of the two groups of gain is too large. | | | | | | | |
| **Pn126** | Hysteresis switching | 0 to 20000 | – | 0 | Immediately | P | S | – |
| | This parameter is used to set the operation hysteresis of gain switching. | | | | | | | |
| **Pn127** | Low speed detection filter | 0 to 100 | 0.25ms | 10 | Immediately | P | S | – |
| | This parameter is used to filter in low speed detection. The speed detection will be lagged if the value is too large. | | | | | | | |
| **Pn128** | Speed gain acceleration relationship during online autotuning | 0 to 3 | – | 3 | Immediately | P | S | – |
| | The increasing multiple of speed loop gain is the same rigidity during online autotuning. The speed loop gain is larger when this value is higher. | | | | | | | |
| **Pn129** | Low speed correction coefficient | 0 to 30000 | – | 0 | Immediately | P | S | – |
| | The intensity of anti-friction and anti-creeping at low speed. Vibration will occur if this value is set too large. | | | | | | | |
| **Pn130** | Friction Load | 0 to 3000 | 0.1% | 0 | Immediately | P | S | – |
| | Frictin load or fixed load compensation | | | | | | | |
| **Pn131** | Friction compensation speed hysteresis area | 0 to 100 | rpm | 0 | Immediately | P | S | – |
| | Threshold of friction compensation start | | | | | | | |
| **Pn132** | Sticking friction load | 0 to 1000 | 0.1%/1000rpm | 0 | Immediately | P | S | – |
| | Sticking damp which is in direct proportion to speed. | | | | | | | |
| **Pn200** | PG divided ratio | 16 to 16384 | 1 Pulse | 16384 | After restart | P | S | – |
| | Analog encoder output orthogonal difference pulses.<br>The meaning of this value is the number of analog encoder output orthogonal difference pulses per one servomotor rotation. | | | | | | | |

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn201</td><td>1st electronic gear numerator</td><td>1 to 65535</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">The electronic gear enables the reference pulse to relate with the servomotor travel distance, so the host controller doesn't change the mechanical deceleration ratio and encoder pulses. In fact, it is the setting of frequency doubling or frequency division to the reference pulses.<br>$$\frac{Numerator\ (Pn201\ or\ Pn203)}{Denominator\ (Pn202)}$$</td></tr>
<tr><td rowspan="2">Pn202</td><td>Electronic gear denominator</td><td>1 to 65535</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">To employ by combining with Pn201.</td></tr>
<tr><td rowspan="2">Pn203</td><td>2nd electronic gear numerator</td><td>1 to 65535</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">Same with the meaning of Pn201.</td></tr>
<tr><td rowspan="2">Pn204</td><td>Position reference acceleration/deceleration time constant</td><td>0 to 32767</td><td>0.25ms</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">This value is used to smooth the input pulses. The effect of smoothness is better when the value is higher, but lag will occur if the value is too large.</td></tr>
<tr><td rowspan="2">Pn205</td><td>Position reference filter form selection</td><td>0 to 1</td><td>–</td><td>0</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8"><table><tr><td>0</td><td>1st order filter</td></tr><tr><td>1</td><td>2nd order filter</td></tr></table></td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn206</td><td>Pluse input selection</td><td>0000 to 0033</td><td>–</td><td>0000</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">
H – – – □ Pluse input port selection
<table>
<tr><td>0</td><td>Use pluse input themselves</td></tr>
<tr><td>1</td><td>Use A-axis pluse input port</td></tr>
<tr><td>2</td><td>Use B-axis pluse input port</td></tr>
<tr><td>3</td><td>Use C-axis pluse input port</td></tr>
</table>
H – – □ – Synchronize selection
<table>
<tr><td>0</td><td>A-axis and B-axis use the same pluse input port setted by Pn206.0, C-axis use C-axis pluse input port</td></tr>
<tr><td>1</td><td>A-axis and C-axis use the same pluse input port setted by Pn206.0, B-axis use B-axis pluse input port</td></tr>
<tr><td>2</td><td>B-axis and C-axis use the same pluse input port setted by Pn206.0, A-axis use A-axis pluse input port</td></tr>
<tr><td>3</td><td>A-axis, B-axis and C-axis use the same pluse input port</td></tr>
</table>
H – □ – – Reserved<br>
H□ – – – Reserved
</td></tr>
<tr><td rowspan="2">Pn304</td><td>Parameter speed</td><td>-6000 to 6000</td><td>rpm</td><td>500</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">The parameter can be set to positive or negative. When control mode is set to D, it determines the speed of motor.</td></tr>
<tr><td rowspan="2">Pn305</td><td>JOG speed</td><td>0 to 6000</td><td>rpm</td><td>500</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">It is used to set JOG rotation speed, and the direction is determined by the pressing key during JOG operation.</td></tr>
<tr><td rowspan="2">Pn306</td><td>Soft start acceleration time</td><td>0 to 10000</td><td>ms</td><td>200</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">The time for trapeziform acceleration to accelerate to 1000rpm.</td></tr>
<tr><td rowspan="2">Pn307</td><td>Soft start deceleration time</td><td>0 to 10000</td><td>ms</td><td>200</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">The time for trapeziform deceleration to decelerate to 1000rpm.</td></tr>
<tr><td rowspan="2">Pn308</td><td>Speed filter time constant</td><td>0 to 10000</td><td>ms</td><td>0</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">1st order filter time constant</td></tr>
<tr><td rowspan="2">Pn309</td><td>S curve risetime</td><td>0 to 10000</td><td>ms</td><td>0</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">The time for transition from one point to another point in S curve.</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn310</td><td>Speed reference curve form</td><td>0 to 3</td><td>–</td><td>0</td><td>After restart</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8"><table>
<tr><td>0</td><td>Slope</td></tr>
<tr><td>1</td><td>S curve</td></tr>
<tr><td>2</td><td>1st order filter</td></tr>
<tr><td>3</td><td>2nd order filter</td></tr>
</table></td></tr>
<tr><td rowspan="2">Pn311</td><td>S form selection</td><td>0 to 3</td><td>–</td><td>0</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">This value determines the transition form of S curve.</td></tr>
<tr><td rowspan="2">Pn316</td><td>Internal speed 1</td><td>-6000 to 6000</td><td>rpm</td><td>100</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">Internal speed is enabled when Pn005.1=3, 4, 5, 6
<table>
<tr><td colspan="3">Input signal</td><td rowspan="2">operating speed</td></tr>
<tr><td>/P-CON</td><td>/PCL</td><td>/NCL</td></tr>
<tr><td rowspan="4">OFF(H)</td><td>OFF(H)</td><td>OFF(H)</td><td>Zero speed or switch to other control modes</td></tr>
<tr><td>OFF(H)</td><td>ON(L)</td><td>Internal speed 1</td></tr>
<tr><td>ON(L)</td><td>OFF(H)</td><td>Internal speed 2</td></tr>
<tr><td>ON(L)</td><td>ON(L)</td><td>Internal speed 3</td></tr>
<tr><td rowspan="4">ON(L)</td><td>OFF(H)</td><td>OFF(H)</td><td>Internal speed 4</td></tr>
<tr><td>OFF(H)</td><td>ON(L)</td><td>Internal speed 5</td></tr>
<tr><td>ON(L)</td><td>OFF(H)</td><td>Internal speed 6</td></tr>
<tr><td>ON(L)</td><td>ON(L)</td><td>Internal speed 7</td></tr>
</table></td></tr>
<tr><td rowspan="2">Pn317</td><td>Internal speed 2</td><td>-6000 to 6000</td><td>rpm</td><td>200</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">See the description of Pn316.</td></tr>
<tr><td rowspan="2">P318</td><td>Internal speed 3</td><td>-6000 to 6000</td><td>rpm</td><td>300</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">See the description of Pn316.</td></tr>
<tr><td rowspan="2">Pn319</td><td>Internal speed 4</td><td>-6000 to 6000</td><td>rpm</td><td>-100</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">See the description of Pn316.</td></tr>
<tr><td rowspan="2">Pn320</td><td>Internal speed 5</td><td>-6000 to 6000</td><td>rpm</td><td>-200</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">See the description of Pn316.</td></tr>
<tr><td rowspan="2">Pn321</td><td>Internal speed 6</td><td>-6000 to 6000</td><td>rpm</td><td>-300</td><td>Immediately</td><td>–</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">See the description of Pn316.</td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn322** | Internal speed 7 | -6000 to 6000 | rpm | 500 | Immediately | – | S | – |
| | See the description of Pn316. | | | | | | | |
| **Pn401** | Forward torque internal limit | 0 to 350 | % | 300 | Immediately | P | S | – |
| | Servomotor output torque limit value .(depending on the actual overload capacity). | | | | | | | |
| **Pn402** | Reverse torque internal limit | 0 to 350 | % | 300 | Immediately | P | S | – |
| | Servomotor output torque limit value .(depending on the actual overload capacity). | | | | | | | |
| **Pn403** | Forward external torque limit | 0 to 300 | % | 100 | Immediately | P | S | – |
| | Servomotor output torque limit value .(depending on the actual overload capacity). | | | | | | | |
| **Pn404** | Reverse external torque limit | 0 to 300 | % | 100 | Immediately | P | S | – |
| | Servomotor output torque limit value .(depending on the actual overload capacity). | | | | | | | |
| **Pn405** | Plug braking torque limit | 0 to 300 | % | 300 | Immediately | P | S | – |
| | Servomotor output torque limit value .(depending on the actual overload capacity). | | | | | | | |
| **Pn406** | Speed limit during torque control | 0 to 6000 | rpm | 1500 | Immediately | – | – | T |
| | Speed limit during torque control. | | | | | | | |
| **Pn407** | Notch filter 1 frequency | 50 to 2000 | Hz | 2000 | Immediately | P | S | – |
| | • In some conditions, vibration will be picked up and response will be lagged after notch filter is set.<br>• When notch filter frequency is set to 5000, the notch filter is invalid. | | | | | | | |
| **Pn408** | Notch filter 1 depth | 0 to 11 | – | 1 | Immediately | P | S | – |
| | – | | | | | | | |
| **Pn409** | Notch filter 2 frequency | 50 to 2000 | Hz | 2000 | Immediately | P | S | – |
| | – | | | | | | | |
| **Pn410** | Notch filter 2 depth | 0 to 11 | – | 1 | Immediately | P | S | – |
| | – | | | | | | | |
| **Pn411** | Low frequency vibration frequency | 50 to 500 | 0.1Hz | 100 | Immediately | P | S | – |
| | Frequency of low frequency vibration with load. | | | | | | | |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn412** | Low frequency vibration damp | 0 to 200 | – | 25 | Immediately | P | S | – |
| | Attenuation damp of low frequency vibration with load. It does not need to change. | | | | | | | |
| **Pn413** | Torque control delay time | 1 to 2000 | 0.25ms | 100 | Immediately | – | – | T |
| | – | | | | | | | |
| **Pn414** | Torque control speed hysteresis | 10 to 1000 | rpm | 50 | Immediately | – | – | T |
| | – | | | | | | | |
| **Pn500** | Positioning error | 0 to 5000 | 1 Pulse | 100 | Immediately | P | – | – |
| | Outputs /COIN signal when deviation counter is less than this value. | | | | | | | |
| **Pn501** | Coincidence difference | 0 to 100 | rpm | 10 | Immediately | P | – | – |
| | Outputs /VCMP signal when the difference between speed reference value and speed feedback value is less than this value. | | | | | | | |
| **Pn502** | Zero clamp speed | 0 to 3000 | rpm | 10 | Immediately | P | – | – |
| | A position loop is created and the Servomotor remains stopped at a position reference of 0 when the corresponding speed of the input analog signal less than this setting. (The current stop position is held.) | | | | | | | |
| **Pn503** | Rotation detection speed TGON | 0 to 3000 | rpm | 20 | Immediately | P | S | – |
| | When the servomotor speed exceeds this parameter setting value, it means that the servomotor has already rotated steadily and outputs /TGON signal. | | | | | | | |
| **Pn504** | Offset counter overflow alarm | 1 to 32767 | 256 Pulses | 1024 | Immediately | P | – | – |
| | When the value in deviation counter exceeds this parameter setting value, it means that deviation counter alarm has occurred and outputs alarm an signal. | | | | | | | |
| **Pn505** | Servo ON waiting time | -2000 to 2000 | ms | 0 | Immediately | P | S | – |
| | Parameters Pn505, Pn506, Pn507 and Pn508 are only enabled when the port output parameters are allocated with /BK signal output.<br>These parameters are used to keep braking (prevent from gravity glissade or continuous outside force on servomotor) time sequence.<br>• For the parameter is plus,/BK signal is output firstly when servo-ON signal is input, and then servomotor excitation signal is created after delaying the parameter setting time.<br>• For the parameter is minus, servomotor excitation signal is output firstly when servo-ON signal is input, and then /BK signal is created after delaying the parameter setting time. | | | | | | | |
| **Pn506** | Basic waiting flow | 0 to 500 | 10ms | 10 | Immediately | P | S | – |
| | Standard setting: /BK output (braking action) and servo-OFF are at the same time.<br>At present, the machine movable part may shift slightly due to gravity according to mechanical configuration and character; it can be eliminated by using the parameters when the servomotor is at stop or at a low speed. | | | | | | | |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn507 | Brake waiting speed | 10 to 100 | rpm | 100 | Immediately | P | S | – |
| | The /BK signal is output when the servomotor speed is decreased below the parameter setting value at servo-OFF. | | | | | | | |
| Pn508 | Brake waiting time | 10 to 100 | 10ms | 50 | Immediately | P | S | – |
| | The /BK signal is output when the delay time exceeds the parameter setting value after servo-OFF.<br>The /BK signal is output as long as either of the brake waiting speed or brake waiting time is satisfied. | | | | | | | |
| Pn509 | Input signals allocation 1 | 0000 to EEEE | – | 3210 | After restart | P | S | – |

H – – – □ Allocate signal to CN1□-10

| | |
|---|---|
| 0 | /S-ON |
| 1 | /P-CON |
| 2 | P-OT |
| 3 | N-OT |
| 4 | ALMRST |
| 5 | /CLR |
| 6 | /PCL |
| 7 | /NCL |
| 8 | /G-SEL |
| 9 | /JDPOS-JOG+ |
| A | /JDPOS-JOG- |
| B | /JDPOS-HALT |
| C | HmRef |
| D | SHOME |
| E | ORG(ZPS) |

H – – □ – Allocate signal to CN1□-11

0 to E: same as the allocation of CN1□-10.

H – □ – – Allocate signal to CN1□-12

0 to E: same as the allocation of CN1□-10.

H□ – – – Allocate signal to CN1□-13

0 to E: same as the allocation of CN1□-10.

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn510 | Input signals allocation 2 | 0000 to EEEE | – | 7654 | After restart | P | S | – |

H – – – □ Allocate signal to CN1□-14

| | |
|---|---|
| 0 | /S-ON |
| 1 | P-CON |
| 2 | P-OT |
| 3 | N-OT |
| 4 | ALMRST |
| 5 | /CLR |
| 6 | /PCL |
| 7 | /NCL |
| 8 | /G-SEL |
| 9 | /JDPOS-JOG+ |
| A | /JDPOS-JOG- |
| B | /JDPOS-HALT |
| C | Reserved |
| D | SHOME |
| E | ORG(ZPS) |

H – – □ – Allocate signal to CN1□-15

0 to E: same as the allocation of CN1□-14.

H – □ – – Allocate signal to CN1□-16

0 to E: same as the allocation of CN1□-14.

H□ – – – Allocate signal to CN1□-17

0 to E: same as the allocation of CN1□-14.

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn511 | Output signals allocation | 0000 to 0999 | – | 0210 | After restart | P | S | – |

H – – – □ Allocate signal to CN1□-7, -8

| | |
|---|---|
| 0 | /COIN(/VCMP) |
| 1 | /TGON |
| 2 | /S-RDY |
| 3 | /CLT |
| 4 | /BK |
| 5 | PGC |
| 6 | OT |
| 7 | /RD |
| 8 | /HOME |
| 9 | /TCR |

H – – □ – Allocate signal to CN1□-1, -2

0 to 9: same as the allocation of CN1□-7, 8.

H – □ – – Allocate signal to CN1□-5, -6

0 to 9: same as the allocation of CN1□-7, 8.

H□ – – – Reserved

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn512 | Bus control input node lower bit enabled | 0000 to 1111 | – | 0000 | Immediately | P | S | – |

b – – – □ CN1□-10 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b – – □ – CN1□-11 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b – □ – – CN1□-12 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b□ – – – CN1□-13 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| Pn513 | Bus control input node higher bit enabled | 0000 to 1111 | – | 0000 | Immediately | P | S | – |

b – – – □ CN1□-14 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b – – □ – CN1□-15 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b – □ – – CN1□-16 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

b□ – – – CN1□-17 in bus selection

| | |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn514</td><td>Input port filter</td><td>0 to 1000</td><td>0.2ms</td><td>1</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">It is used to set input port filter time. The signal will be lagged if the parameter setting is too high.</td></tr>
<tr><td rowspan="2">Pn515</td><td>Alarm port filter</td><td>0 to 3</td><td>0.2ms</td><td>1</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">It is used to set alarm filter time. The signal will be lagged if the parameter setting is too high</td></tr>
<tr><td rowspan="2">Pn516</td><td>Input signal inversion 1</td><td>0000 to 1111</td><td>–</td><td>0000</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">
b – – – □ CN1□-10 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b – – □ – CN1□-11 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b – □ – – CN1□-12 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b□ – – – CN1□-13 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn517</td><td>Input signal inversion 2</td><td>0000 to 1111</td><td>–</td><td>0000</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">
b – – – □ CN1□-14 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b – – □ – CN1□-15 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b – □ – – CN1□-16 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
b□ – – – CN1□-17 inverse selection
<table>
<tr><td>0</td><td>Do not inverse</td></tr>
<tr><td>1</td><td>Inverse</td></tr>
</table>
</td></tr>
<tr><td rowspan="2">Pn518</td><td>Dynamic brake time</td><td>50 to 2000</td><td>0.5ms</td><td>125</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">–</td></tr>
<tr><td>Pn519</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td rowspan="2">Pn520</td><td>Position complete time</td><td>0 to 60000</td><td>0.25ms</td><td>200</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">–</td></tr>
<tr><td>Pn521</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>Pn522</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>Pn523</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>Pn524</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td rowspan="2">Pn525</td><td>Overload alarm threshold</td><td>100 to 150</td><td>%</td><td>100</td><td>Immediately</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">When load percentage is larger than overload alarm threshold, A04 will occur soon.<br>Pn525 is recommended to set below 120, otherwise the servo drive and motor will be damaged.</td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn526** | Reserved | – | – | – | – | – | – | – |
| **Pn527** | Reserved | – | – | – | – | – | – | – |
| **Pn528** | Output signal inversion | 0000 to 1111 | – | 0000 | Immediately | P | S | – |

b – – – □ CN1□-3, 4 inverse selection

| | |
|---|---|
| 0 | Do not inverse |
| 1 | Inverse |

b – – □ – CN1□-7, 8 inverse selection

| | |
|---|---|
| 0 | Do not inverse |
| 1 | Inverse |

b – □ – – CN1□-1, 2 inverse selection

| | |
|---|---|
| 0 | Do not inverse |
| 1 | Inverse |

b□ – – – CN1□-5, 6 inverse selection

| | |
|---|---|
| 0 | Do not inverse |
| 1 | Inverse |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Pn529** | Torque detection output signal threshold value | 3 to 300 | % | 100 | Immediately | P | S | T |
| | When motor torque output is higher than Pn529 setting value,/TCR is ON. When motor torque output is lower than Pn529 setting value,/TCR is OFF. | | | | | | | |
| **Pn530** | Torque detection output signal time | 1 to 1000 | ms | 100 | After restart | P | S | T |
| | Torque detection output signal time. | | | | | | | |
| **Pn600** | JPOS0 Position pulse in point to point control | -9999 to 9999 | 10000 Pulses | 0 | Immediately | P | – | – |
| | The two parameters Pn600 and Pn601 are used in combination, and the algebraic sum of them is the position JPOS0 needs to reach. (Thenumber of servomotor rotation revolutions is related with the programme mode of point to point control.) | | | | | | | |
| **Pn601** | JPOS0 Position pulse in point to point control | -9999 to 9999 | 1 Pulse | 0 | Immediately | P | – | – |
| | See the description of Pn600. | | | | | | | |

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn602 to Pn631** | Position pulse in point to point control from JPOS1 to JPOS15. | | | | | | | |
| **Pn632** | JPOS0 Point to point speed control | 0 to 6000 | rpm | 500 | Immediately | P | – | – |
| | – | | | | | | | |
| **Pn633 to Pn647** | Point to point speed control from JPOS0 to JPOS15. | | | | | | | |
| **Pn648** | JPOS0 Point to point 1st order filter | 0 to 32767 | 0.25ms | 0 | Immediately | P | – | – |
| | 1st order filter time of JPOS0 point to point control can stop or start the servomotor mildly. | | | | | | | |
| **Pn649 to Pn663** | Point to point 1st order filter from JPOS1 to JPOS15. | | | | | | | |
| **Pn664** | JPOS0 point to point control stop time | 0 to 300 | 50ms | 10 | Immediately | P | – | – |
| | – | | | | | | | |

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td>Pn665 to Pn679</td><td colspan="8">Point to point control stop time from JPOS1 to JPOS15.</td></tr>
<tr><td>Pn680</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td rowspan="5">Pn681</td><td>Function selection 681</td><td>0000 to 0133</td><td>–</td><td>0000</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">H – – – □ Single/cyclic, start/reference point selection
<table>
<tr><td>0</td><td>Cyclic operation, PCL start signal, NCL search reference point in forward direction.</td></tr>
<tr><td>1</td><td>Single operation, PCL start signal, NCL search reference point in forward direction.</td></tr>
<tr><td>2</td><td>Cyclic operation, NCL start operation, PCL search reference point in forward direction.</td></tr>
<tr><td>3</td><td>Single operation, NCL start operation, PCL search reference point in forward direction.</td></tr>
</table></td></tr>
<tr><td colspan="8">H – – □ – Change step and start mode
<table>
<tr><td>0</td><td>Delay to change step, no need of start signal, delay to start after S-ON.</td></tr>
<tr><td>1</td><td>PCON change step, no need of start signal, PCON delay to start after S-ON, but inside pulse can not stop when PCON off.</td></tr>
<tr><td>2</td><td>Delay to change step, need start signal, canceling start signal can immediately stop inside pulse. Return to programme start point process step when reset.</td></tr>
<tr><td>3</td><td>PCON change step, need start signal, canceling start signal can immediately stop inside pulse. Return to programme start point process step when reset.</td></tr>
</table></td></tr>
<tr><td colspan="8">H – □ – – Change step input signal mode
<table>
<tr><td>0</td><td>Change step input signal electrical level mode</td></tr>
<tr><td>1</td><td>Change step input signal pulse mode</td></tr>
</table></td></tr>
<tr><td colspan="8">H□ – – – Reserved</td></tr>
<tr><td rowspan="2">Pn682</td><td>Programming mode</td><td>0 to 1</td><td>–</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">
<table>
<tr><td>0</td><td>Incremental programming</td></tr>
<tr><td>1</td><td>Absolute programming</td></tr>
</table></td></tr>
<tr><td rowspan="2">Pn683</td><td>Programming start step</td><td>0 to 15</td><td>–</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">Select the start point of the point to point control</td></tr>
<tr><td rowspan="2">Pn684</td><td>Programming stop step</td><td>0 to 15</td><td>–</td><td>1</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="8">Select the stop point of the point to point control.</td></tr>
</table>

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td rowspan="2">Pn685</td><td>Speed during searching reference position</td><td>0 to 3000</td><td>rpm</td><td>1500</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">–</td></tr>
<tr><td rowspan="2">Pn686</td><td>Speed during leaving reference position</td><td>0 to 200</td><td>rpm</td><td>30</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">–</td></tr>
<tr><td rowspan="2">Pn687</td><td>Position teaching pulse</td><td>-9999 to 9999</td><td>10000 Pulses</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">The two parameters Pn687 and Pn688 are used in combination, and the algebraic sum of them is the current position of position teaching. When performing the position teaching by utility function, the algebraic sum of the two parameters are given to the current position</td></tr>
<tr><td rowspan="2">Pn688</td><td>Position teaching pulse</td><td>-9999 to 9999</td><td>1 Pulse</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">See the description of Pn687.</td></tr>
<tr><td rowspan="2">Pn689</td><td>Homing Mode Setting</td><td>0000 to 0111</td><td>–</td><td>0000</td><td>After restart</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">
b – – – □ Homing Mode<br>
0 Homing in the forward direction<br>
1 Homing in the reverse direction<br><br>
b – – □ – Search C-Pulse Mode<br>
0 Return to search C-Pulse when homing<br>
1 Directly search C-Pulse when homing<br><br>
b – □ – – Homing trigger starting mode<br>
0 Disabled<br>
1 Homing triggered by SHOM signal (rising edge)<br><br>
b□ – – – Reserved
</td></tr>
<tr><td rowspan="2">Pn690</td><td>Number of error pulses during homing</td><td>0 to 9999</td><td>10000 Pulses</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">–</td></tr>
<tr><td rowspan="2">Pn691</td><td>Number of error pulses during homing</td><td>0 to 9999</td><td>1 Pulse</td><td>0</td><td>Immediately</td><td>P</td><td>–</td><td>–</td></tr>
<tr><td colspan="7">–</td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn692** | Reserved | – | – | – | – | – | – | – |
| **Pn693** | Reserved | – | – | – | – | – | – | – |
| **Pn694** | Reserved | – | – | – | – | – | – | – |
| **Pn700** | MODBUS Setting | 0000 to 0182 | – | 0151 | After restart | P | S | – |

H – – – □ MODBUS communication baud rate

| | |
|---|---|
| 0 | 4800bps |
| 1 | 9600bps |
| 2 | 19200bps |
| 3 | 38400bps |
| 4 | 57600bps |
| 5 | 115200bps |

H – – □ – MODBUS protocol selection

| | |
|---|---|
| 0 | 7, N, 2 (Modbus,ASCII) |
| 1 | 7, E, 1 (Modbus,ASCII) |
| 2 | 7, O, 1 (Modbus,ASCII) |
| 3 | 8, N, 2 (Modbus,ASCII) |
| 4 | 8, E, 1 (Modbus,ASCII) |
| 5 | 8, O, 1 (Modbus,ASCII) |
| 6 | 8, N, 2 (Modbus,RTU) |
| 7 | 8, E, 1 (Modbus,RTU) |
| 8 | 8, O, 1 (Modbus,RTU) |

H – □ – – Reserved

H□ – – – Reserved

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Pn701** | MODBUS Axis address | 1 to 247 | – | 1 | After restart | P | S | – |
| | Axis address of MODBUS protocol communication | | | | | | | |

<table>
<tr><th>No.</th><th>Name</th><th>Range</th><th>Unit</th><th>Default</th><th>When Enabled</th><th colspan="3">Related Control</th></tr>
<tr><td>Pn702</td><td>Reserved</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td rowspan="2">Pn703</td><td>CAN communication speed</td><td>0000 to 0015</td><td>–</td><td>0004</td><td>After restart</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">
H – – – □ CAN communication baud rate
<table>
<tr><td>0</td><td>50Kbps</td></tr>
<tr><td>1</td><td>100Kbps</td></tr>
<tr><td>2</td><td>125Kbps</td></tr>
<tr><td>3</td><td>250Kbps</td></tr>
<tr><td>4</td><td>500Kbps</td></tr>
<tr><td>5</td><td>1Mbps</td></tr>
</table>
H – – □ – Reserved<br>
H – □ – – Reserved<br>
H□ – – – Reserved
</td></tr>
<tr><td rowspan="2">Pn704</td><td>CAN communication contact</td><td>1 to 127</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>S</td><td>–</td></tr>
<tr><td colspan="8">CANopen Aix address of communication</td></tr>
<tr><td rowspan="2">Pn705</td><td>1st electronic gear numerator in 32 bits (H)</td><td>0 to 9999</td><td>–</td><td>0</td><td>After restart</td><td>P</td><td>S</td><td>T</td></tr>
<tr><td colspan="8">The parameters are valid,when Pn009.2=1.<br>The electronic gear enables the reference pulse to relate with the servomotor travel distance, so the host controller doesn't change the mechanical deceleration ratio and encoder pulses. In fact, it is the setting of frequency doubling or frequency division to the reference pulses.<br>$$\frac{Numerator\ (Pn705 \times 10000 + Pn706\ or\ Pn709 \times 10000 + Pn710)}{Denominator\ (Pn707 \times 10000 + Pn708)}$$</td></tr>
<tr><td rowspan="2">Pn706</td><td>1st electronic gear numerator in 32 bits (L)</td><td>0 to 9999</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>S</td><td>T</td></tr>
<tr><td colspan="8">See the description of Pn705.</td></tr>
<tr><td rowspan="2">Pn707</td><td>Electronic gear denominator in 32 bits (H)</td><td>0 to 9999</td><td>–</td><td>0</td><td>After restart</td><td>P</td><td>S</td><td>T</td></tr>
<tr><td colspan="8">See the description of Pn705.</td></tr>
<tr><td rowspan="2">Pn708</td><td>Electronic gear denominator in 32 bits (L)</td><td>0 to 9999</td><td>–</td><td>1</td><td>After restart</td><td>P</td><td>S</td><td>T</td></tr>
<tr><td colspan="8">See the description of Pn705.</td></tr>
</table>

| No. | Name | Range | Unit | Default | When Enabled | Related Control | | |
|---|---|---|---|---|---|---|---|---|
| **Pn709** | 2nd electronic gear numerator in 32 bits (H) | 0 to 9999 | – | 0 | After restart | P | S | T |
| | See the description of Pn705. | | | | | | | |
| **Pn710** | 2nd electronic gear numerator in 32 bits (L) | 0 to 9999 | – | 1 | After restart | P | S | T |
| | See the description of Pn705. | | | | | | | |
| **Pn840** | Encoder Function | 0003 to 0F3F | – | 0003 | After restart | P | S | – |

H – – – □ Encoder model selection

| | |
|---|---|
| 3 to F | Reserved for manufacturer |

H – – □ – Motor designing sequence

| | |
|---|---|
| 0 to 3 | Reserved for manufacturer |

H – □ – – Power level of Machine

| | |
|---|---|
| 0 | 200W |
| 1 | 400W |
| 2 | 750W |
| 3 | 1000W |
| E | 50W |
| F | 100W |

H□ – – – Reserved

# Appendix B Alarms List

| Alarm No. | Output | Alarm Name | Description |
|---|---|---|---|
| A．01 | × | Parameter breakdown | The checksum results of parameters are abnormal. |
| A．03 | × | Overspeed | The servomotor speed is excessively high and the servomotor is out of control. |
| A．04 | × | Overload | The servomotor is operating continuously under a torque largely exceeding ratings. |
| A．05 | × | Position deviation counter overflow | Internal counter overflow |
| A．06 | × | Position error pulse overflow | Position error pulse exceededparameter (Pn504) |
| A．07 | × | The setting of electronic gear or given pulse frequency is not reasonable. | The setting of electronic gear is not reasonable or the given pulse frequency is too high. |
| A．08 | × | The 1st channel of current detection is wrong. | Something wrong with the inside chip of the 1st channel. |
| A．09 | × | The 2nd channel of current detection is wrong. | Something wrong with the inside chip of the 2nd channel. |
| A．10 | × | Incremental Encoder is break off. | At least one of Incremental Encoder PA,PB,PC is broken off. |
| A．12 | × | Overcurrent | An overcurrent flowed through the IPM. |
| A．13 | × | Overvoltage | Main circuit voltage for servomotor rotation is excessively high. |
| A．14 | × | Undervoltage | Main circuit voltage for servomotor rotation is excessively low. |
| A．15 | × | Bleeder resistor error | Bleeder resistor is faulty. |
| A．16 | × | Regeneration error | Regenerative circuit error |
| A．20 | × | Power line phase shortage | One phase does not bring into main circuit power supply. |
| A．25 | × | Motor power line U over current | Mechanical stuck or motor power line U phase sequence is wrong |
| A．26 | × | Motor power line V over current | Mechanical stuck or motor power line V phase sequence is wrong |
| A．27 | × | Motor power line W over current | Mechanical stuck or motor power line W phase sequence is wrong |
| A．38 | × | Photosensitive component of the encoder error | Photoelectric encoder was damaged. |
| A．39 | × | EEPROM in the encoder error | Encoder was damaged. |
| A．42 | × | Servomotor model error | The parameter setting of servo drive does not match the servomotor. |
| A．43 | × | Servo drive type error | The parameter setting of servo drive does not match the servomotor. |
| A．45 | × | Absolute encoder multiturn information error | Absolute encoder multiturn information is faulty. |
| A．46 | × | Absolute encoder multiturn information overflow | Absolute encoder multiturn information overflow. |

| Alarm No. | Output | Alarm Name | Description |
|---|---|---|---|
| A．47 | ╳ | Battery voltage below 2.5V | Absolute encoder multiturn information is lost. |
| A．48 | ╳ | Battery voltage below 3.1V | Battery voltage is too low. |
| A．50 | ╳ | Serial encoder communication overtime | Encoder disconnected; encoder signal disturbed; encoder error or encoder decoding circuit error. |
| A．51 | ╳ | Absolute encoder overspeed alarm detected | Absolute encoder multiturn information may be faulty.<br>Error reasons:<br>1. The battery is not connected or the battery voltage is insufficient.<br>2. The power supply to servo drive is not turned ON when the battery voltage is normal, or the servomotor running acceleration is too high due to external reason. |
| A．52 | ╳ | Absolute state of serial encoder error | Encoder or the encoder decoding circuit is faulty. |
| A．53 | ╳ | Serial encoder calcaution error | Encoder or the encoder decoding circuit is faulty. |
| A．54 | ╳ | Parity bit or end bit in serial encoder control domain error | Encoder signal is disturbed or the encoder decoding circuit is faulty. |
| A．55 | ╳ | Serial encoder communication data checking error | Encoder signal is disturbed or the encoder decoding circuit is faulty. |
| A．56 | ╳ | End bit in serial encoder control domain error | Encoder signal is disturbed or the encoder decoding circuit is faulty. |
| A．58 | ╳ | Serial encoder data empty | The EEPROM data of serial encoder is empty. |
| A．59 | ╳ | Serial encoder data format error | The EEPROM data format of serial encoder is incorrect. |
| A．66 | ╳ | CAN communication abnormal | CAN communication is faulty because of abnormal communication connection or disturbance. |
| A．67 | ╳ | Receiving heartbeat timeout | The master station sends heartbeat time timeout. |
| A．68 | ╳ | CAN Synchronization frame interval is too short | The filling time and the cycle of the synchronous frame does not matchor communication is faulty. |
| A．69 | ╳ | CAN Synchronization frame interval is too long | The filling time and the cycle of the synchronous frame does not match or communication is faulty. |
| A．00 | ○ | Not an error | Normal operation status. |

○: Output transistor is ON; ╳: Output transistor is OFF.

# Appendix C Object dictionary

| Index | Subindex | Object | Name | Type | Attr. | PDO | Support | | | | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | All | IP | PP | PV | PT | HM | |
| 2 | -- | VAR | od_integer8 | INT8 | RW | YES | ● | | | | | | |
| 3 | -- | VAR | od_integer16 | INT16 | RW | YES | ● | | | | | | |
| 4 | -- | VAR | od_integer32 | INT32 | RW | YES | ● | | | | | | |
| 5 | -- | VAR | od_unsigned8 | UINT8 | RW | YES | ● | | | | | | |
| 6 | -- | VAR | od_unsigned16 | UINT16 | RW | YES | ● | | | | | | |
| 7 | -- | VAR | od_unsigned32 | UINT32 | RW | YES | ● | | | | | | |
| 1000 | -- | VAR | device_type | UINT32 | RO | NO | ● | | | | | | |
| 1001 | -- | VAR | error_register | UINT8 | RO | NO | ● | | | | | | |
| 1003 | -- | VAR | pre_defined_error_field | UINT8 | RW | NO | ● | | | | | | |
| 1005 | -- | VAR | cob_id_sync | UINT32 | RW | NO | ● | | | | | | |
| 1006 | -- | VAR | communication_cycle_period | UINT32 | RW | NO | ● | | | | | | |
| 1007 | -- | VAR | synchronous_window_length | UINT32 | RW | NO | ● | | | | | | |
| 1014 | -- | VAR | cob_id_emergency_message | UINT32 | RW | NO | ● | | | | | | |
| 1016 | -- | ARRAY | consumer_heartbeat_time | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | consumer_heartbeat_time1 | UINT32 | RW | NO | ● | | | | | | |
| 1017 | | VAR | producer_heartbeat_time | UINT16 | RW | NO | ● | | | | | | |
| 1018 | -- | RECORD | identity_object | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | vendor_id | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | product_code | UINT32 | RO | NO | ● | | | | | | |
| | 3 | | revision_number | UINT32 | RO | NO | ● | | | | | | |
| | 4 | | serial_number | UINT32 | RO | NO | ● | | | | | | |
| 1029 | -- | ARRAY | error_behaviour | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | communication_error | UINT8 | RW | NO | ● | | | | | | |
| 1200 | -- | RECORD | server_sdo_parameter | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | cob_id_client_server | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | cob_id_server_client | UINT32 | RO | NO | ● | | | | | | |
| 1400 | -- | RECORD | receive_pdo_parameter_rpdo1 | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries_rpdo1 | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | cob_id_used_by_pdo_rpdo1 | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | transmission_type_rpdo1 | UINT8 | RW | NO | ● | | | | | | |
| 1401 | -- | RECORD | receive_pdo_parameter_rpdo2 | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries_rpdo2 | UINT8 | RO | NO | ● | | | | | | |

| Index | Subindex | Object | Name | Type | Attr. | PDO | Support | | | | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | All | IP | PP | PV | PT | HM | |
| | 1 | | cob_id_used_by_pdo_rpdo2 | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | transmission_type_rpdo2 | UINT8 | RW | NO | ● | | | | | | |
| 1402 | -- | RECORD | receive_pdo_parameter_rpdo3 | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries_rpdo3 | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | cob_id_used_by_pdo_rpdo3 | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | transmission_type_rpdo3 | UINT8 | RW | NO | ● | | | | | | |
| 1403 | -- | RECORD | receive_pdo_parameter_rpdo4 | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries_rpdo4 | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | cob_id_used_by_pdo_rpdo4 | UINT32 | RO | NO | ● | | | | | | |
| | 2 | | transmission_type_rpdo4 | UINT8 | RW | NO | ● | | | | | | |
| 1600 | -- | RECORD | receive_pdo_mapping_rpdo1 | -- | -- | -- | ● | | | | | | |
| | 0 | | number_of_entries | UINT8 | RO | NO | ● | | | | | | |
| | 1 | | first_mapped_object_rpdo1 | UINT32 | RW | NO | ● | | | | | | |
| | 2 | | second_mapped_object_rpdo1 | UINT32 | RW | NO | ● | | | | | | |
| | 3 | | third_mapped_object_rpdo1 | UINT32 | RW | NO | ● | | | | | | |
| | 4 | | fourth_mapped_object_rpdo1 | UINT32 | RW | NO | ● | | | | | | |


南京埃斯顿自动控制技术有限公司

地址：南京江宁经济技术开发区水阁路16号　邮编：211106
电话：025-52785866
传真：025-52785576
公司主页：www.estun.com
电子信箱：info@estun.com

www.estun.com

ESTUN