CANopen Mode 11

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

11.1 Basic configuration ····· 11-2
11.1.1 Supported functions ····· 11-2
11.1.2 Hardware configuration ····· 11-3
11.1.3 Parameter settings in CANopen mode ····· 11-4
11.2 Communication specification ····· 11-5
11.2.1 Servo communication architecture ····· 11-5
11.2.2 Communication objects ····· 11-6
11.2.2.1 Process data object (PDO) ····· 11-7
11.2.2.2 Service data object (SDO) ····· 11-8
11.2.2.3 SDO abort codes ····· 11-11
11.2.2.4 Synchronization object (SYNC) ····· 11-12
11.2.2.5 Emergency object (EMCY) ····· 11-13
11.2.2.6 NMT services ····· 11-14
11.3 CANopen operation modes ····· 11-17
11.3.1 Profile Position mode ····· 11-17
11.3.2 Interpolated Position mode ····· 11-22
11.3.3 Homing mode ····· 11-25
11.3.4 Profile Velocity mode ····· 11-27
11.3.5 Profile Torque mode ····· 11-29
11.4 Object dictionary ····· 11-31
11.4.1 Specifications for objects ····· 11-31
11.4.2 List of objects ····· 11-32
11.4.3 Details of objects ····· 11-34
11.4.3.1 OD 1XXXh communication object group ····· 11-34
11.4.3.2 OD 2XXXh servo parameter group ····· 11-53
11.4.3.3 OD 6XXXh communication object group ····· 11-54
11.5 Diagnostics and troubleshooting ····· 11-96

11

11.1 Basic configuration

11.1.1 Supported functions

CANopen functions supported by Delta servo drives:

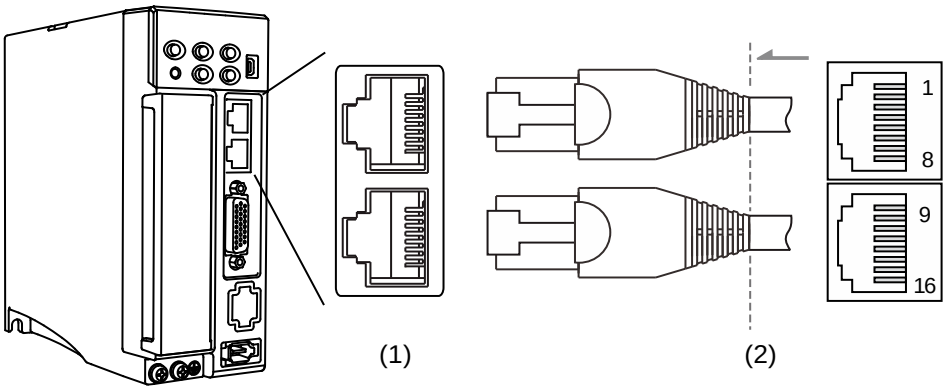
- CANopen communication objects: NMT, SYNC, SDO, PDO, and EMCY.
- SDO transmission: acyclic data exchange for reading / writing parameters and communication related settings.
- PDO transmission / reception: time-triggered, event-triggered, synchronous transmission (cyclic), and asynchronous transmission (acyclic).
- Node guarding.
- Heartbeat.

CANopen function not supported by Delta servo drives:

- Time stamp.

11.1.2 Hardware configuration

Pin assignment (RJ-45) for CAN bus wiring



(1) CN3 connector (female); (2) CN3 connector (male)

Pin assignment:

Pin No.	Signal	Description
1, 9	CAN_H	CAN_H bus line (dominant high)
2, 10	CAN_L	CAN_L bus line (dominant low)
3, 11	GND_ISO	Signal GND
4, 12	RS-485-	For the servo drive to transmit the data to differential terminal (-).
5, 13	RS-485+	For the servo drive to transmit the data to differential terminal (+).
6, 14	-	Reserved
7, 15	GND_ISO	Signal GND
8, 16	-	Reserved

■ Baud rate setting

Baud rate and bus length

Baud rate	Maximum bus length
1 Mbps	25 m (82 ft)
800 Kbps	50 m (164 ft)
500 Kbps (default)	100 m (328 ft)
250 Kbps	250 m (820 ft)
125 Kbps	500 m (1640 ft)

11

11.1.3 Parameter settings in CANopen mode

Follow these instructions to connect the CANopen controller and the servo drive:

1. Set to CANopen mode: set P1.001.YX to 0C.
2. Set the node ID: set P3.000 to 0x0001 - 0x007F.
3. Set the transmission rate (baud rate): set P3.001.Z to 4
   (Z = 0: 125 Kbps; 1: 250 Kbps; 2: 500 Kbps; 3: 800 Kbps; 4: 1 Mbps).
4. It is suggested that you change the setting value of P3.012.Z from 0 (default) to 1 to enable the non-volatile setting for the parameter. Note that the default E-Gear ratio varies with the set value of P3.012.Z.

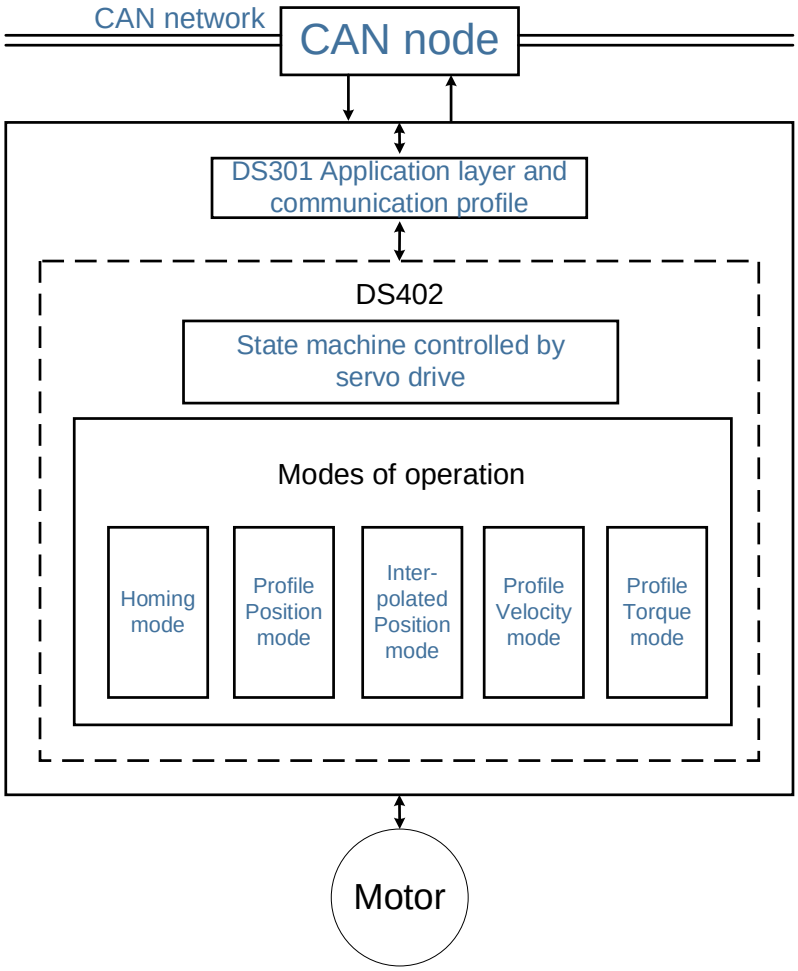
Function	P3.012 = 0x0100 (Z = 1)		P3.012 = 0x0000 (Z = 0)	
	Servo parameter	Default	OD address	Default
Motor stop mode	P1.032	0x0000	605Bh	0
S-curve acceleration constant	P1.034	200	6087h	200
Zero speed range	P1.038	100 (0.1 rpm)	606Fh	100 (0.1 rpm)
E-Gear ratio - numerator N1	P1.044	16777216	6093h sub1	1
E-Gear ratio - denominator M	P1.045	100000	6093h sub2	1
Speed reached (DO.SP_OK) range	P1.047	10 (rpm)	606Dh	100 (0.1 rpm)
Accumulated time to reach desired speed	P1.049	0	606Eh	0
Maximum speed limit	P1.055	Depending on the motor (rpm)	607Fh	Depending on the motor (0.1 rpm)
			6080h	Depending on the motor (rpm)
Excessive deviation warning condition of Position command	P2.035	50331648	6065h	50331648
Positive software limit (PP / CSP / CSV / CST mode)	P5.008	2147483647	607Dh sub2	2147483647
Negative software limit (PP / CSP / CSV / CST mode)	P5.009	-2147483648	607Dh sub1	-2147483648
Origin definition (HM mode)	P6.001	0	607Ch	0

5. It is suggested that you enable the dynamic brake function (P1.032 = 0x0000).

11.2 Communication specification

11.2.1 Servo communication architecture

The CANopen architecture of the servo drive is as follows:



- DS301 is the communication profile. This protocol includes the communication objects (PDO, SDO, SYNC, and Emergency object), NMT service, and related communication object dictionary.
- DS402 is the device profile for drives and motion control. It defines the behavior of each operation mode and the required object index settings for execution.

11

11.2.2 Communication objects

The default values of the Delta servo drive object dictionary comply with the DS301 protocol. All CANopen data contains an 11-bit identifier, generally referred to as "COB-ID". The COB-ID data format is as follows:

Bit	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 6	Node-ID	The data size is 7-bit and the setting range is 1 - 127.
Bit 7 - Bit 10	Function code	The data size is 4-bit and the setting range is 0 - 15.

The following table lists the supported objects and the corresponding COB-IDs:

Communication object	Function code Bit 10 9 8 7	Node ID Bit 6 5 4 3 2 1 0	COB-ID DEC (HEX)	Object index
NMT service	0 0 0 0	0 0 0 0 0 0 0	0 (0h)	-
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80h)	1005h, 1006h
EMCY object	0 0 0 1	X X X X X X X	128 (80h) + Node-ID	1014h
TxPDO1	0 0 1 1	X X X X X X X	384 (180h) + Node-ID	1800h
RxPDO1	0 1 0 0	X X X X X X X	512 (200h) + Node-ID	1400h
TxPDO2	0 1 0 1	X X X X X X X	640 (280h) + Node-ID	1801h
RxPDO2	0 1 1 0	X X X X X X X	768 (300h) + Node-ID	1401h
TxPDO3	0 1 1 1	X X X X X X X	896 (380h) + Node-ID	1802h
RxPDO3	1 0 0 0	X X X X X X X	1024 (400h) + Node-ID	1402h
TxPDO4	1 0 0 1	X X X X X X X	1152 (480h) + Node-ID	1803h
RxPDO4	1 0 1 0	X X X X X X X	1280 (500h) + Node-ID	1403h
TxSDO	1 0 1 1	X X X X X X X	1408 (580h) + Node-ID	1200h
RxSDO	1 1 0 0	X X X X X X X	1536 (600h) + Node-ID	1200h
NMT error control	1 1 1 0	X X X X X X X	1792 (700h) + Node-ID	1016h, 1017h

Note: 0 indicates the bit is off, 1 indicates the bit is on, and X indicates the bit is set according to the requirement.

Communication object dictionary:

Communication object index	Object area
1000h - 1FFFh	Communication Profile Area
2000h - 2FFFh	Manufacturer Specific Profile Area
6000h - 9FFFh	Standardized Device Profile Area

11.2.2.1 Process data object (PDO)

Real-time data transmission can be achieved with Process data objects (PDOs). There are two types of PDOs: transmit PDOs (TxPDOs) and receive PDOs (RxPDOs). This definition is from the perspective of the servo drive, for example, the TxPDO refers to the object that the servo drive sends to the controller. Set the communication parameters and mapping parameters as shown in the following table to use the PDOs.

RxPDOs

Communi-cation object	Communication object index	Mapping object index
RxPDO1	1400h	1600h
RxPDO2	1401h	1601h
RxPDO3	1402h	1602h
RxPDO4	1403h	1603h

TxPDOs

Communi-cation object	Communication object index	Mapping object index
TxPDO1	1800h	1A00h
TxPDO2	1801h	1A01h
TxPDO3	1802h	1A02h
TxPDO4	1803h	1A03h

The format of PDO mapping parameter is:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function
Bit 0 - Bit 7	Object data length
Bit 8 - Bit 15	Object sub-index
Bit 16 - Bit 31	Object index

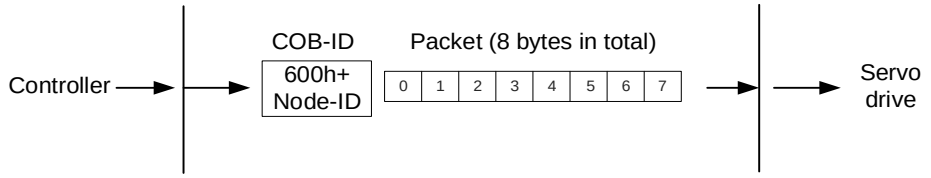
Example:

To set the three PDOs, OD 6040h, OD 607Ah, and OD 6060h, in the first group of PDOs, the setting is as follows:

Mapping parameter setting for RxPDO	Data			Description
OD 1600h sub0	3			Set 3 PDO mappings.
OD 1600h sub1	6040h	00h	10h	Mapping the Controlword (OD 6040h); data length is 16-bit.
OD 1600h sub2	607Ah	00h	20h	Mapping the target position (OD 607Ah); data length is 32-bit.
OD 1600h sub3	6060h	00h	08h	Mapping the operation mode (OD 6060h); data length is 8-bit.
Note	The total length is 38h (56-bit) which meets the specification of less than 64-bit.			

11.2.2.2 Service data object (SDO)

With Service data objects (SDOs), you can write or read objects. The SDO message format is mainly composed of COB-ID and SDO packets. SDO packets can transmit up to 4 bytes.

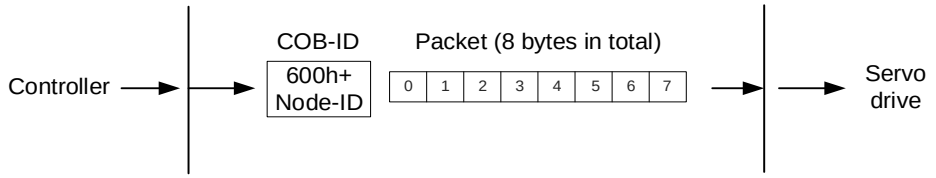


Byte	Function
Byte 0	Command code
Byte 1 - Byte 2	Object index
Byte 3	Object sub-index
Byte 4 - Byte 7	Data

■ Write data with SDO

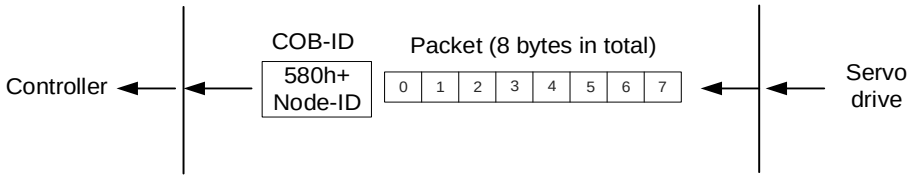
To use an SDO to write data with the controller, you need to write the command code, indexes, and data according to the SDO format. The servo drive then returns the corresponding message based on the written data.

The following figure shows the packet format when the controller sends the SDO for writing data:



Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
23h	-	-	-	Data				Write 4 bytes of data.
2Bh	-	-	-	Data				Write 2 bytes of data.
2Fh	-	-	-	Data				Write 1 byte of data.

The following figure shows the packet format returned by the servo drive when the controller sends the SDO for writing data:



Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
60h	-	-	-					Write-in is successful.
80h	-	-	-	SDO abort codes				Error code.

Note: for SDO abort codes, refer to Section 11.2.2.3.

Example:

Write the value of 300,000 (493E0h) to the servo parameter P7.001 (OD 2701h).

The write-in format is as follows:

Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
23h	01	27	0	E0	93	04	00	Write 4 bytes of data.

The returned packet is as follows:

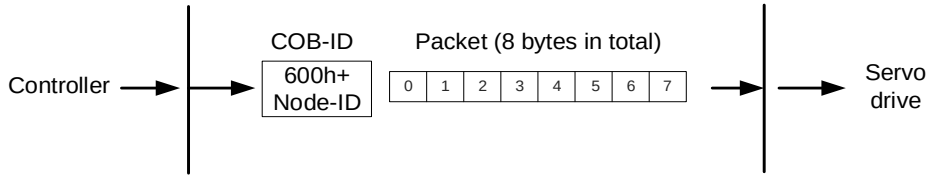
Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
60h	01	27	0					Write-in is successful.

11

■ Read data with SDO

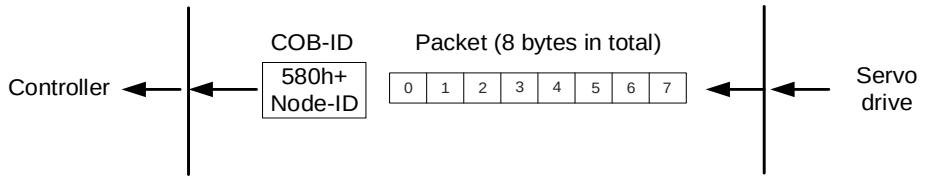
To use an SDO to read data with the controller, you need to write the command code and indexes according to the SDO format. The servo drive then returns the object's data based on the object to be read.

The following figure shows the packet format when the controller sends the SDO for reading data:



Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
40h	-	-	-					Read data.

The following figure shows the packet format returned by the servo drive when the controller sends the SDO for reading data:



Command code	Object index		Object sub-index	Data				Description
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
43h	-	-	-	Data				Read 4 bytes of data.
4Bh	-	-	-	Data				Read 2 bytes of data.
4Fh	-	-	-	Data				Read 1 byte of data.
80h	-	-	-	SDO abort codes				Error code.

Note: for SDO abort codes, refer to Section 11.2.2.3.

11.2.2.3 SDO abort codes

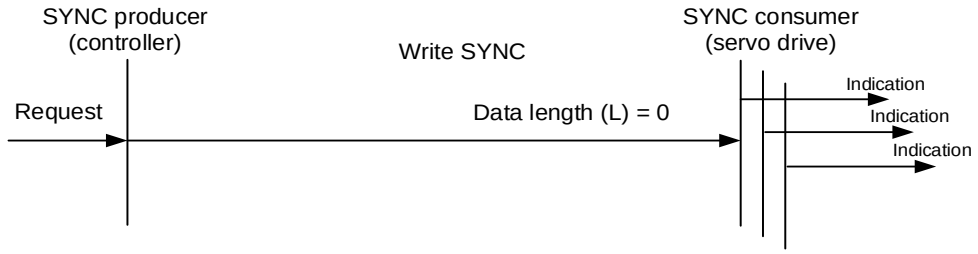
The abort codes are as follows:

SDO abort code	Description
05040001h	Client / server command is invalid or does not exist.
06010002h	Attempt to write a read-only object.
06020000h	Object does not exist in the object dictionary.
06040041h	Unable to map the object to the PDO.
06040042h	The number and length of mapped objects exceed the PDO length.
06060000h	Access failed due to hardware error (storage or restore error).
06070010h	Data type does not match; parameter length does not match.
06090011h	Sub-index does not exist.
06090030h	The written parameter value is out of range.
08000000h	General error.
080000a1h	An error occurred when an object is read from EEPROM.
080000a2h	An error occurred when an object is written to EEPROM.
080000a3h	Invalid range when accessing EEPROM.
080000a4h	EEPROM data content error occurred when EEPROM is accessed.
080000a5h	The entered password is incorrect when data is written to the encryption area.
08000020h	Unable to transfer data or save data to the application.
08000021h	Unable to transfer data or save data to the application due to restrictions (storage or restore in the wrong state).
08000022h	Object is in use.

11.2.2.4 Synchronization object (SYNC)

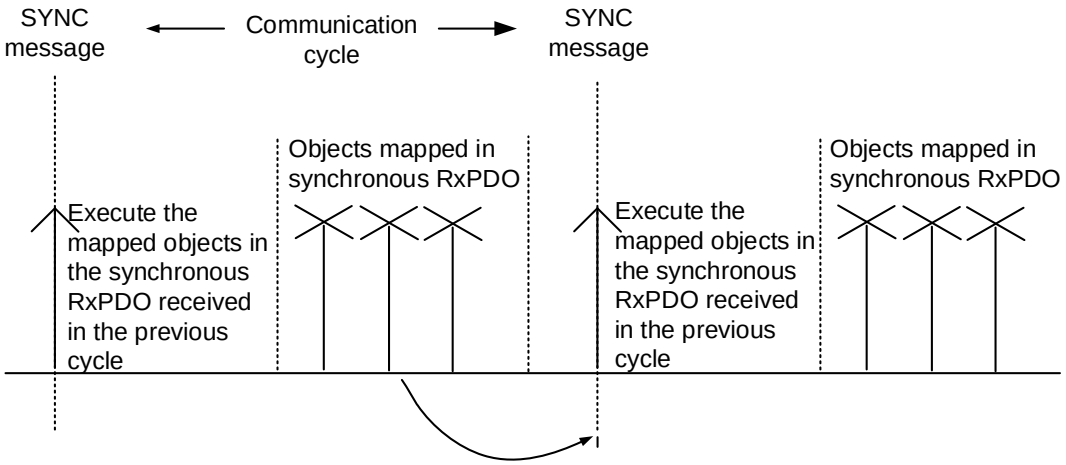
The Synchronization objects (SYNCs) are periodically broadcast by the SYNC producer. There is no data in the SYNC packet (L = 0).

The SYNC protocol is as follows:

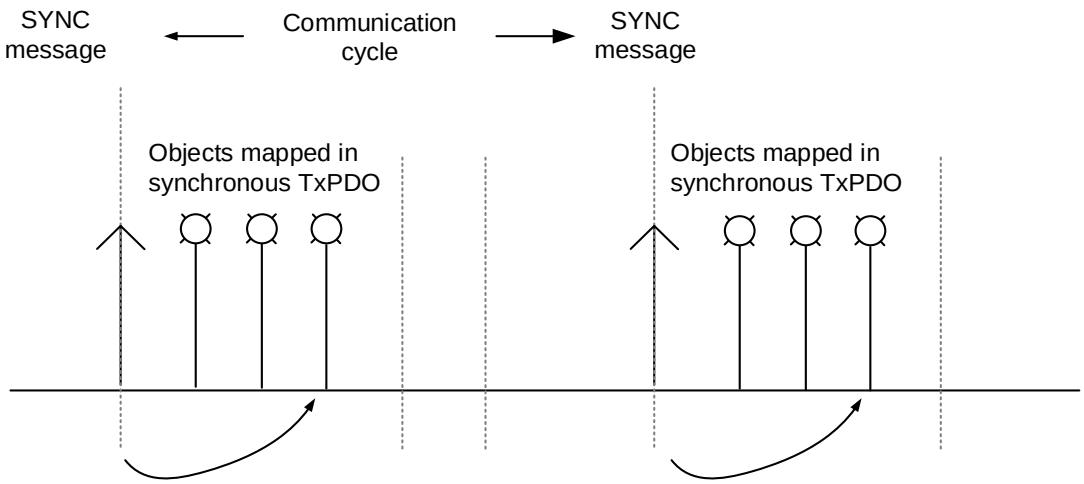


The SYNC object is used to achieve the synchronization of PDO transmission and reception between the controller and servo drive. The SYNC object transmission cycle is set by the object OD 1006h (see Section 11.4 Object dictionary for detailed settings).

The following figure shows the timing sequence between the servo drive RxPDO reception and the controller SYNC transmission. The controller transmits RxPDO to the servo drive between two SYNCs (communication cycle), and the servo drive will not execute the RxPDO received in the previous communication cycle until it receives the SYNC.

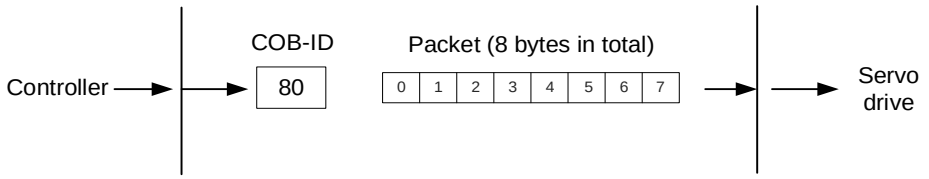


The following figure shows the timing sequence between the servo drive TxPDO transmission and the controller SYNC transmission. The servo drive transmits the TxPDO data to the controller as soon as it receives the SYNC.



11.2.2.5 Emergency object (EMCY)

When the servo detects an abnormality, it sends an alarm and notifies the controller with the Emergency object. The Emergency object can transmit only one alarm at a time. When a higher priority alarm occurs before the previous lower priority alarm is cleared, the higher priority alarm overwrites the previous alarm and is transmitted to the controller as an Emergency object.



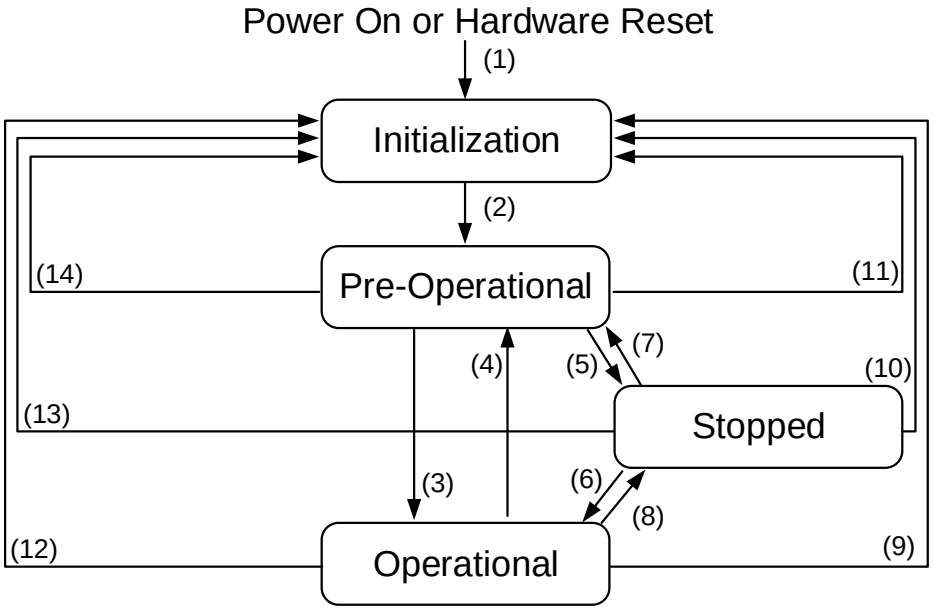
Error code		Error register	Servo alarm	N/A
Byte 0	Byte 1	Byte 2	Byte 3	Bytes 4 - 7
Refer to Section 12.5.2 Alarm list for details.		OD 1001h	Refer to Chapter 14 Troubleshooting for details.	

11.2.2.6 NMT services

11

■ State machine

The NMT state machine is shown as follows. After the servo drive completes the initialization, it enters the Pre-Operational state. The NMT state machine determines the behavior of the communication objects, such as PDO functions only in the Operational state.



Status	Description
Initialization	The servo drive successfully completes initialization after being powered on without errors occurring. The packets cannot yet be transmitted in this state.
Pre-Operational	Data can be exchanged with SDOs. If an alarm occurs in the servo drive, an emergency message is sent to notify the controller.
Stopped	The servo drive can use SDO and TxPDO data packets to exchange data with the controller.
Operational	All data exchanges, including SDOs and PDOs (TxPDOs and RxPDOs), are allowed.

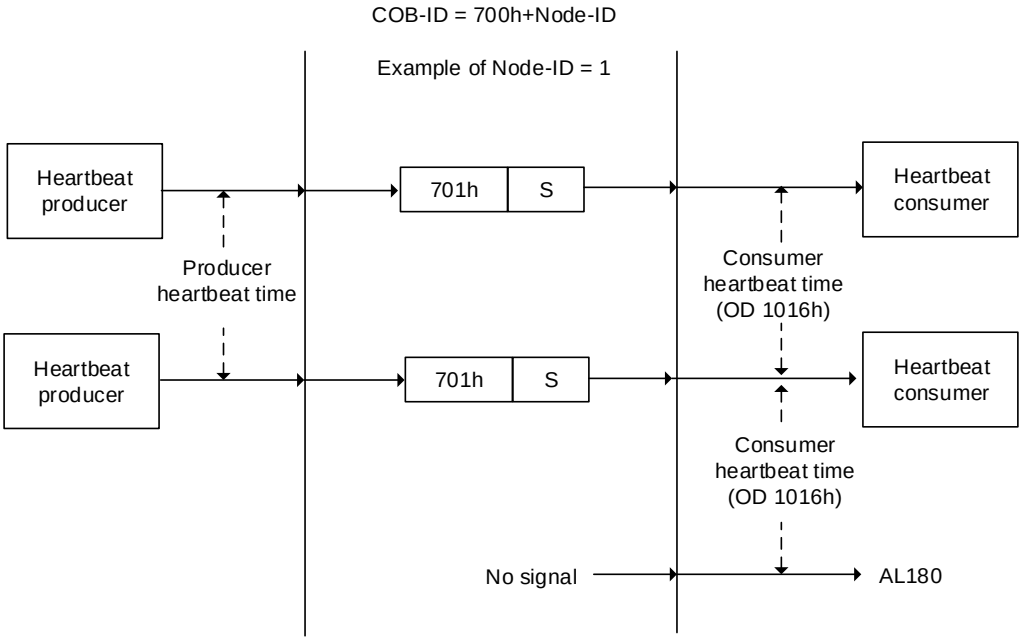
The following table shows the available communication objects in each communication state:

Communication object	Initialization	Pre-Operational	Operational	Stopped
PDO	-	-	V	TxPDO
SDO	-	V	V	V
Synchronization object	-	V	V	-
Emergency object	-	V	V	-
Boot-up object	V	-	-	-
NMT object	-	V	V	V

■ Heartbeat

The Heartbeat mechanism is mainly to enable the producer to send packets to the consumer periodically. The producer can be a controller or servo drive; on the other hand, a controller or servo drive can also be the consumer.

If you use the controller to send the heartbeat and the servo drive as the consumer, you need to set the consumer heartbeat time (OD 1016h) for the servo drive. When the servo drive does not receive the heartbeat signal within the receiving time, it triggers the heartbeat event, meaning AL180 is triggered. Consumer heartbeat time (OD 1016h) is defined as the time the servo drive expects to receive a heartbeat. To start the Heartbeat mechanism, set the consumer heartbeat time (OD 1016h) and then have the controller send the heartbeat signal. The consumer heartbeat time (OD 1016h) must be greater than the producer heartbeat time which is set by the controller. Since there are delays and other uncontrollable external factors in transmitting the heartbeat message, you must retain a tolerance time for the transmission.



The S code is described as follows:

S	State
0	Bootup
4	Stopped
5	Operational
127	Pre-Operational

If you want to use the servo drive as the producer, then the heartbeat is sent by the drive. When the controller does not receive the heartbeat signal within the receiving time, it triggers the heartbeat event which corresponds to the alarm defined by the controller.

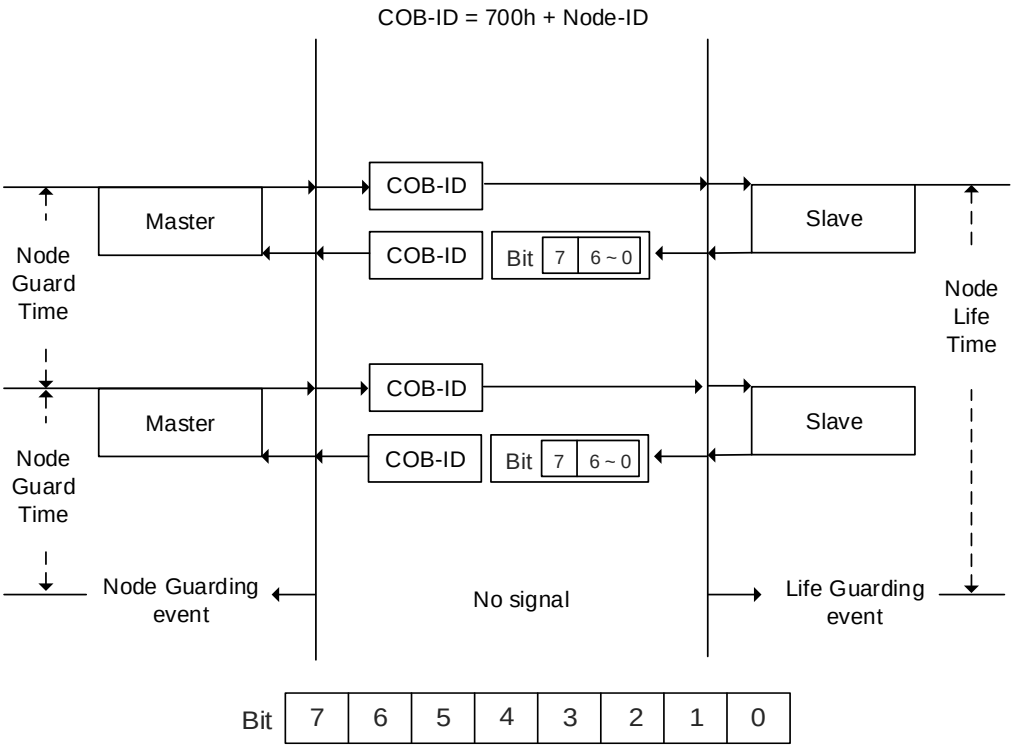
The servo drive can be the consumer and the producer simultaneously. In that case, you need to set OD 1016h and OD 1017h at the same time, and the controller must be set as the producer and the consumer as well.

11

■ Node/Life guarding

The Node/Life guarding mechanism is similar to the Heartbeat mechanism. The main difference between the two is that Heartbeat only uses the consumer but not the producer to judge whether there are packets or not. The mechanism of Node/Life guarding is mainly based on the two-way relationship between the master and slave. The master periodically sends packets to the slave, and the slave must return the packets to the master within the set guard time (OD 100Ch), otherwise an error occurs. You must set the life time for the slave and the master must send the packets within the guard time. If the slave does not receive the packets, AL180 is triggered. Life time is set by multiplying the guard time by a life time factor (OD 100Dh).

The Node/Life Guarding architecture is as follows:



Bit	Function	Description
Bit 0 - Bit 6	State of the NMT slave	4: Stopped 5: Operational 127: Pre-Operational
Bit 7	Reserved	-

11.3 CANopen operation modes

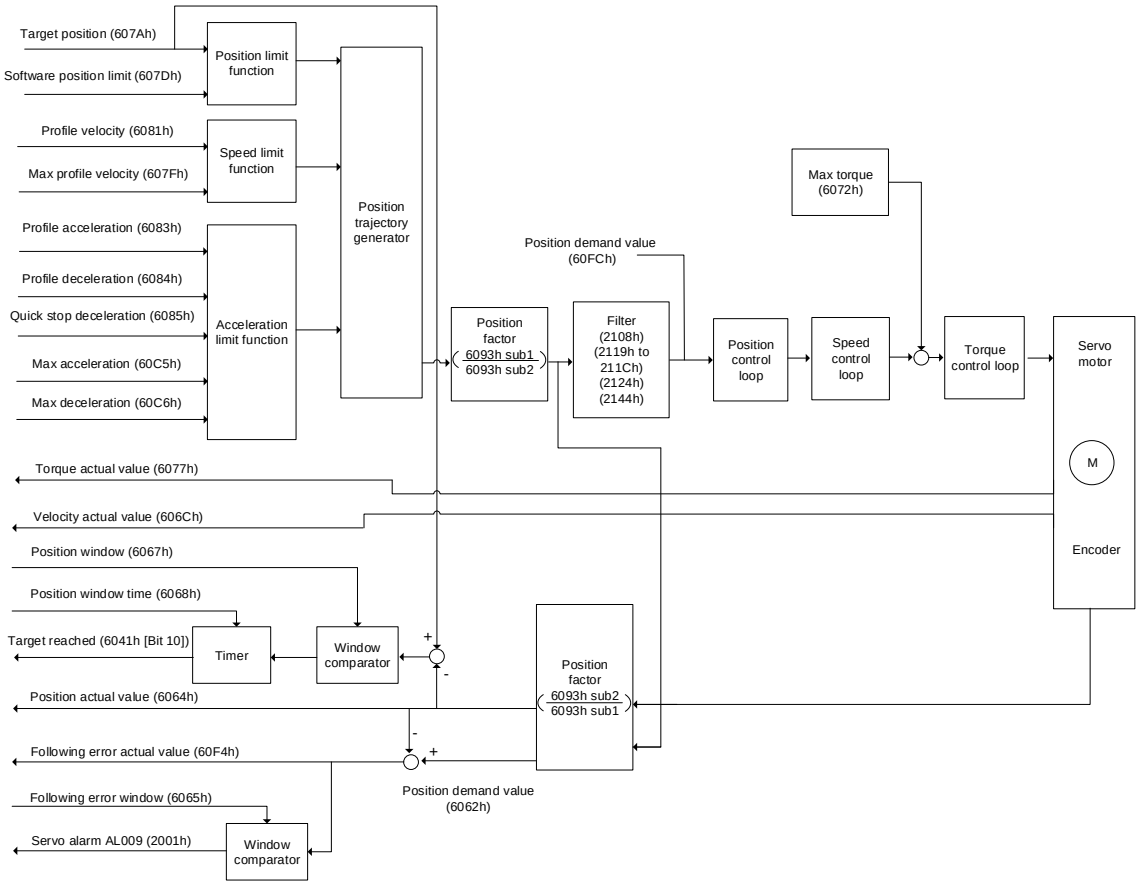
This section describes the modes of operation specified by CiA DS402 when the servo is in the CANopen mode. The content includes basic operation settings and related object descriptions.

11

11.3.1 Profile Position mode

After receiving the position command transmitted from the controller, the servo drive controls the servo motor to reach the target position. In Profile Position (PP) mode, the controller only informs the servo drive of the target position, speed command, and acceleration / deceleration settings at the beginning. The motion planning from command triggering to the arrival of the target position is performed by the trajectory generator in the servo drive.

The following figure shows the Profile Position mode architecture of the servo drive:



Operation steps:

1. Set OD 6060h to 01h to set the mode as Profile Position mode.
2. Set OD 607Ah for the target position (unit: PUU).
3. Set OD 6081h for the profile velocity (unit: PUU/sec).
4. Set OD 6083h for the profile acceleration (unit: ms).
5. Set OD 6084h for the profile deceleration (unit: ms).
6. Set the Controlword (OD 6040h). Follow these steps for operation. Steps 6.1 and 6.2 are to bring the servo drive's state machine into the ready state. For the description of the state machine, refer to the OD 6040h description in Section 11.4.3.3.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
6.1	0	0	1	1	0	Shutdown.
6.2	0	0	1	1	1	Switch on (ready for Servo On).
6.3	0	1	1	1	1	Enable operation (Servo On).
6.4	1	1	1	1	1	Command triggering (rising-edge triggered)

7. After the servo completes the first motion command, the servo sets the target position, speed, and other conditions to execute the next motion command.
8. Set the Controlword (OD 6040h). Since the command is rising-edge triggered, switch Bit 4 to Off first and then to On.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
8.1	0	1	1	1	1	Enable operation (Servo On).
8.2	1	1	1	1	1	Command triggering (rising-edge triggered)

Read the servo drive information:

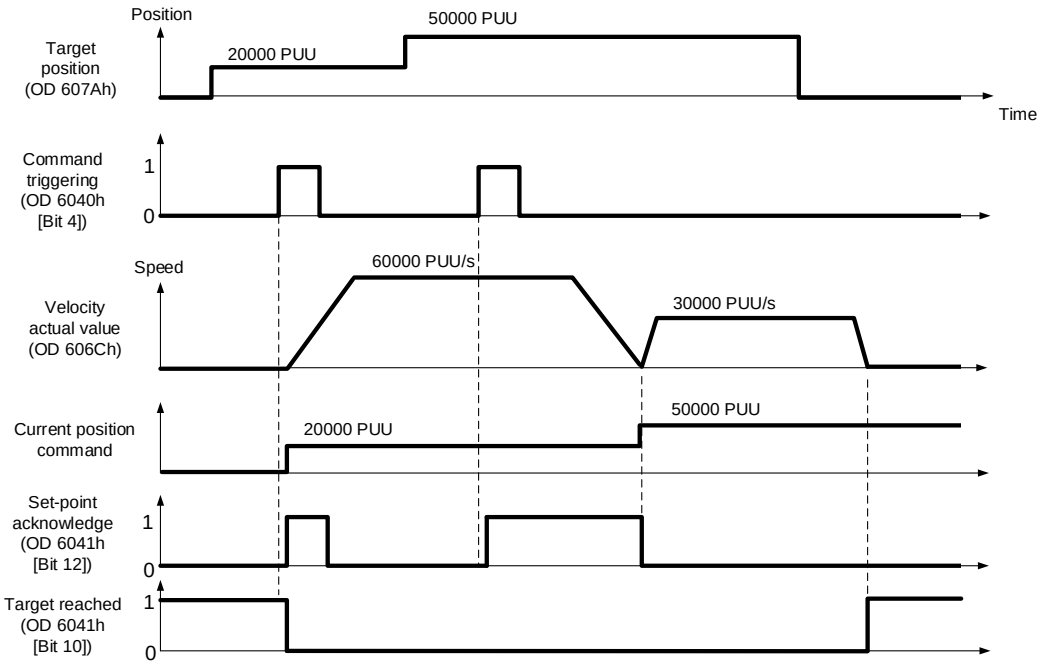
1. Read OD 6064h to obtain the actual value of the motor position at present.
2. Read OD 6041h to obtain the servo drive status, including the following error and notifications for set-point acknowledge and target reached.

Function for the command to take immediate effect

In Profile Position mode, set the command to take effect immediately or not with OD 6040h [Bit 5].

- Set OD 6040h [Bit 5] to 0 to disable the command from taking immediate effect

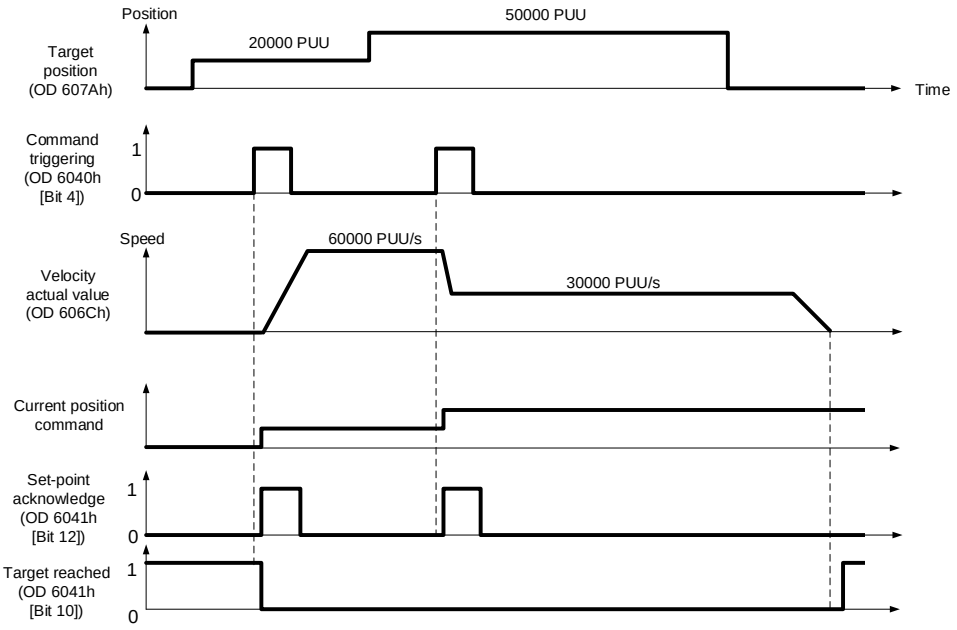
If the command is not enabled to take immediate effect, when the current motion command is in execution (not yet complete), the servo continues to execute the current motion command even if a new command is triggered. The new command is acknowledged and executed only after the current command is complete.



- Set OD 6040h [Bit 5] to 1 to enable the command to take immediate effect (only valid in Profile Position mode)

11

If the command is enabled to take immediate effect, when the current motion command is in execution (not yet complete), the servo immediately interrupts the current command and executes the new command once receiving the new triggered command.



Relevant object list

Index	Name	Data type	Access
6040h	Controlword	UNSIGNED16	RW
6041h	Statusword	UNSIGNED16	RO
6060h	Modes of operation	INTEGER8	RW
6061h	Modes of operation display	INTEGER8	RO
6062h	Position demand value [PUU]	INTEGER32	RO
6063h	Position actual internal value [Pulse]	INTEGER32	RO
6064h	Position actual value [PUU]	INTEGER32	RO
6065h	Following error window	UNSIGNED32	RW
6067h	Position window	UNSIGNED32	RW
6068h	Position window time	UNSIGNED16	RW
606Ch	Velocity actual value	INTEGER32	RO
6072h	Max torque	UNSIGNED16	RW
6077h	Torque actual value	INTEGER16	RO
607Ah	Target position	INTEGER32	RW
607Dh	Software position limit	INTEGER32	RW
607Fh	Max profile velocity	UNSIGNED32	RW
6081h	Profile velocity	UNSIGNED32	RW
6083h	Profile acceleration	UNSIGNED32	RW
6084h	Profile deceleration	UNSIGNED32	RW
6085h	Quick stop deceleration	UNSIGNED32	RW
6093h	Position factor	UNSIGNED32	RW
60C5h	Max acceleration	UNSIGNED32	RW

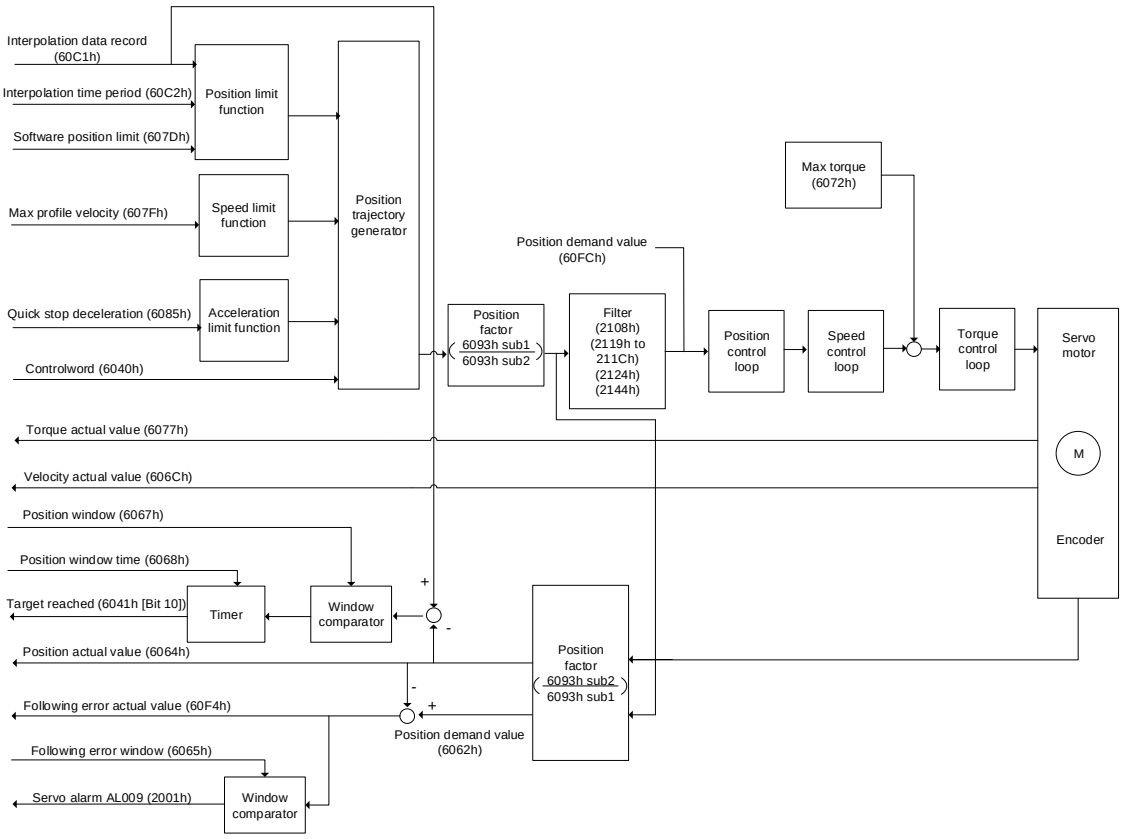
Index	Name	Data type	Access
60C6h	Max deceleration	UNSIGNED32	RW
60F4h	Following error actual value	INTEGER32	RO
60FCh	Position demand value	INTEGER32	RO

Note: for more details, refer to Section 11.4.3 Details of objects.

11

11.3.2 Interpolated Position mode

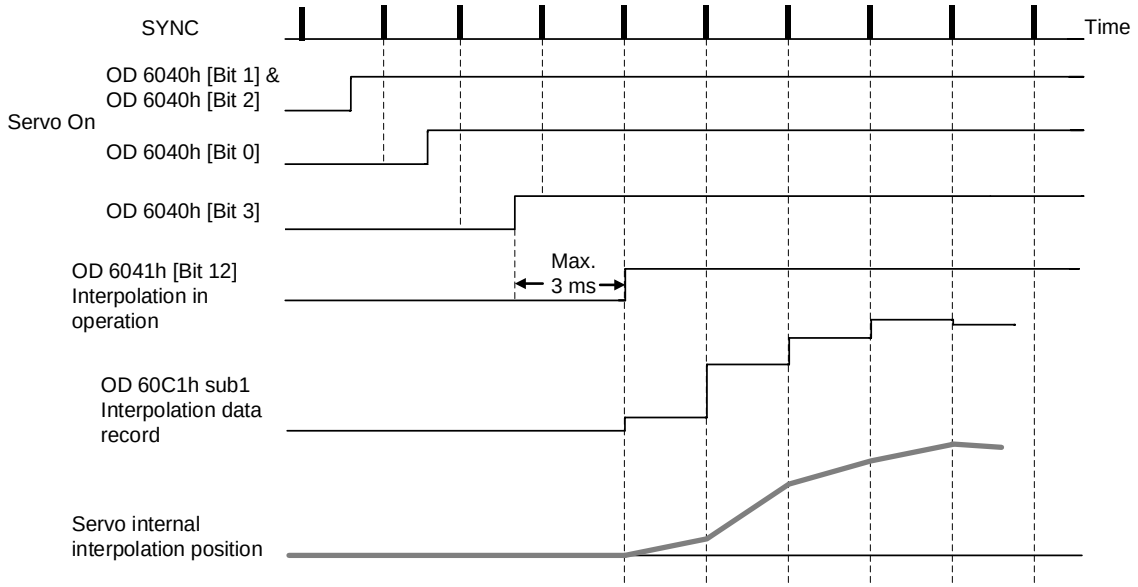
Interpolated Position (IP) mode requires a series of position data to complete the interpolation for positioning. Different from PP (Profile Position) mode, all the motion command paths in IP mode are issued by the controller. The servo drive only follows each position that the controller issues and finally completes a motion command. Delta servo drives only support synchronous operation in which the controller periodically sends the SYNC object (COB-ID = 0x80). The interpolation time period can be set with OD 60C2h. And the controller issues the position command to the interpolation position of OD 60C1h.



Operation steps:

1. Set OD 6060h to 07h to set the mode as Interpolated Position mode.
2. Set OD 60C2h for the interpolation time period. The setting must be the same as the communication cycle period (OD 1006h).
3. In the PDO mapping setting of the controller, configure one set of RxPDO to be OD 60C1h sub1 and OD 60C1h sub2.
4. In the PDO mapping setting of the controller, configure the objects to be monitored in TxPDO according to the requirements, such as the position actual value (OD 6064h).
5. Set the Controlword (OD 6040h). Follow these steps for operation. Steps 5.1 and 5.2 are to bring the servo drive's state machine into the ready state. For more details of the state machine, refer to the OD 6040h description in Section 11.4.3.3.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
5.1	0	0	1	1	0	Shutdown.
5.2	0	0	1	1	1	Switch on (ready for Servo On).
5.3	0	1	1	1	1	Enable operation (Servo On).



Adjustment method:

It is suggested that you set the SYNC communication cycle period (OD 1006h) between 1 ms and 10 ms. If the cycle period is too long, the interval between cycles also increases. If the position change is big, it causes speed fluctuations. In this case, use P1.036 (S-curve acceleration / deceleration smoothing constant) or P1.068 (Position command - moving filter) to smooth the position difference. Since the jitter of each controller is different, the time the servo receives the SYNC differs from the SYNC communication cycle time. When this happens, adjust the value of P3.009.U to increase the error range and have the servo drive automatically correct the internal timer so it is consistent with the communication cycle of the controller.

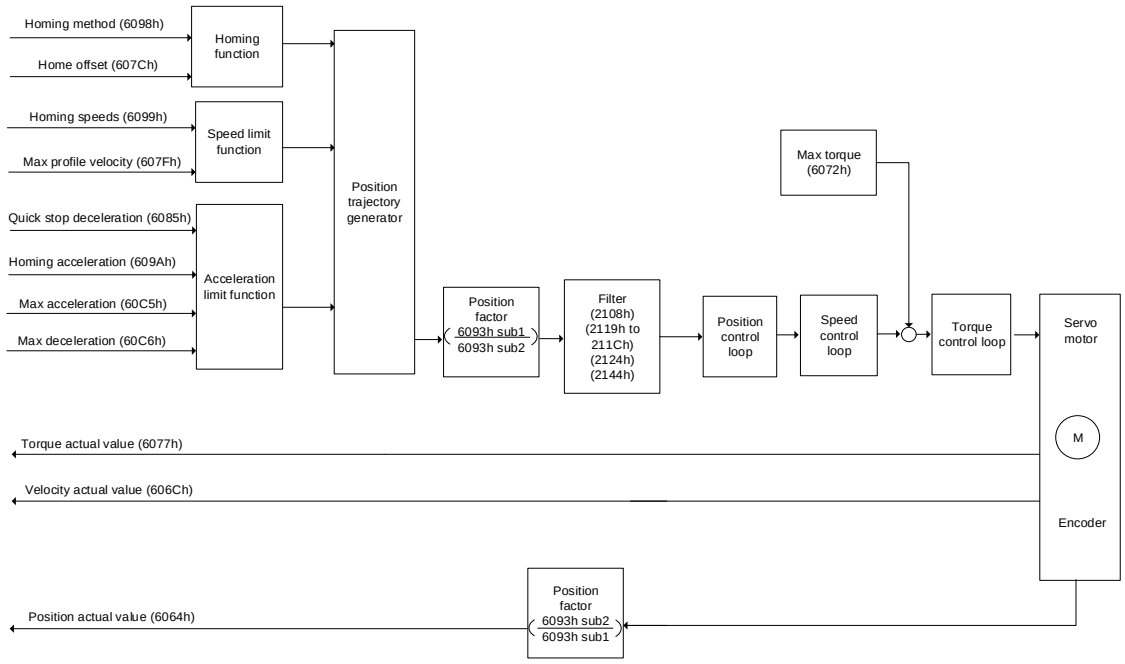
Relevant object list

Index	Name	Data type	Access
6040h	Controlword	UNSIGNED16	RW
6041h	Statusword	UNSIGNED16	RO
6060h	Modes of operation	INTEGER8	RW
6061h	Modes of operation display	INTEGER8	RO
6093h	Position factor	UNSIGNED32	RW
60C0h	Interpolation sub mode select	INTEGER16	RW
60C1h	Interpolation data record	INTEGER32	RW

Note: for more details, refer to Section 11.4.3 Details of objects.

11.3.3 Homing mode

After homing is complete, the position system of the servo drive is established and the drive can start executing the position command issued by the controller. The Delta servo drive offers 39 homing methods, including homing on the home switch, positive or negative limit, motor Z pulse, and hard stop.



Operation steps:

1. Set OD 6060h to 06h to set the mode as Homing mode.
2. Set OD 607Ch for the home offset.
3. Set OD 6098h for the homing method.
4. Set OD 6099h sub1 for the speed when searching for the home switch.
5. Set OD 6099h sub2 for the speed when searching for the Z pulse.
6. Set OD 609Ah for the homing acceleration.
7. Set the Controlword (OD 6040h). Follow these steps for operation. Steps 7.1 and 7.2 are to bring the servo drive's state machine into the ready state. For more details of the state machine, refer to the OD 6040h description in Section 11.4.3.3.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
7.1	0	0	1	1	0	Shutdown.
7.2	0	0	1	1	1	Switch on (ready for Servo On).
7.3	0	1	1	1	1	Enable operation (Servo On).
7.4	1	1	1	1	1	Homing (rising-edge triggered).

11

Read the servo drive information:

1. Read OD 6041h to obtain the servo drive status.
2. Read OD 6064h to obtain the actual value of the motor position at present.

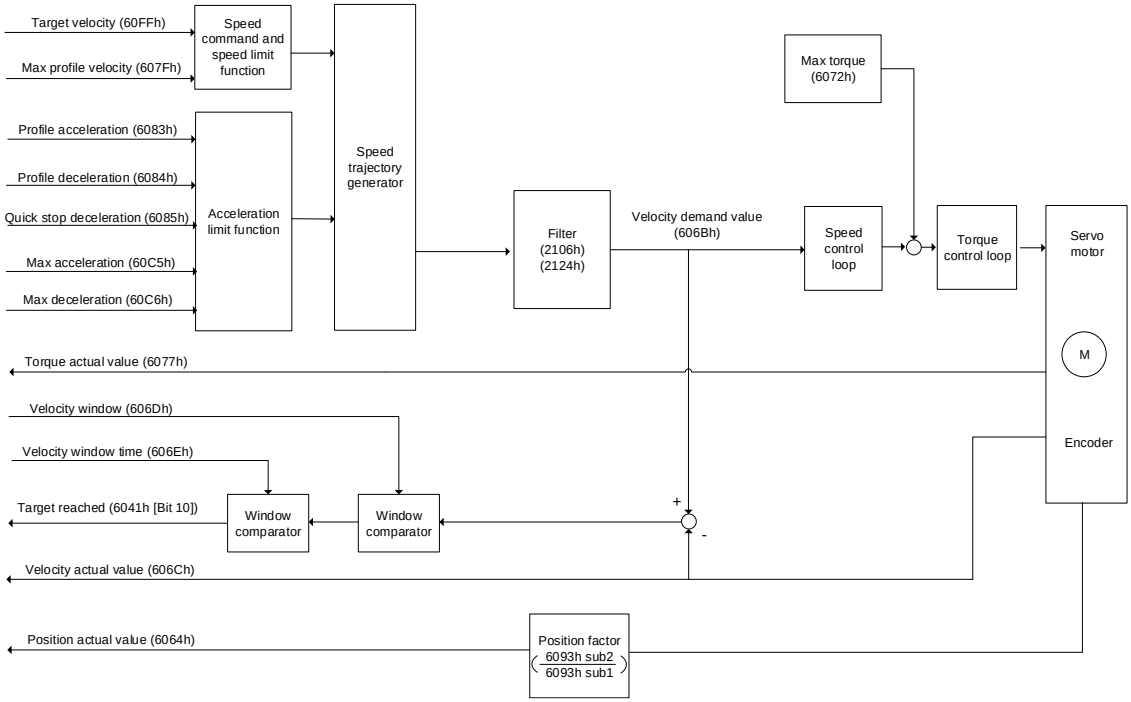
Relevant object list

Index	Name	Data type	Access
6040h	Controlword	UNSIGNED16	RW
6041h	Statusword	UNSIGNED16	RO
6060h	Modes of operation	INTEGER8	RW
6061h	Modes of operation display	INTEGER8	RO
6064h	Position actual value [PUU]	INTEGER32	RO
606Ch	Velocity actual value	INTEGER32	RO
6072h	Max torque	UNSIGNED16	RW
607Ch	Home offset	INTEGER32	RW
607Fh	Max profile velocity	UNSIGNED32	RW
6085h	Quick stop deceleration	UNSIGNED32	RW
6093h	Position factor	UNSIGNED32	RW
6098h	Homing method	INTEGER8	RW
6099h	Homing speeds	UNSIGNED32	RW
609Ah	Homing acceleration	UNSIGNED32	RW
60C5h	Max acceleration	UNSIGNED32	RW
60C6h	Max deceleration	UNSIGNED32	RW

Note: for more details, refer to Section 11.4.3 Details of objects.

11.3.4 Profile Velocity mode

In Profile Velocity (PV) mode, the controller specifies the speed command and acceleration / deceleration settings, and then the trajectory generator of the servo drive plans the motion path according to these conditions.



Operation steps:

1. Set OD 6060h to 03h to set the mode as Profile Velocity mode.
2. Set OD 6083h for the profile acceleration.
3. Set OD 6084h for the profile deceleration.
4. Set the target velocity (OD 60FFh) to 0. In Profile Velocity mode, the servo motor starts operating once the servo drive is switched to Servo On (Step 5). Therefore, setting the target velocity (OD 60FFh) to 0 is to ensure that the motor maintains at 0 rpm at the moment of Servo On.
5. Set the Controlword (OD 6040h). Follow these steps for operation. Steps 5.1 and 5.2 are to bring the servo drive's state machine into the ready state. For more details of the state machine, refer to the OD 6040h description in Section 11.4.3.3.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
5.1	0	0	1	1	0	Shutdown.
5.2	0	0	1	1	1	Switch on (ready for Servo On).
5.3	0	1	1	1	1	Enable operation (Servo On).

6. Set OD 60FFh for the target velocity.

Read the servo drive information:

1. Read OD 6041h to obtain the servo drive status.
2. Read OD 606Ch to obtain the current speed feedback.

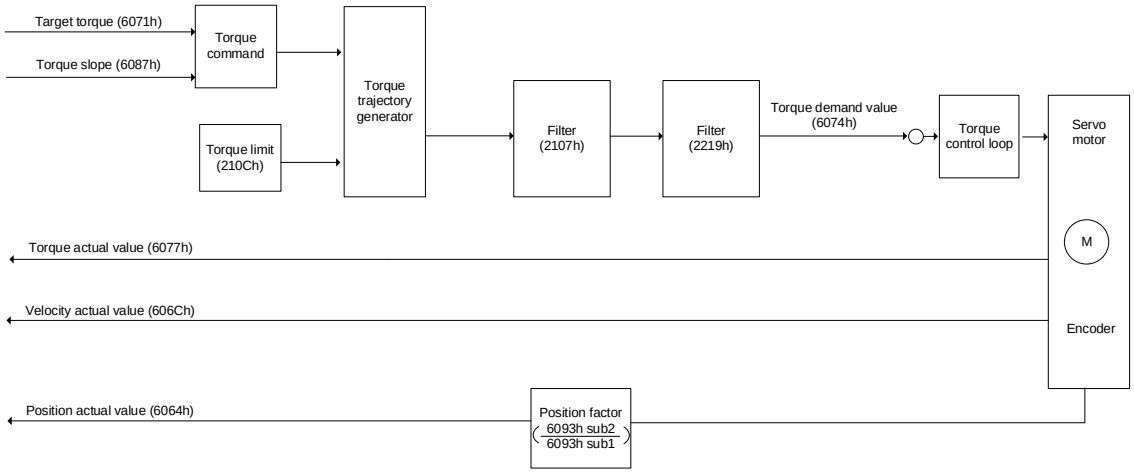
Relevant object list

Index	Name	Data type	Access
6040h	Controlword	UNSIGNED16	RW
6041h	Statusword	UNSIGNED16	RO
6060h	Modes of operation	INTEGER8	RW
6061h	Modes of operation display	INTEGER8	RO
6064h	Position actual value [PUU]	INTEGER32	RO
606Bh	Velocity demand value	INTEGER32	RO
606Ch	Velocity actual value	INTEGER32	RO
606Dh	Velocity window	UNSIGNED16	RW
606Eh	Velocity window time	UNSIGNED16	RW
606Fh	Velocity threshold	UNSIGNED16	RW
6072h	Max torque	UNSIGNED16	RW
6077h	Torque actual value	INTEGER16	RO
607Fh	Max profile velocity	UNSIGNED32	RW
6083h	Profile acceleration	UNSIGNED32	RW
6084h	Profile deceleration	UNSIGNED32	RW
6085h	Quick stop deceleration	UNSIGNED32	RW
6093h	Position factor	UNSIGNED32	RW
60C5h	Max acceleration	UNSIGNED32	RW
60C6h	Max deceleration	UNSIGNED32	RW
60FFh	Target velocity	INTEGER32	RW

Note: for more details, refer to Section 11.4.3 Details of objects.

11.3.5 Profile Torque mode

In Profile Torque (PT) mode, the controller specifies the torque command and filtering conditions, and then the trajectory generator of the servo drive plans the torque slope according to these conditions.



Operation steps:

1. Set OD 6060h to 04h to set the mode as Profile Torque mode.
2. Set OD 6087h for the torque slope.
3. Set the target torque (OD 6071h) to 0. In Profile Torque mode, the servo target torque takes effect once the servo drive is switched to Servo On (Step 4). Therefore, set the target torque (OD 6071h) to 0 for safety reasons.
4. Set the Controlword (OD 6040h). Follow these steps for operation. Steps 4.1 and 4.2 are to bring the servo drive's state machine into the ready state. For more details of the state machine, refer to the OD 6040h description in Section 11.4.3.3.

Step	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
4.1	0	0	1	1	0	Shutdown.
4.2	0	0	1	1	1	Switch on (ready for Servo On).
4.3	0	1	1	1	1	Enable operation (Servo On).

5. Set OD 6071h for the target torque.

Read the servo drive information:

1. Read OD 6041h to obtain the servo drive status.
2. Read OD 6077h to obtain the current torque feedback.

Relevant object list

Index	Name	Data type	Access
6040h	Controlword	UNSIGNED16	RW
6041h	Statusword	UNSIGNED16	RO
6060h	Modes of operation	INTEGER8	RW
6061h	Modes of operation display	INTEGER8	RO
6064h	Position actual value [PUU]	INTEGER32	RO
606Ch	Velocity actual value	INTEGER32	RO
6071h	Target torque	INTEGER16	RW
6074h	Torque demand value	INTEGER16	RO
6075h	Motor rated current	UNSIGNED32	RO
6077h	Torque actual value	INTEGER16	RO
6078h	Current actual value	INTEGER16	RO
6087h	Torque slope	UNSIGNED32	RW
6093h	Position factor	UNSIGNED32	RW

Note: for more details, refer to Section 11.4.3 Details of objects.

11.4 Object dictionary

This section details the CANopen objects supported by the servo. The contents include object index, name, data type, data length, and read / write permissions (access).

11.4.1 Specifications for objects

Object code

Object code	Description
VAR	A single value, such as an UNSIGNED8, Boolean, float, and INTEGER16.
ARRAY	An object of multiple data fields consisting of multiple variables of the same data type, such as an UNSIGNED16 array. The sub-index 0 data type is UNSIGNED8, so it is not an ARRAY data.
RECORD	An object of multiple data fields consisting of multiple variables of different data types. The sub-index 0 data type is UNSIGNED8, so it is not a RECORD data.

Data type

Refer to CANopen DS301.

11

11.4.2 List of objects

OD 1XXXh communication object group

Index	Object code	Name	Data type	Access
1000h	VAR	Device type	UNSIGNED32	RO
1001h	VAR	Error register	UNSIGNED8	RO
1003h	ARRAY	Pre-defined error field	UNSIGNED32	RW
1005h	VAR	COB-ID SYNC message	UNSIGNED32	RO
1006h	VAR	Communication cycle period	UNSIGNED32	RW
100Ch	VAR	Guard time	UNSIGNED16	RW
100Dh	VAR	Life time factor	UNSIGNED8	RW
1010h	ARRAY	Store parameters	UNSIGNED32	RW
1011h	ARRAY	Restore parameters	UNSIGNED32	RW
1014h	VAR	COB-ID emergency message	UNSIGNED32	RO
1016h	ARRAY	Consumer heartbeat time	UNSIGNED32	RW
1017h	VAR	Producer heartbeat time	UNSIGNED16	RW
1018h	RECORD	Identity object	UNSIGNED32	RO
1029h	ARRAY	Error behavior	UNSIGNED8	RW
1200h	RECORD	Server SDO parameter	SDO parameter	RO
1400h - 1403h	RECORD	Receive PDO communication parameter	UNSIGNED16/32	RW
1600h - 1603h	RECORD	Receive PDO mapping parameter	UNSIGNED32	RW
1800h - 1803h	RECORD	Transmit PDO communication parameter	UNSIGNED16/32	RW
1A00h - 1A03h	RECORD	Transmit PDO mapping parameter	UNSIGNED32	RW

Note: only 1001h can be mapped to PDO.

OD 2XXXh servo parameter group

Index	Object code	Name	Data type	Access	Mappable
2XXXh	VAR	Parameter mapping	INTEGER16/32	RW	Y

OD 6XXXh communication object group

Index	Object code	Name	Data type	Access	Mappable
603Fh	VAR	Error code	UNSIGNED16	RO	Y
6040h	VAR	Controlword	UNSIGNED16	RW	Y
6041h	VAR	Statusword	UNSIGNED16	RO	Y
605Bh	VAR	Shutdown option code	INTEGER16	RW	Y
6060h	VAR	Modes of operation	INTEGER8	RW	Y
6061h	VAR	Modes of operation display	INTEGER8	RO	Y
6062h	VAR	Position demand value [PUU]	INTEGER32	RO	Y
6063h	VAR	Position actual internal value [Pulse]	INTEGER32	RO	Y
6064h	VAR	Position actual value [PUU]	INTEGER32	RO	Y
6065h	VAR	Following error window	UNSIGNED32	RW	Y
6067h	VAR	Position window	UNSIGNED32	RW	Y
6068h	VAR	Position window time	UNSIGNED16	RW	Y
606Bh	VAR	Velocity demand value	INTEGER32	RO	Y
606Ch	VAR	Velocity actual value	INTEGER32	RO	Y
606Dh	VAR	Velocity window	UNSIGNED16	RW	Y
606Eh	VAR	Velocity window time	UNSIGNED16	RW	Y

Index	Object code	Name	Data type	Access	Mappable
606Fh	VAR	Velocity threshold	UNSIGNED16	RW	Y
6071h	VAR	Target torque	INTEGER16	RW	Y
6072h	VAR	Max torque	UNSIGNED16	RW	Y
6074h	VAR	Torque demand value	INTEGER16	RO	Y
6075h	VAR	Motor rated current	UNSIGNED32	RO	Y
6076h	VAR	Motor rated torque	UNSIGNED32	RO	Y
6077h	VAR	Torque actual value	INTEGER16	RO	Y
6078h	VAR	Current actual value	INTEGER16	RO	Y
607Ah	VAR	Target position	INTEGER32	RW	Y
607Ch	VAR	Home offset	INTEGER32	RW	Y
607Dh	ARRAY	Software position limit	INTEGER32	RW	Y
607Fh	VAR	Max profile velocity	UNSIGNED32	RW	Y
6080h	VAR	Max motor speed	UNSIGNED32	RW	Y
6081h	VAR	Profile velocity	UNSIGNED32	RW	Y
6083h	VAR	Profile acceleration	UNSIGNED32	RW	Y
6084h	VAR	Profile deceleration	UNSIGNED32	RW	Y
6085h	VAR	Quick stop deceleration	UNSIGNED32	RW	Y
6087h	VAR	Torque slope	UNSIGNED32	RW	Y
6093h	ARRAY	Position factor	UNSIGNED32	RW	Y
6098h	VAR	Homing method	INTEGER8	RW	Y
6099h	ARRAY	Homing speeds	UNSIGNED32	RW	Y
609Ah	VAR	Homing acceleration	UNSIGNED32	RW	Y
60C0h	VAR	Interpolation sub mode select	INTEGER16	RW	Y
60C1h	ARRAY	Interpolation data record	INTEGER32	RW	Y
60C2h	RECORD	Interpolation time period	UNSIGNED8	RW	Y
60C5h	VAR	Max acceleration	UNSIGNED32	RW	Y
60C6h	VAR	Max deceleration	UNSIGNED32	RW	Y
60F4h	VAR	Following error actual value	INTEGER32	RO	Y
60FCh	VAR	Position demand value	INTEGER32	RO	Y
60FDh	VAR	Digital inputs	UNSIGNED32	RO	Y
60FEh	ARRAY	Digital outputs	UNSIGNED32	RW	Y
60FFh	VAR	Target velocity	INTEGER32	RW	Y
6502h	VAR	Supported drive modes	UNSIGNED32	RO	Y

11

11.4.3 Details of objects

11.4.3.1 OD 1XXXh communication object group

Object 1000h: Device type

Index	1000h
Name	Device type
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32

Format of this object: (High word h) DCBA; (Low word L) UZYX

A	Bit 16 - Bit 31 Model type	X	Bit 0 - Bit 15 Device profile number
B		Y	
C		Z	
D		U	

Definitions are as follows:

- UZYX: device profile number (servo drive: 0192)
- DCBA: model type

DCBA	Model type
0402	A2
0602	M
0702	A3
0B02	B3

Object 1001h: Error register

Index	1001h
Name	Error register
Object code	VAR
Data type	UNSIGNED8
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED8
Default	0

Object function:

The bits and corresponding functions are as follows:

Bit	7	6	5	4	3	2	1	0

Bit	Function
Bit 0	Generic error
Bit 1	Current
Bit 2	Voltage
Bit 3	Temperature
Bit 4	Communication error
Bit 5 - Bit 7	Reserved

Object 1003h: Pre-defined error field

Index	1003h
Name	Pre-defined error field
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	No

Sub-index	0
Description	Number of errors
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	0 - 5
Default	0

Sub-index	1 - 5
Description	Standard error field
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	0

11

Format of this object: (High word h) DCBA; (Low word L) UZYX

A	Bit 16 - Bit 31 Delta servo alarm	X	Bit 0 - Bit 15 Error code
B		Y	
C		Z	
D		U	

Definitions are as follows:

- UZYX: error code. Refer to the error code definition in DS402.
- DCBA: Delta servo alarm. Refer to Chapter 14 Troubleshooting.

Example:

When you operate the servo, if the encoder cable is not correctly connected, the servo drive panel displays AL011 and the error code is stored in the OD 1003h array. The display is as follows:

Byte:	High word	Low word
	Delta servo alarm (UINT16)	Error code (UINT16)
	0x0011	0x7305

AL011 is defined as “CN2 communication failed” according to the Delta servo alarm.

Error code: 0x7305 is defined as “Incremental sensor 1 fault” according to DS402.

Object 1005h: COB-ID SYNC message

Index	1005h
Name	COB-ID SYNC message
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	80h

Object function:

This object is read-only and cannot be set.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 10	SYNC-COB-ID = 0x80	-
Bit 11 - Bit 31	Reserved	-

Object 1006h: Communication cycle period

Index	1006h
Name	Communication cycle period
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	0
Unit	μs

Object function:

This object is to set the communication cycle, which is the interval between two SYNCs. If you are not using SYNC, set this object to 0.

Object 100Ch: Guard time

Index	100Ch
Name	Guard time
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	UNSIGNED16
Default	0
Unit	ms

Object function:

OD 100Ch (guard time) multiplied by OD 100Dh (multiplying factor) gives the life time for the Life Guarding Protocol. If the guard time (OD 100Ch) is set to 0, then the Life Guarding Protocol is invalid.

Example: if OD 100Ch = 5 ms and OD 100Dh = 10, then the life time is 50 ms.

Object 100Dh: Life time factor

Index	100Dh
Name	Life time factor
Object code	VAR
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

Object function:

OD 100Ch (guard time) multiplied by OD 100Dh (multiplying factor) gives the life time for the Life Guarding Protocol. If the guard time (OD 100Ch) is set to 0, then the Life Guarding Protocol is invalid.

Example: if OD 100Ch = 5 ms and OD 100Dh = 10, then the life time is 50 ms.

Object 1010h: Store parameters

Index	1010h
Name	Store parameters
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	No

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	UNSIGNED8
Default	1

Sub-index	1
Description	Store communication parameters
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	0x65766173 (save)
Default	1

Object function:

You can only write 0x65766173 (save) to OD 1010h sub1, writing all current OD setting values to the EEPROM.

Object 1011h: Restore parameters

Index	1011h
Name	Restore parameters
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	No

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	UNSIGNED8
Default	1

Sub-index	1
Description	Restore communication parameters
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	0x64616F6C (load)
Default	1

Object function:

You can only write 0x64616F6C (load) to OD 1011h sub1, resetting all ODs to their default values.

11

Object 1014h: COB-ID emergency message

Index	1014h
Name	COB-ID emergency message
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	80h + Node-ID

Object function:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 10	COB-ID	80h + Node-ID. The data size is 11-bit.
Bit 11 - Bit 30	Reserved	-
Bit 31	Emergency (EMCY) function	0: enabled (servo drive sends the EMCY command). 1: disabled (servo drive does not send the EMCY command).

The COB-ID setting format is as follows:

Communication object	Function code Bit 10 9 8 7	Node ID Bit 6 5 4 3 2 1 0	COB-ID DEC (HEX)
EMCY object	0001	1	129 (81h)
		2	130 (82h)
		…	…
		127	255 (FFh)

Object 1016h: Consumer heartbeat time

Index	1016h
Name	Consumer heartbeat time
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	No

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	1
Default	1

Sub-index	1
Description	Consumer heartbeat time
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	0

Object function:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 15	Heartbeat time	UNSIGNED8; unit: ms
Bit 16 - Bit 23	Node-ID	UNSIGNED8
Bit 24 - Bit 31	Reserved	-

Consumer heartbeat time is defined as the time the servo drive expects to receive a heartbeat. When the servo drive does not receive the heartbeat signal within the receiving time, it triggers the heartbeat event, meaning AL180 is triggered. The consumer heartbeat time must be greater than the producer heartbeat time. Since there are delays and other uncontrollable external factors in transmitting the heartbeat message, you must retain a tolerance time for the transmission.

11

Object 1017h: Producer heartbeat time

Index	1017h
Name	Producer heartbeat time
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	UNSIGNED16
Default	0

Object function:

Producer heartbeat time is defined as the cycle time of the heartbeat. When this value is set to 0, this function is invalid.

Object 1018h: Identity object

Index	1018h
Name	Identity object
Object code	RECORD
Data type	Identity
Access	RO
PDO mapping	No

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	3
Default	3

Sub-index	1
Description	Vendor ID
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	1DDh

Sub-index	2
Description	Product code
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	6000h: A2 series 6010h: A3 series 6030h: M series 6080h: B3 series

Sub-index	3
Description	Version
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	N/A

Object function:

This object includes the servo drive information.

Object 1029h: Error behavior

Index	1029h
Name	Error behavior
Object code	ARRAY
Data type	UNSIGNED8
Access	RW
PDO mapping	No

Sub-index	0
Description	Number of error types
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	1
Default	1

11

Sub-index	1
Description	Communication error
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

Object function:

Generally, when a serious fault is detected in the Operational state, the servo drive automatically switches to the Pre-Operational state. Use this object setting to switch the state to the Pre-Operational state, keep the original state, or switch to the Stopped state.

OD 1029h sub1 setting	Switch the state to
0	Pre-Operational (only when the servo is currently in the Operational state)
1	Keep the original state
2	Stopped

Object 1200h: Server SDO parameter

Index	1200h
Name	Server SDO parameter
Object code	RECORD
Data type	SDO parameter
Access	RO
PDO mapping	No

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	2
Default	2

Sub-index	1
Description	Controller sends to servo drive COB-ID Client->Server (rx)
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	Index 1200h: 600h + Node-ID

Sub-index	2
Description	Servo drive returns to controller COB-ID Server->Client (tx)
Data type	UNSIGNED32
Access	RO
PDO mapping	No
Setting range	UNSIGNED32
Default	Index 1200h: 580h + Node-ID

Object function:

This object is read-only and cannot be set. Read the station number for transmitting and receiving the SDO with this object.

Example:

If the servo drive station number for receiving is 10:

600h + Node-ID: Ah = 600h + Ah = 60Ah

OD 1200h sub1 reads 60Ah.

If the servo drive station number for transmitting is 10:

580h + Node-ID: Ah = 580h + Ah = 58Ah

OD 1200h sub2 reads 58Ah.

11

Objects 1400h - 1403h: Receive PDO communication parameter

Index	1400h, 1401h, 1402h, 1403h
Name	Receive PDO communication parameter
Object code	RECORD
Data type	PDO CommPar
Access	RW

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	5
Default	5

Sub-index	1
Description	COB-ID used by PDO
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	Node-ID: 0

Object function:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 10	COB-ID	The data size is 11-bit.
Bit 11 - Bit 30	Reserved	-
Bit 31	PDO function switch	0: enable 1: disable Enable / disable the PDO function to determine if the PDO is used in the Operational state.

The COB-ID setting format is as follows:

Communication object	Object index	COB-ID DEC (HEX)
RxPDO1	1400h	512 (200h) + Node-ID
RxPDO2	1401h	768 (300h) + Node-ID
RxPDO3	1402h	1024 (400h) + Node-ID
RxPDO4	1403h	1280 (500h) + Node-ID

Sub-index	2
Description	Transmission type
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

Object function:

The transmission type setting is as follows.

Setting value	Transmission type				
	Cyclic	Acyclic	Synchronous	Asynchronous	RTR only
00h (0)	-	V	V	-	-
01h - F0h (1 - 240)	V	-	V	-	-
F1h - FBh (241 - 251)	Reserved				
FCh (252)	-	-	V	-	V
FDh (253)	-	-	-	V	V
FEh (254)	-	-	-	V	-
FFh (255)	-	-	-	V	-

Sub-index	3
Description	Inhibit time (not used for RxPDO)
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	UNSIGNED16
Default	0

Sub-index	4
Description	Compatibility entry
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

11

Sub-index	5
Description	Event timer (not used for RxPDO)
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	UNSIGNED16
Default	0

Objects 1600h - 1603h: Receive PDO mapping parameter

Index	1600h, 1601h, 1602h, 1603h
Name	Receive PDO mapping parameter
Object code	RECORD
Data type	PDO mapping
Access	RW
Note	The total length of objects in a group of PDO cannot exceed 64 bits.

Sub-index	0
Description	Number of PDO mappings
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	0: disable 1 - 8: set the number of PDO mapping and enable the function
Default	0

Sub-index	1 - 8
Description	Specify the 1st (to 8th) object and its content to be mapped
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	0

The format of this object is as follows:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function
Bit 0 - Bit 7	Object data length
Bit 8 - Bit 15	Object sub-index
Bit 16 - Bit 31	Object index

Example:

To set the three PDOs, OD 6040h, OD 607Ah, and OD 6060h, in the first group of PDO, the setting is as follows:

Mapping parameter setting for RxPDO	Data			Description
OD 1600h sub0	3			Set 3 PDO mappings.
OD 1600h sub1	6040h	00h	10h	Mapping the Controlword (OD 6040h); data length is 16-bit.
OD 1600h sub2	607Ah	00h	20h	Mapping the target position (OD 607Ah); data length is 32-bit.
OD 1600h sub3	6060h	00h	08h	Mapping the operation mode (OD 6060h); data length is 8-bit.
Note	The total length is 38h (56-bit) which meets the specification of less than 64-bit.			

Objects 1800h - 1803h: Transmit PDO communication parameter

Index	1800h, 1801h, 1802h, 1803h
Name	Transmit PDO communication parameter
Object code	RECORD
Data type	PDO CommPar
Access	RW

Sub-index	0
Description	Largest sub-index supported
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	5
Default	5

11

Sub-index	1
Description	COB-ID used by PDO
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	Default Node-ID: 0 OD 1800h: 180h + Node-ID OD 1801h: 280h + Node-ID OD 1802h: 380h + Node-ID OD 1803h: 480h + Node-ID

Object function:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0 - Bit 10	COB-ID	The data size is 11-bit.
Bit 11 - Bit 30	Reserved	-
Bit 31	PDO function switch	0: enable 1: disable Enable / disable the PDO function to determine if the PDO is used in the Operational state.

Sub-index	2
Description	Transmission type
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

Object function:

The transmission type setting is as follows:

Setting value	Transmission type				
	Cyclic	Acyclic	Synchronous	Asynchronous	RTR only
00h (0)	-	V	V	-	-
01h - F0h (1 - 240)	V	-	V	-	-
F1h - FBh (241 - 251)	Reserved				
FCh (252)	-	-	V	-	V
FDh (253)	-	-	-	V	V
FEh (254)	-	-	-	V	-
FFh (255)	-	-	-	V	-

Sub-index	3
Description	Inhibit time
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	UNSIGNED16
Default	0

Sub-index	4
Description	Reserved
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	UNSIGNED8
Default	0

Sub-index	5
Description	Event timer
Data type	UNSIGNED16
Access	RW
PDO mapping	No
Setting range	0: not in use UNSIGNED16
Default	0

11

Objects 1A00h - 1A03h: Transmit PDO mapping parameter

Index	1A00h, 1A01h, 1A02h, 1A03h
Name	Transmit PDO mapping parameter
Object code	RECORD
Data type	PDO mapping
Access	RW
Note	The total length of objects in a group of PDO cannot exceed 64 bits.

Sub-index	0
Description	Number of PDO mappings
Data type	UNSIGNED8
Access	RW
PDO mapping	No
Setting range	0: disable 1 - 8: set the number of PDO mapping and enable the function
Default	0

Sub-index	1 - 8
Description	Specify the 1st (to 8th) object and its content to be mapped
Data type	UNSIGNED32
Access	RW
PDO mapping	No
Setting range	UNSIGNED32
Default	0

Format of this object: (High word h) DCBA; (Low word L) UZYX

DCBA	Bit 16 - Bit 31 Object index	YX	Bit 0 - Bit 7 Object data length
		UZ	Bit 8 - Bit 15 Object sub-index

11.4.3.2 OD 2XXXh servo parameter group

Object 2XXXh: Parameter mapping

Index	2XXXh
Name	Parameter mapping
Object code	VAR
Data type	INTEGER16 / INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER16 / INTEGER32
Default	N/A

Object function:

Access the corresponding servo parameters with the OD 2XXXh group. The conversion between the parameter number and object index is as follows:

Object index	Servo parameter	Description
2**aBC**h	P**a**.**bcd**	"**BC**" is the hexadecimal format of "**bcd**".

You can read the object index first to get the information of the parameter length, and then use the SDO or PDO to change the data.

Example 1:

Object 2300h: Node-ID [P3.000]

Index	2300h
Name	Node-ID
Object code	VAR
Data type	INTEGER16
Access	RW
PDO mapping	Yes
Setting range	INTEGER16
Default	7F

11

Example 2:

Object 212Ch: Electronic gear [P1.044]

Index	212Ch
Name	Electronic gear
Object code	VAR
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER32
Default	1

11.4.3.3 OD 6XXXh communication object group

Object 603Fh: Error code (CANopen-defined)

Index	603Fh
Name	Error code
Object code	VAR
Data type	UNSIGNED16
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED16
Default	0

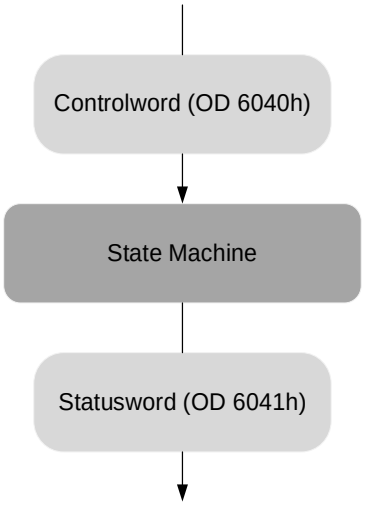
Object 6040h: Controlword

Index	6040h
Name	Controlword
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED16
Default	0x0004

Object function:

The Controlword contains many functions, such as Servo On, command triggering, fault reset, and quick stop.

The state machine architecture is as follows:



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function	Description
Bit 0	Switch on	Ready for Servo On.
Bit 1	Enable voltage	-
Bit 2	Quick stop (B contact (NC))	-
Bit 3	Enable operation	Servo On.
Bit 4 - Bit 6	Defined in each operation mode	These bits are individually defined according to the operation mode, as shown in the following table.
Bit 7	Fault reset	-
Bit 8	Halt	-
Bit 9 - Bit 15	Reserved	-

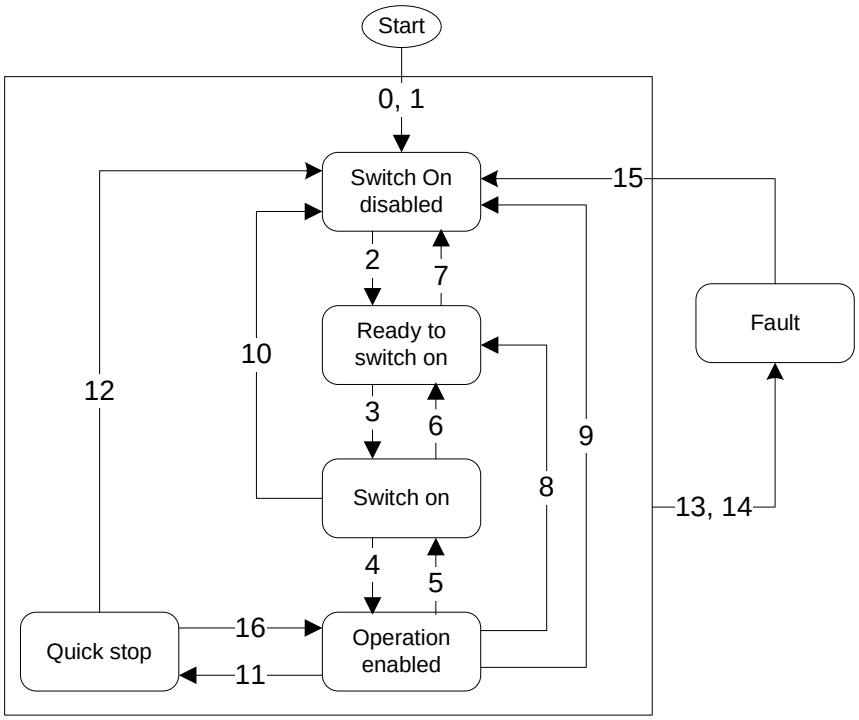
Bits 4 - 6 are individually defined according to the operation mode, as shown in the following table:

Bit	Definition in each operation mode		
	Profile Position mode	Homing mode	Profile Velocity mode Profile Torque mode Interpolated Position mode
Bit 4	Command triggering (rising-edge triggered)	Homing (rising-edge triggered)	-
Bit 5	Function for the command to take immediate effect	-	-
Bit 6	0: absolute position command 1: relative position command	-	-

Note: - indicates the bit is invalid.

11

Finite state machine (as shown in the following diagram) defines the behavior of a servo drive system. Each state represents an internal or external behavior. For example, the servo drive can execute point-to-point motion only in the Operation enabled state.



The state transition is defined as follows:

Transition	Event	Action
0, 1	Automatic transition after power-on	Device boot and initialization
2	Shutdown command	N/A
3	Switch on command	Servo is ready for Servo On
4	Enable operation command	Servo switches to Servo On and enters the mode in which the controller is allowed to issue a motion command.
5	Disable operation command	Servo switches to Servo Off
6	Shutdown command	N/A
7	Disable voltage or quick stop command	N/A
8	Shutdown command	Servo switches to Servo Off
9	Disable voltage command	Servo switches to Servo Off
10	Disable voltage or quick stop command	N/A
11	Quick stop command The following two errors belong to this quick stop type: 1. Positive / negative limit switch triggered 2. Quick stop triggered by the Controlword (OD 6040h [Bit 2] = 0)	Quick stop function is enabled. The time setting for deceleration to a stop is different for the two errors. 1. OD 2503h (P5.003) 2. OD 6085h
12	Disable voltage command (OD 6040h = 0000 0110 or OD 6040h [Bit 1] = 0)	Servo switches to Servo Off
13, 14	Alarm occurs	Servo switches to Servo Off
15	Fault reset	N/A
16	Enable operation command; no alarm	Motion operation restart. The restart action is mode-dependent.

State transition can be achieved by issuing commands with the Controlword (OD 6040h). The settings of OD 6040h for different commands are as follows:

OD 6040h					Command	Transition
Bit 7	Bit 3	Bit 2	Bit 1	Bit 0		
0	X	1	1	0	Shutdown	2, 6, 8
0	0	1	1	1	Switch on	3
0	1	1	1	1	Switch on + Enable operation	3 + 4
0	X	X	0	X	Disable voltage	7, 9, 10, 12
0	X	0	1	X	Quick stop	7, 10, 11
0	0	1	1	1	Disable operation	5
0	1	1	1	1	Enable operation	4, 16
	X	X	X	X	Fault reset	15

Object 6041h: Statusword

Index	6041h
Name	Statusword
Object code	VAR
Data type	UNSIGNED16
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED16
Default	0

Object function:

The Statusword contains many statuses, such as Servo On, command statuses, fault signal, and quick stop. The state machine architecture is as follows:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Status		Description
Bit 0	Ready to switch on	Ready to be activated	Current status of the servo drive (see the following table for details).
Bit 1	Switched on	Servo ready	
Bit 2	Operation enabled	Servo On	
Bit 3	Fault	Fault signal	
Bit 4	Voltage enabled	Servo is powered on	
Bit 5	Quick stop	Quick stop	
Bit 6	Switch on disabled	Servo disabled	
Bit 7	Warning	Warning signal	When outputting the warning signal, the servo keeps outputting the Servo On signal.
Bit 8	Reserved	-	-
Bit 9	Remote	Remote control	-
Bit 10	Target reached	Target reached	-
Bit 11	Reserved	-	-

11

Bit	Status		Description
Bit 12 - Bit 13	-	-	These bits are individually defined according to the operation mode, as shown in the following table.
Bit 14	Positive limit	Positive limit	-
Bit 15	Negative limit	Negative limit	-

Bit 0 - Bit 6: current status of the servo drive.

Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description
0	-	-	0	0	0	0	Not ready to switch on.
1	-	-	0	0	0	0	Switch on disabled.
0	1	-	0	0	0	1	Ready to switch on.
0	1	-	0	0	1	1	Switched on.
0	1	-	0	1	1	1	Operation enabled (Servo On).
0	0	-	0	1	1	1	Quick stop active.
0	-	-	1	1	1	1	Fault reaction active.
0	-	-	1	0	0	0	Servo fault (servo switches to Servo Off).

Note: 0 indicates the bit is off, 1 indicates the bit is on, and - indicates the bit is invalid.

Bit 12 - Bit 13: current status of the servo drive.

Bit	Definition in each operation mode				
	Profile Position mode	Homing mode	Interpolated Position mode	Profile Velocity mode	Profile Torque mode
Bit 12	Set-point acknowledge (servo received the command signal)	Homing is complete	Interpolation in operation	Zero speed	-
Bit 13	Following error	Homing error	-	-	-

Note: - indicates the bit is invalid.

Object 605Bh: Shutdown option code

Index	605Bh
Name	Shutdown option code
Object code	VAR
Data type	INTEGER16
Access	RW
PDO mapping	Yes
Setting range	INTEGER16
Default	0

Object function:

OD 605Bh = 0: when Servo Off, the dynamic brake has no effect, so the motor runs freely and the machine stops only by friction.

OD 605Bh = -1: when Servo Off, the servo stops with the operation of the dynamic brake.

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 6060h: Modes of operation

Index	6060h
Name	Modes of operation
Object code	VAR
Data type	INTEGER8
Access	RW
PDO mapping	Yes
Setting range	INTEGER8
Default	0

Object function:

This object sets the mode for operation.

Setting value	Mode
0	Reserved
1	Profile Position mode
2	Reserved
3	Profile Velocity mode
4	Profile Torque mode
5	Reserved
6	Homing mode
7	Interpolated Position mode

Object 6061h: Modes of operation display

Index	6061h
Name	Modes of operation display
Object code	VAR
Data type	INTEGER8
Access	RO
PDO mapping	Yes
Setting range	INTEGER8
Default	0

Object function:

This object displays the current operation mode. Refer to the table in OD 6060h.

11

Object 6062h: Position demand value (PUU)

Index	6062h
Name	Position demand value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object function:

This position demand value is the interpolation command calculated by the servo internal interpolator. This command passes through the servo internal filter. For its detailed location, refer to the servo architecture diagram of each mode.

Object 6063h: Position actual internal value (Pulse)

Index	6063h
Name	Position actual internal value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	Pulse (unit for encoder pulse resolution) The ASDA-A2 servo drive generates 1,280,000 pulses per motor revolution. The ASDA-A3 / ASDA-B3 servo drive generates 16,777,216 pulses per motor revolution.

Object 6064h: Position actual value (PUU)

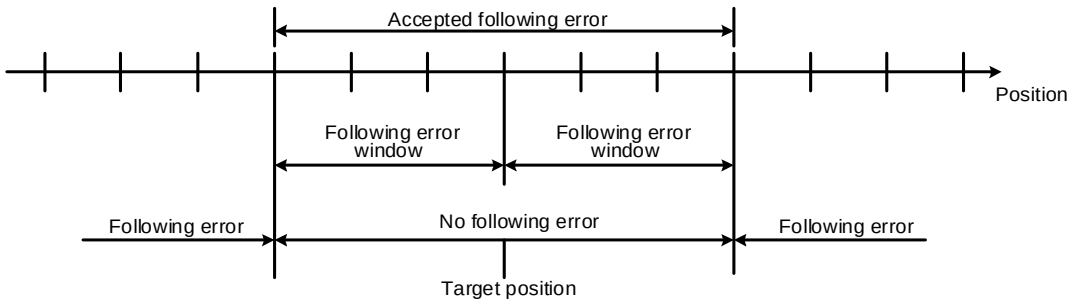
Index	6064h
Name	Position actual value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object 6065h: Following error window

Index	6065h
Name	Following error window
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	50331648
Unit	PUU

Object function:

When the following error actual value (OD 60F4h) exceeds this setting range, AL009 (Excessive deviation of Position command) is triggered.



Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

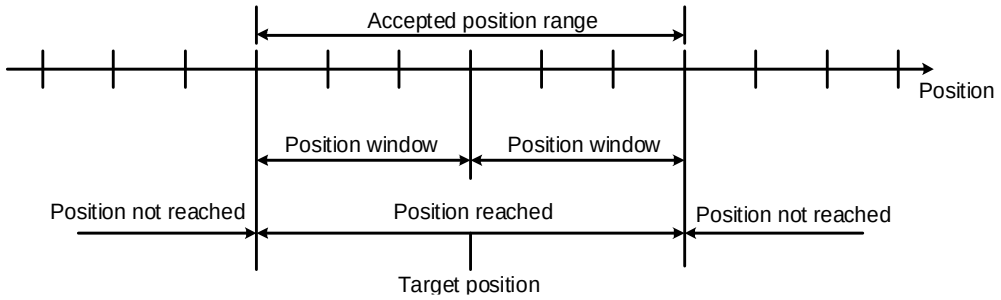
11

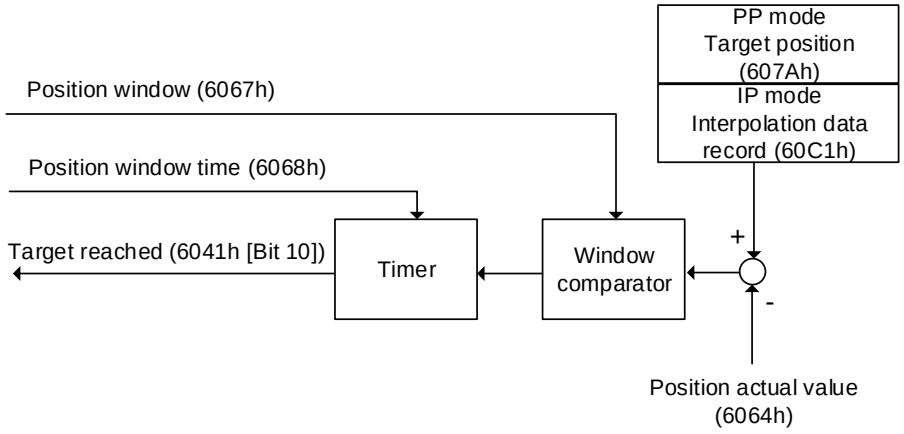
Object 6067h: Position window

Index	6067h
Name	Position window
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	100
Unit	PUU

Object function:

When the difference (absolute value) between the position command (PP mode: OD 607Ah; IP mode: OD 60C1h) and the position actual value (OD 6064h) is within the range set in OD 6067h (Position window), and the duration of this condition is longer than the time set in OD 6068h (Position window time), OD 6041h [Bit 10] (Target reached) is output.



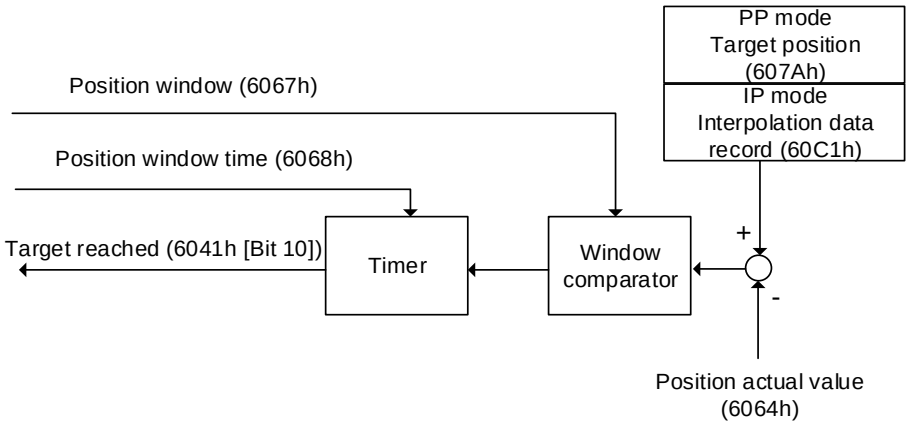


Object 6068h: Position window time

Index	6068h
Name	Position window time
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED16
Default	0
Unit	ms

Object function:

When the difference (absolute value) between the position command (PP mode: OD 607Ah; IP mode: OD 60C1h) and the position actual value (OD 6064h) is within the range set in OD 6067h (Position window), and the duration of this condition is longer than the time set in OD 6068h (Position window time), OD 6041h [Bit 10] (Target reached) is output.



Object 606Bh: Velocity demand value

Index	606Bh
Name	Velocity demand value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0.1 rpm

Object function:

The velocity demand value is the command generated by the speed trajectory generator and filtered by the command filter of the drive. This object only works in Profile Velocity mode.

11

Object 606Ch: Velocity actual value

Index	606Ch
Name	Velocity actual value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0.1 rpm

Object function:

Returns the motor speed at present for monitoring.

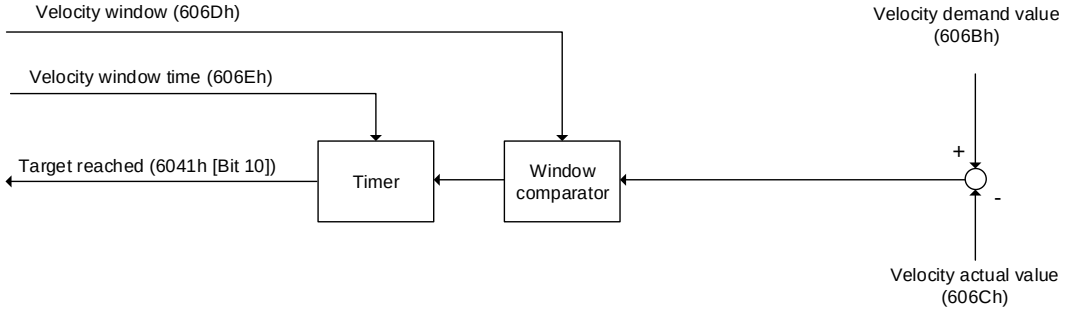
Object 606Dh: Velocity window

Index	606Dh
Name	Velocity window
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	0 - 3000
Default	100
Unit	0.1 rpm

Object function:

The window comparator compares the speed difference with the velocity window (OD 606Dh). When the difference (absolute value) is within the range set in the velocity window and the duration of this condition is longer than the time set in the velocity window time (OD 606Eh), OD 6041h [Bit 10] (Target reached) is output. This object only works in Profile Velocity mode.

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.



Object 606Eh: Velocity window time

Index	606Eh
Name	Velocity window time
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED16
Default	0
Unit	ms

Object function:

Refer to OD 606Dh for the description of the object.

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 606Fh: Velocity threshold

Index	606Fh
Name	Velocity threshold
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	0 - 2000
Default	100
Unit	0.1 rpm

Object function:

This object sets the range for the zero-speed signal output. When the forward or reverse speed (absolute value) of the motor is lower than the setting value of OD 606Fh, OD 6041h [Bit 12] (zero-speed signal) outputs 1.

Note: when P3.012.Z is set to1, the non-volatile setting for this object is enabled.

Object 6071h: Target torque

Index	6071h
Name	Target torque
Object code	VAR
Data type	INTEGER16
Access	RW
PDO mapping	Yes
Setting range	-3500 to +3500
Default	0
Unit	0.1%

Object function:

This object sets the target torque in Profile Torque mode. If OD 6071h = 1000 (100.0%), it corresponds to the motor rated torque.

Object 6072h: Max torque

Index	6072h
Name	Max torque
Object code	VAR
Data type	UNSIGNED16
Access	RW
PDO mapping	Yes
Setting range	0 - 3500
Default	3500
Unit	0.1%

Object function:

This object sets the maximum torque in Profile Torque mode.

Object 6074h: Torque demand value

Index	6074h
Name	Torque demand value
Object code	VAR
Data type	INTEGER16
Access	RO
PDO mapping	Yes
Setting range	INTEGER16
Default	0
Unit	0.1%

Object function:

The torque demand value is the command generated by the speed trajectory generator and filtered by the command filter of the drive. This object only works in Profile Torque mode.

Object 6075h: Motor rated current

Index	6075h
Name	Motor rated current
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED32
Default	0
Unit	mA

Object function:

This object displays the rated current specified on the motor nameplate.

Object 6076h: Motor rated torque

Index	6076h
Name	Motor rated torque
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED32
Default	0
Unit	0.001 N-m

Object function:

This object displays the rated torque specified on the motor nameplate.

Object 6077h: Torque actual value

Index	6077h
Name	Torque actual value
Object code	VAR
Data type	INTEGER16
Access	RO
PDO mapping	Yes
Setting range	INTEGER16
Default	0
Unit	0.1%

Object function:

This object is the motor torque feedback in percentage at present.

Object 6078h: Current actual value

Index	6078h
Name	Current actual value
Object code	VAR
Data type	INTEGER16
Access	RO
PDO mapping	Yes
Setting range	INTEGER16
Default	0
Unit	0.1%

Object function:

This object is the motor current feedback in percentage at present.

Object 607Ah: Target position

Index	607Ah
Name	Target position
Object code	VAR
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object function:

This object only works in Profile Position mode. For more details, refer to Section 11.3.1.

11

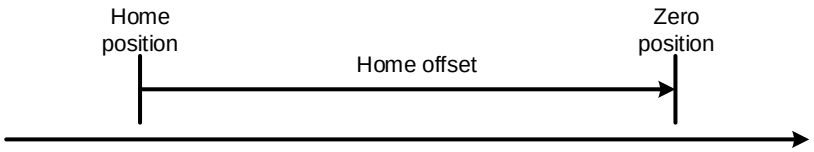
Object 607Ch: Home offset

Index	607Ch
Name	Home offset
Object code	VAR
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object function:

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

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.



Object 607Dh: Software position limit

Index	607Dh
Name	Software position limit
Object code	ARRAY
Data type	INTEGER32
Access	RW

Sub-index	0
Description	Number of entries
Data type	UNSIGNED8
Access	RO
PDO mapping	Yes
Setting range	2
Default	2

Sub-index	1
Description	Min position limit
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	-2147483648 to +2147483647
Default	-2147483648
Unit	PUU

Sub-index	2
Description	Max position limit
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	-2147483648 to +2147483647
Default	+2147483647
Unit	PUU

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 607Fh: Max profile velocity

Index	607Fh
Name	Max profile velocity
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	Varies depending on the motor model
Corresponding servo parameter	P1.055 (rpm) / 10
Unit	0.1 rpm

Object function:

The setting value of OD 607Fh (unit: 0.1 rpm) multiplied by 10 is equivalent to the setting value of P1.055 (Maximum speed limit; unit: 1 rpm).

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 6080h: Max motor speed

Index	6080h
Name	Max motor speed
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	Varies depending on the motor model
Corresponding servo parameter	P1.055
Unit	rpm

Object function:

OD 6080h is equivalent to P1.055 (Maximum speed limit).

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 6081h: Profile velocity

Index	6081h
Name	Profile velocity
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	10000
Unit	PUU/s

Object function:

This object only works in Profile Position mode. For more details, refer to Section 11.3.1.

Object 6083h: Profile acceleration

Index	6083h
Name	Profile acceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 65500
Default	200
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to accelerate from 0 rpm to 3,000 rpm. This object only works in Profile Position mode and Profile Velocity mode.

Object 6084h: Profile deceleration

Index	6084h
Name	Profile deceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 65500
Default	200
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to decelerate from 3,000 rpm to 0 rpm. This object only works in Profile Position mode and Profile Velocity mode.

Object 6085h: Quick stop deceleration

Index	6085h
Name	Quick stop deceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 65500
Default	200
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to decelerate from 3,000 rpm to 0 rpm using the quick stop function.

Object 6087h: Torque slope

Index	6087h
Name	Torque slope
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	0 - 65500
Default	200
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to change from 0% to 100% of the rated torque.

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

Object 6093h: Position factor

Index	6093h
Name	Position factor
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Corresponding servo parameter	P1.044 and P1.045
Note	Position factor = Numerator / Feed_constant

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	2
Default	2

Sub-index	1
Description	E-Gear ratio numerator
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Default	1
Corresponding servo parameter	P1.044
Note	For the E-Gear ratio setting, refer to Section 6.2.5.

Sub-index	2
Description	E-Gear ratio denominator
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Default	1
Corresponding servo parameter	P1.045
Note	For the E-Gear ratio setting, refer to Section 6.2.5.

Note: when P3.012.Z is set to 1, the non-volatile setting for this object is enabled.

11

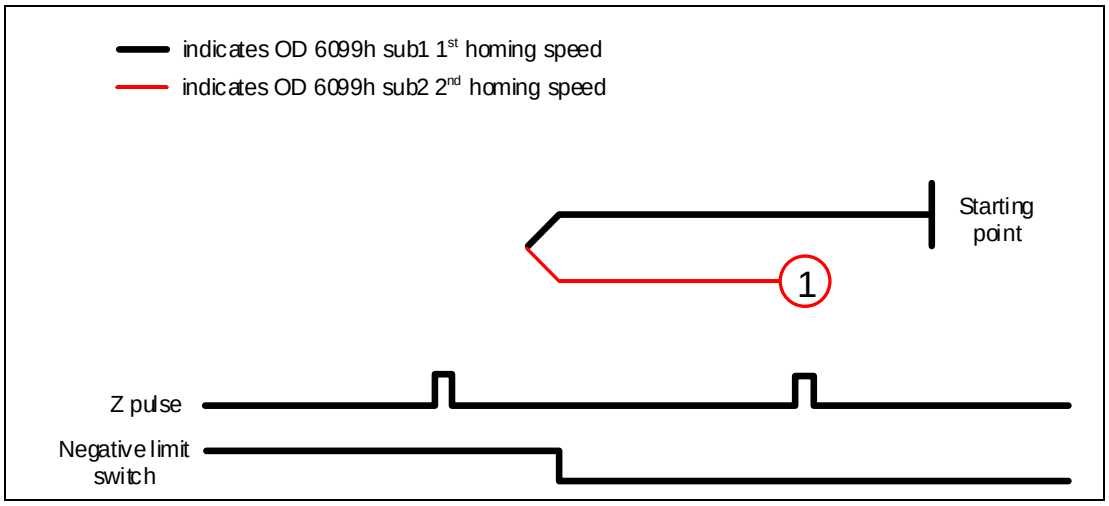
Object 6098h: Homing method

Index	6098h
Name	Homing method
Object code	VAR
Data type	INTEGER8
Access	RW
PDO mapping	Yes
Setting range	-4 to 35
Default	0

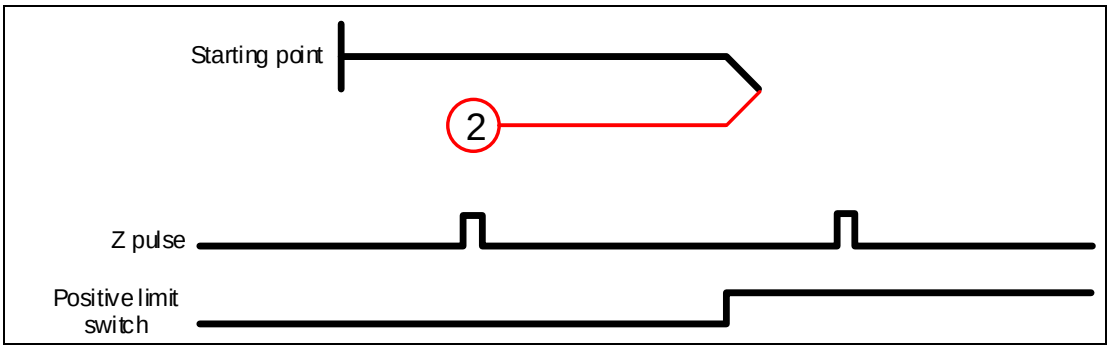
Object function:

The homing methods include looking for the Z pulse (Methods 1 - 14, 33, 34, 36, 37), not looking for the Z pulse (Methods 17 - 30), defining the current position as the origin (Method 35), and looking for the hard stop (Methods 36 - 39). Methods 15, 16, 31, and 32 are reserved. To use Methods 1 to 35, set OD 6098h to 1 to 35. To use Methods 36 to 39, set OD 6098h to -1 to -4.

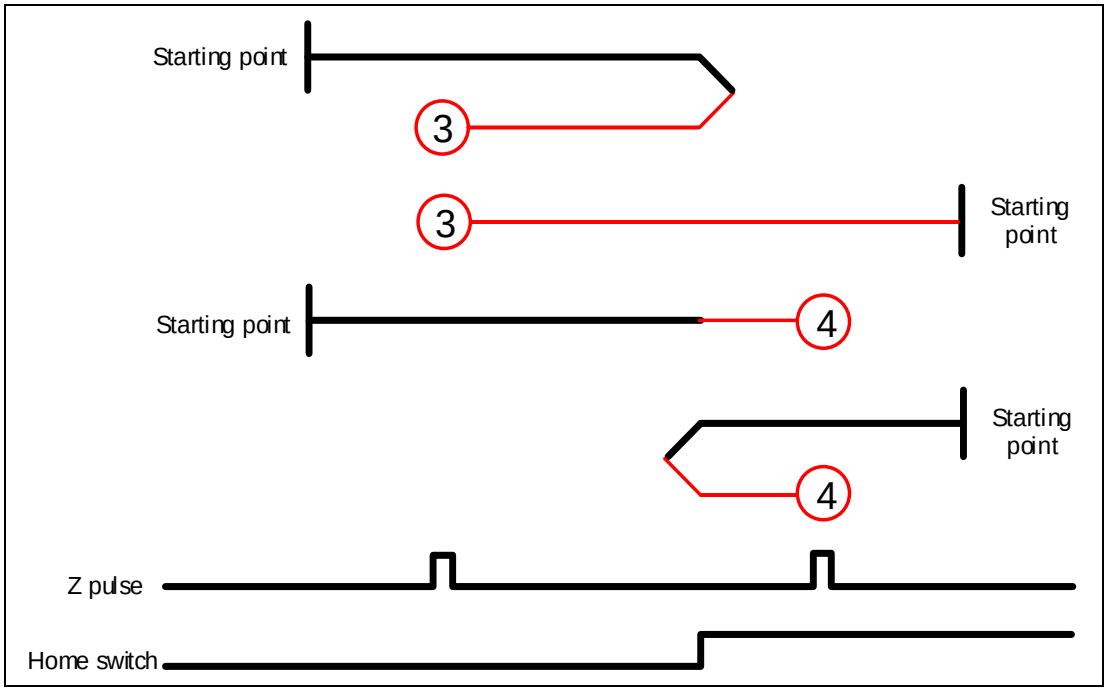
Method 1: homing on the negative limit switch and Z pulse



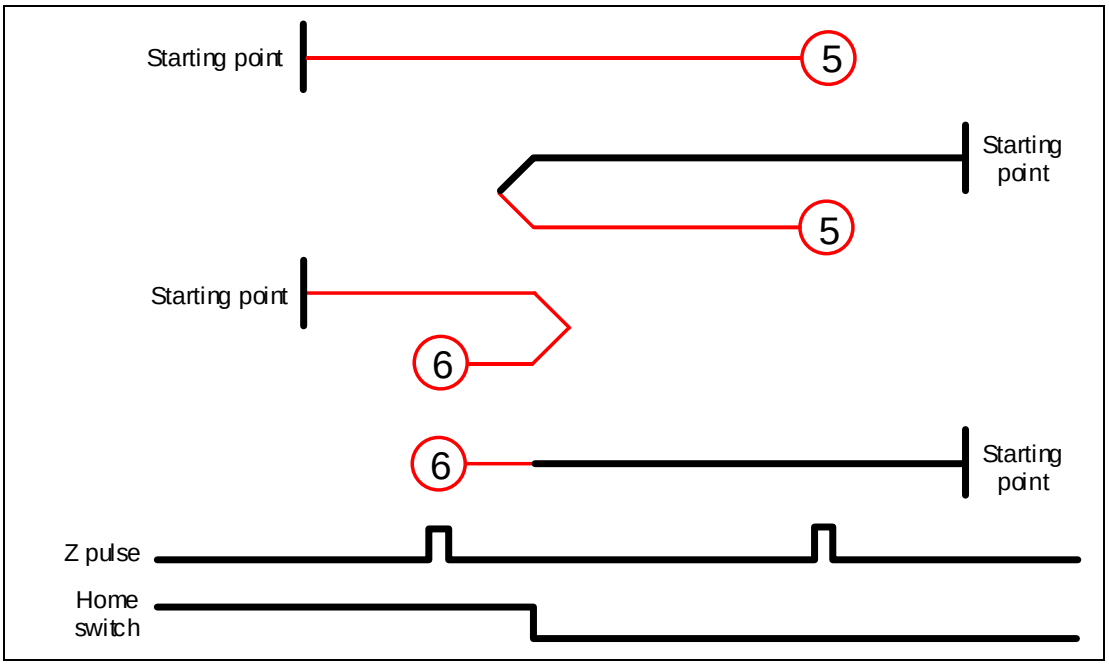
Method 2: homing on the positive limit switch and Z pulse



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

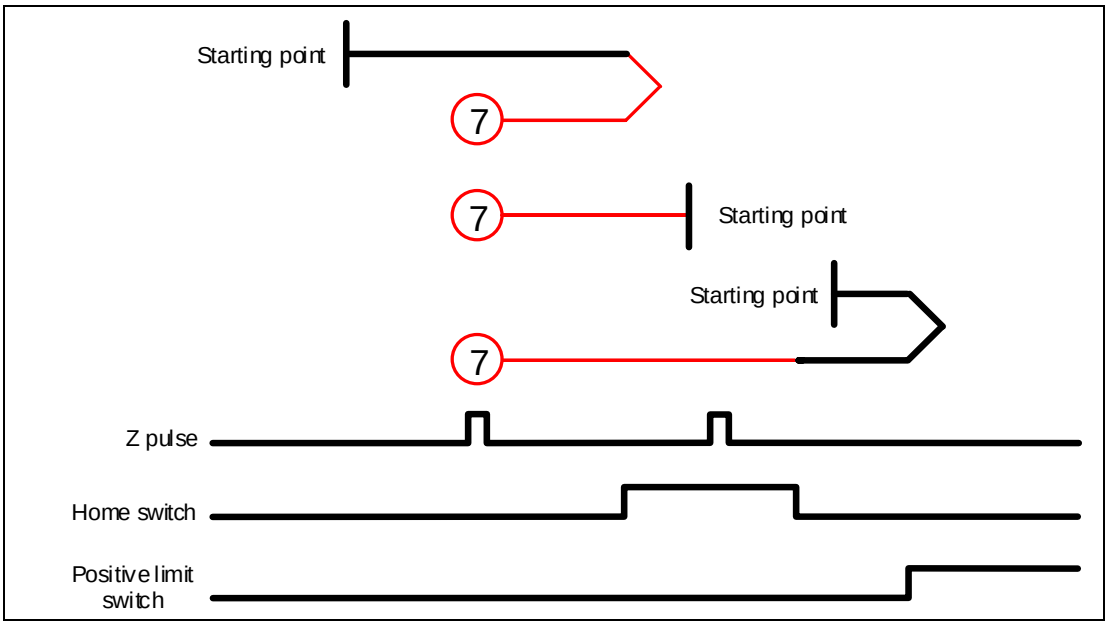


Methods 5 and 6: homing on the home switch and Z pulse

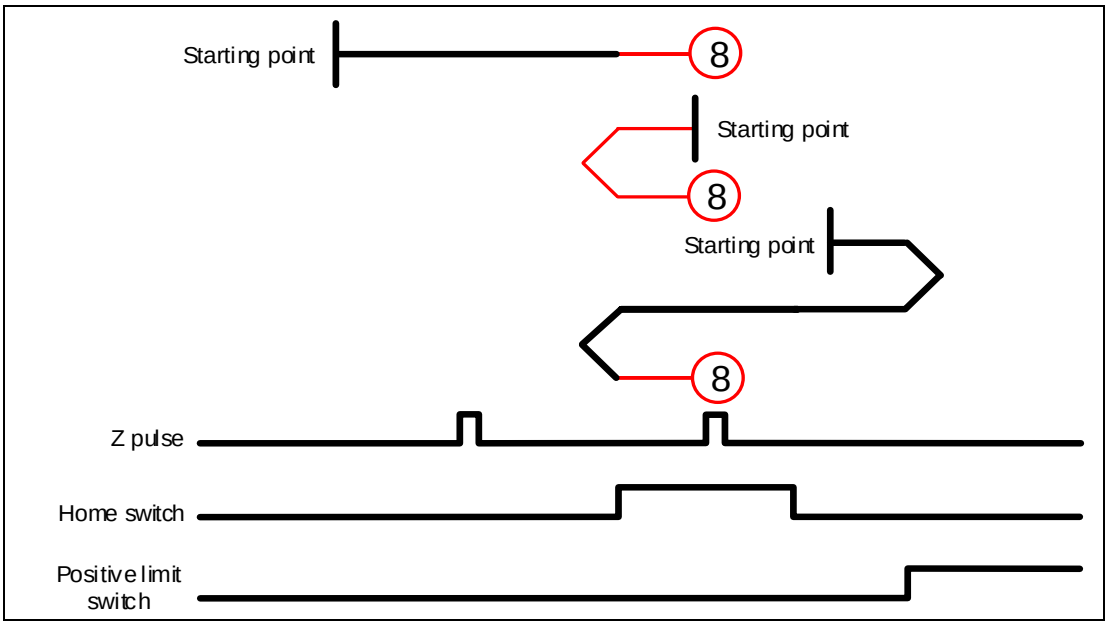


11

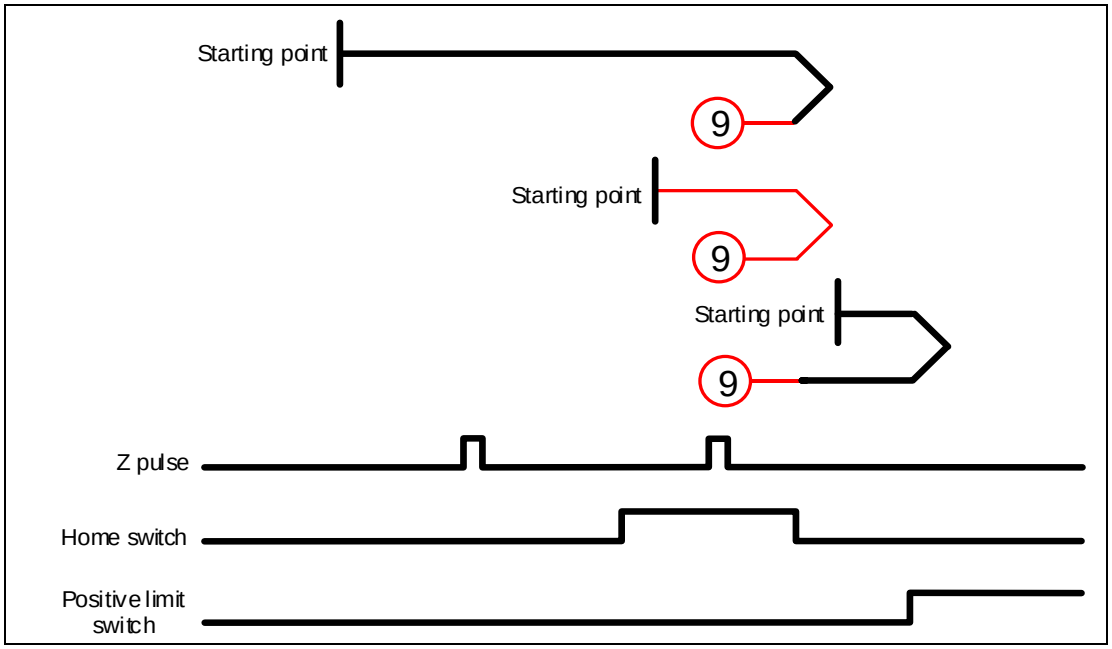
Method 7: homing on the positive limit switch, home switch, and Z pulse



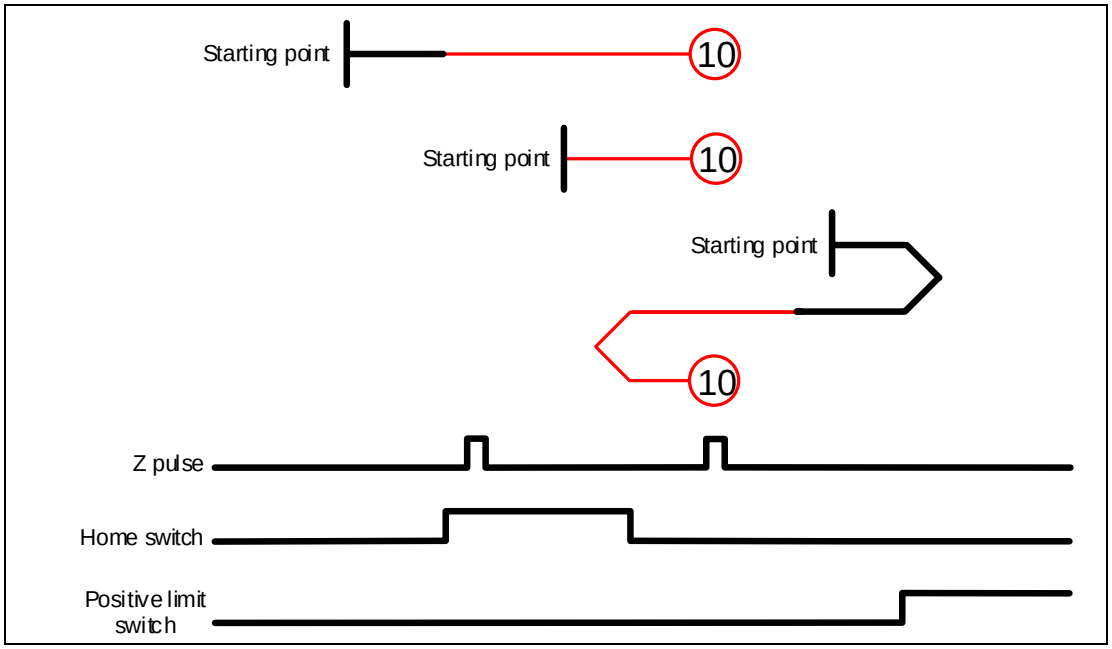
Method 8: homing on the positive limit switch, home switch, and Z pulse



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

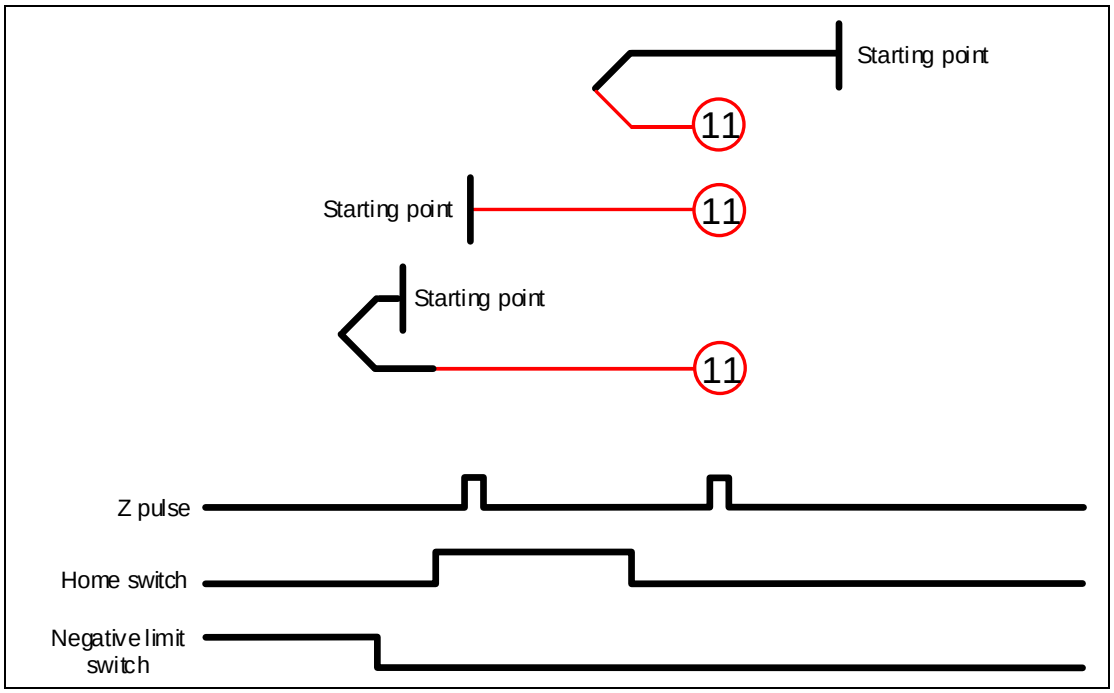


Method 10: homing on the positive limit switch, home switch, and Z pulse

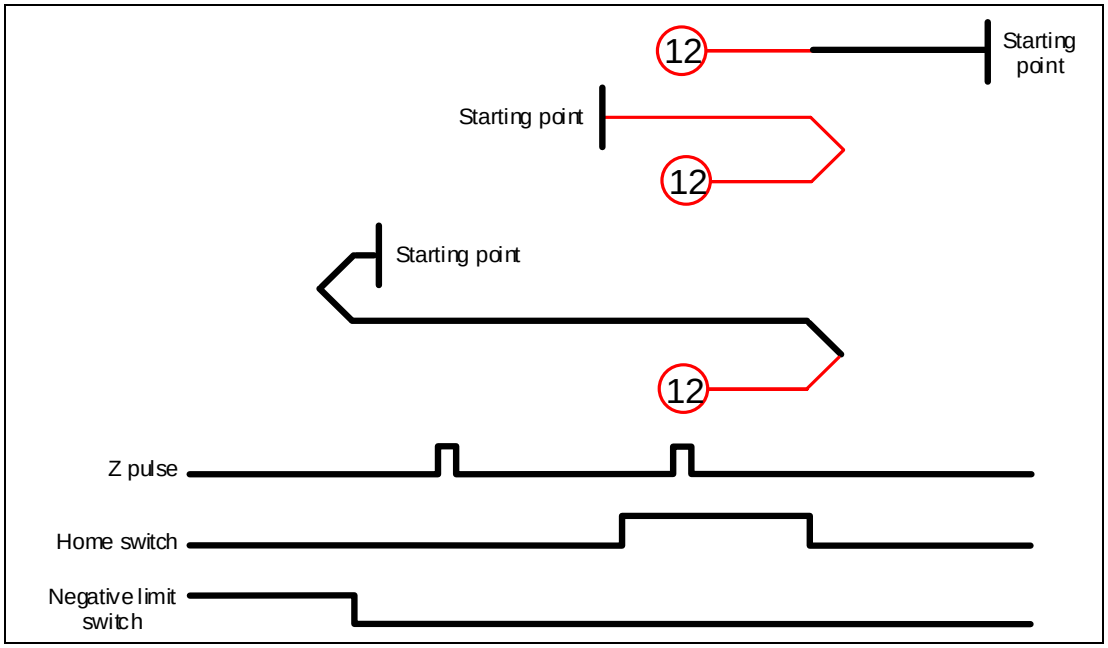


11

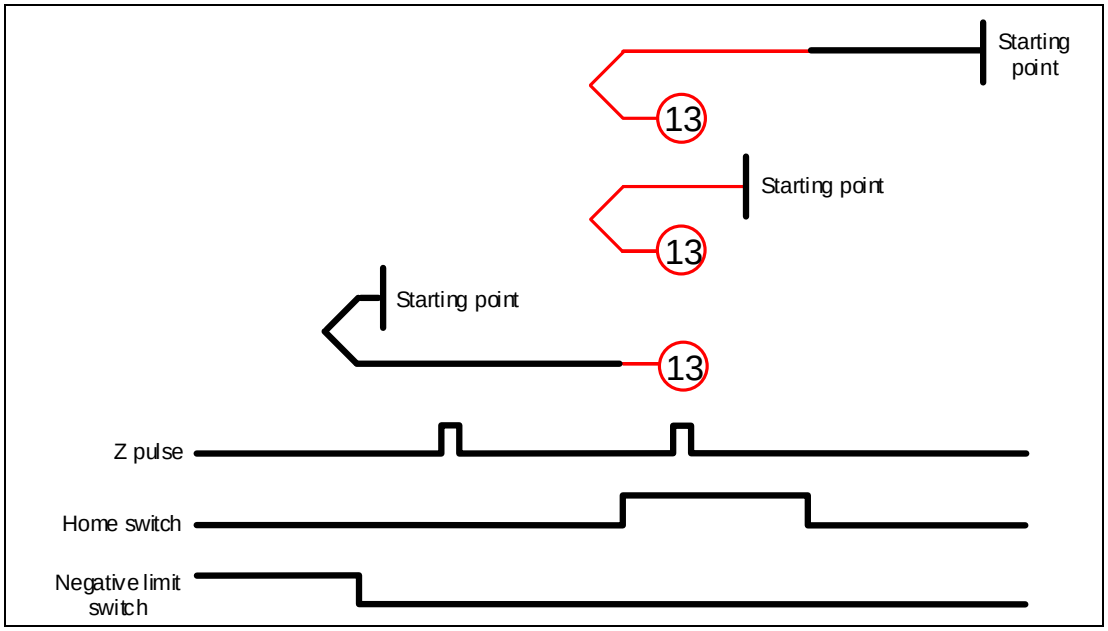
Method 11: homing on the negative limit switch, home switch, and Z pulse



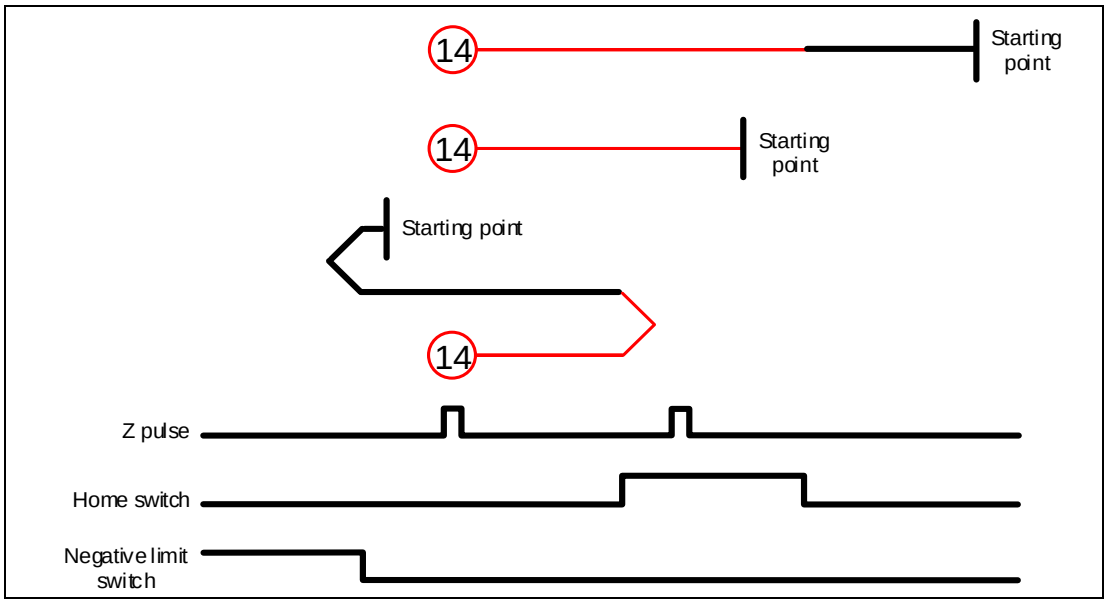
Method 12: homing on the negative limit switch, home switch, and Z pulse



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

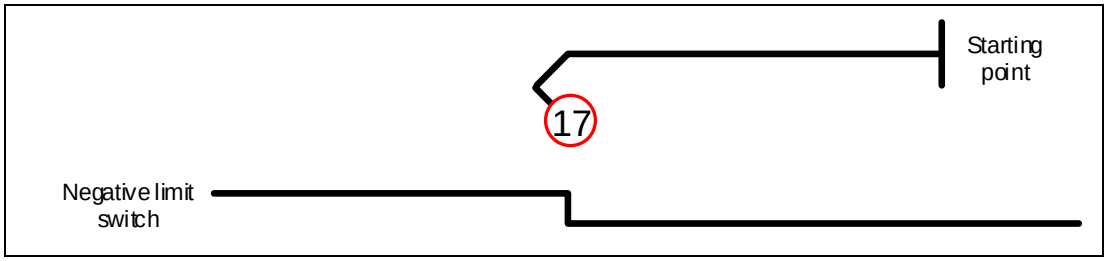


Method 14: homing on the negative limit switch, home switch, and Z pulse



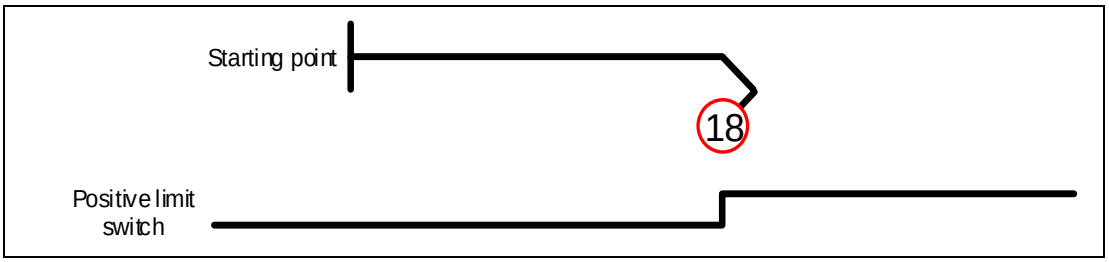
Methods 15 and 16: reserved

Method 17: homing on the negative limit switch

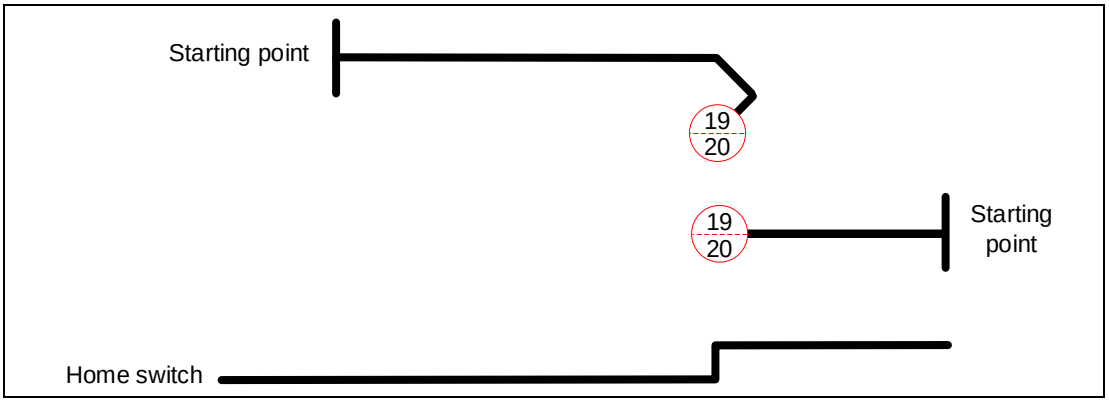


11

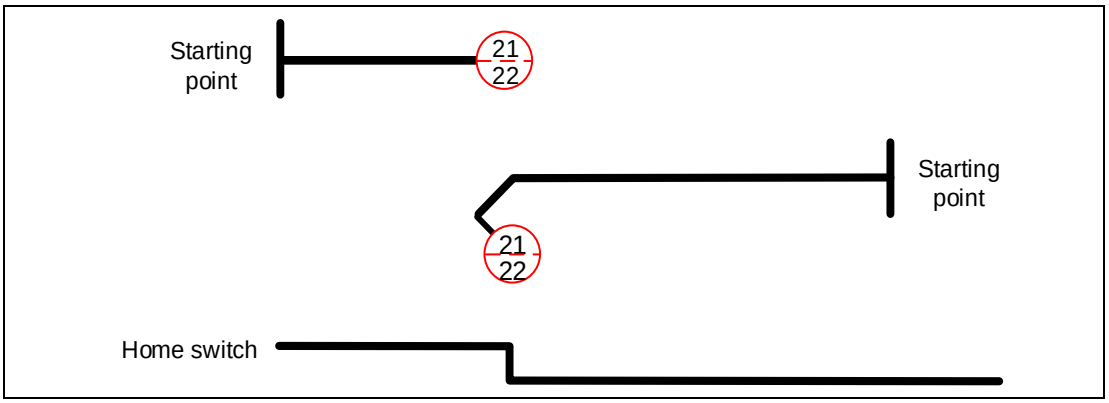
Method 18: homing on the positive limit switch



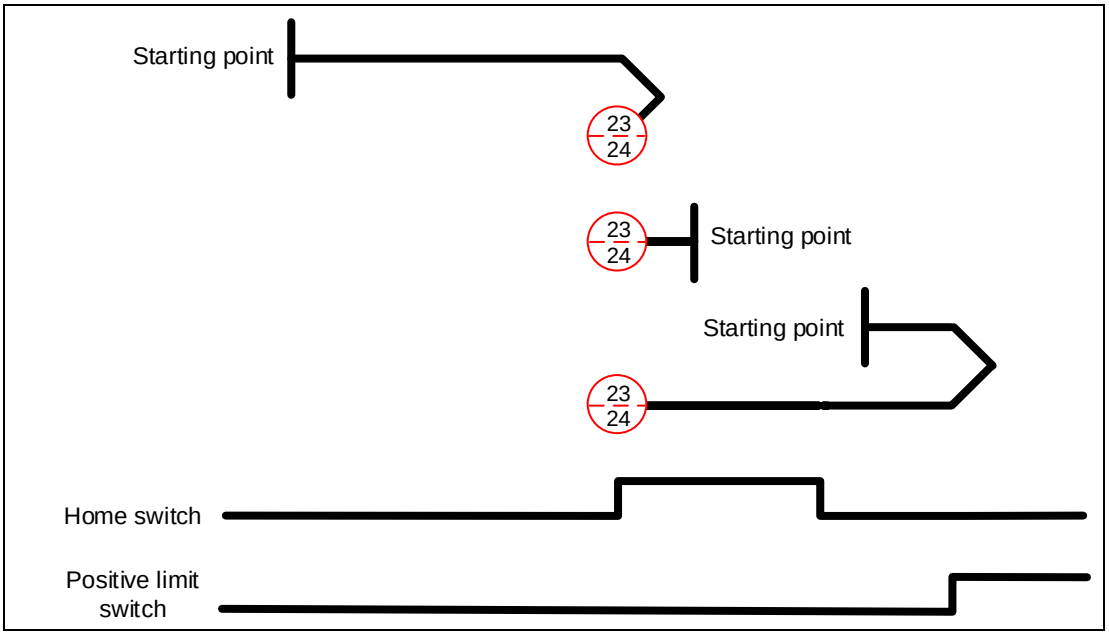
Methods 19 and 20: homing on the home switch



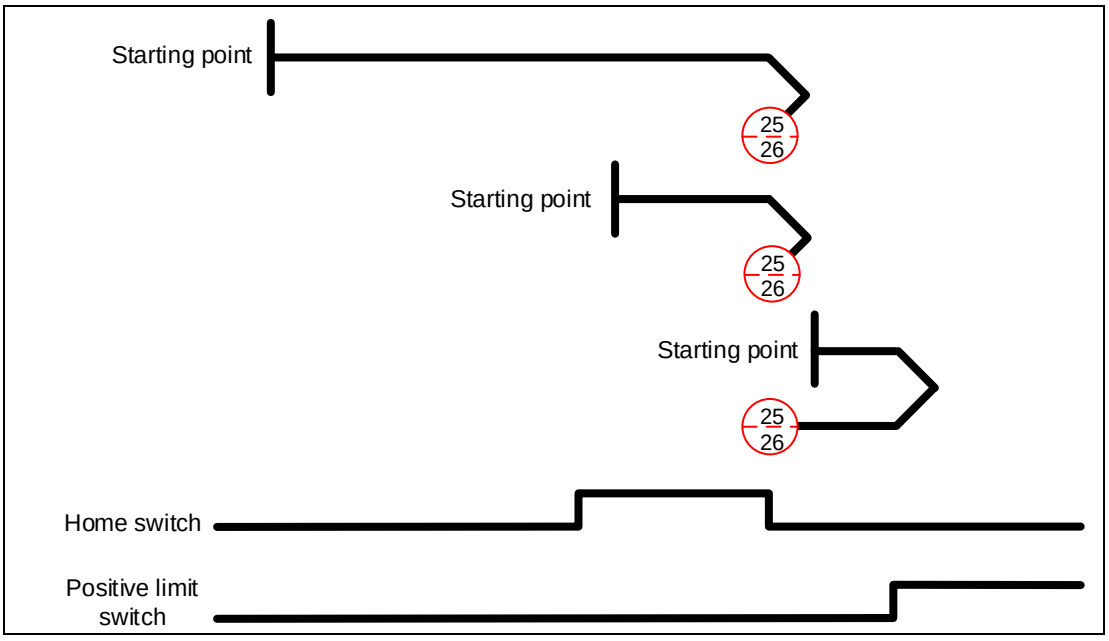
Methods 21 and 22: homing on the home switch



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

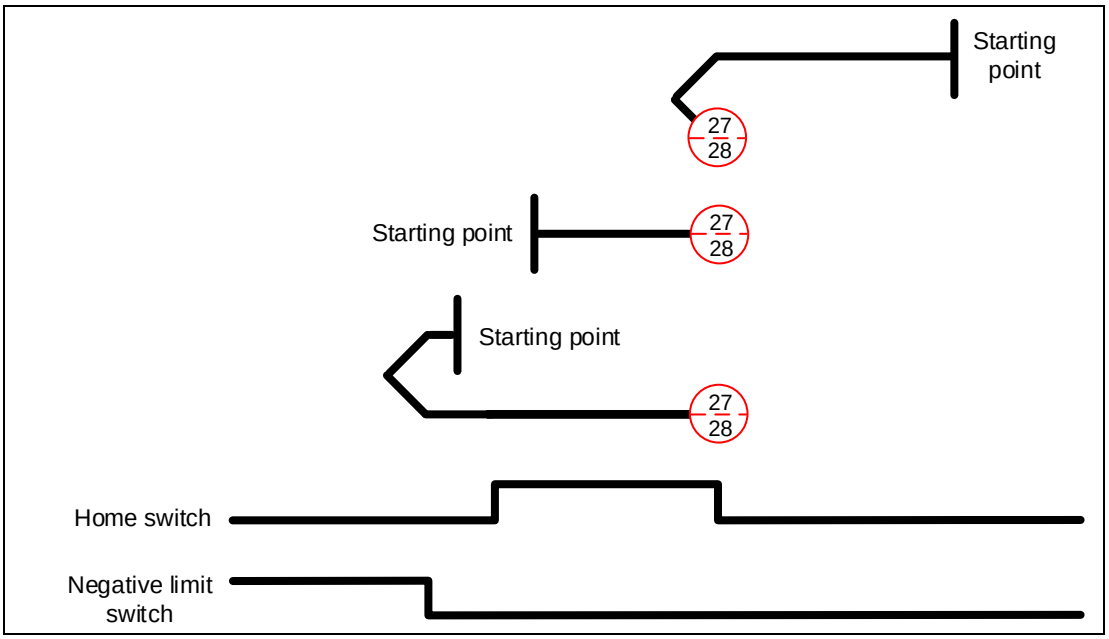


Methods 25 and 26: homing on the positive limit switch and home switch

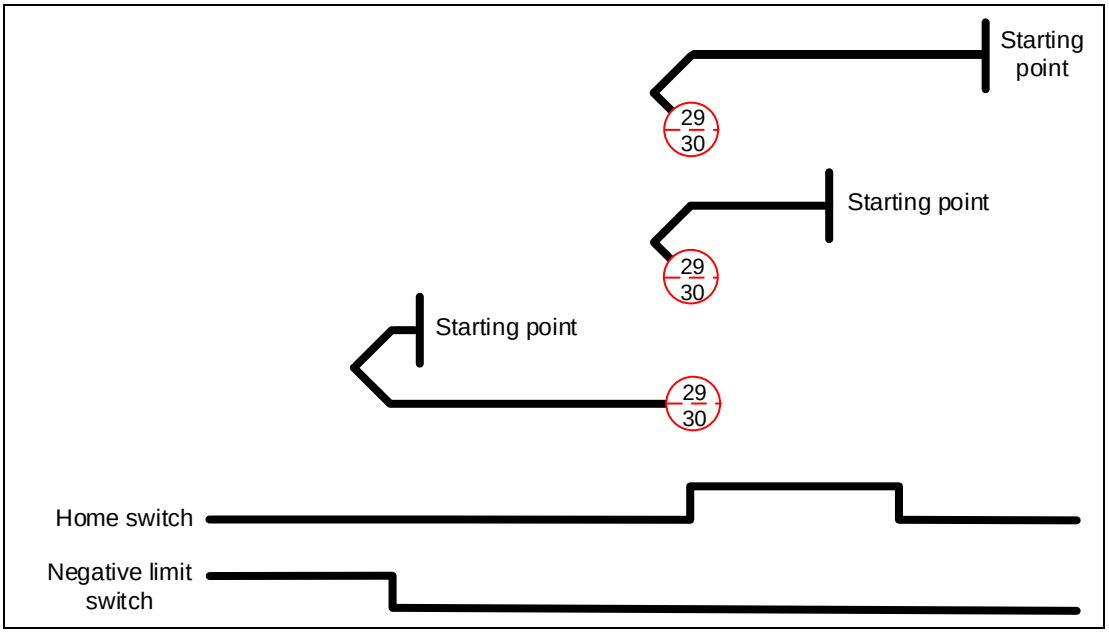


11

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

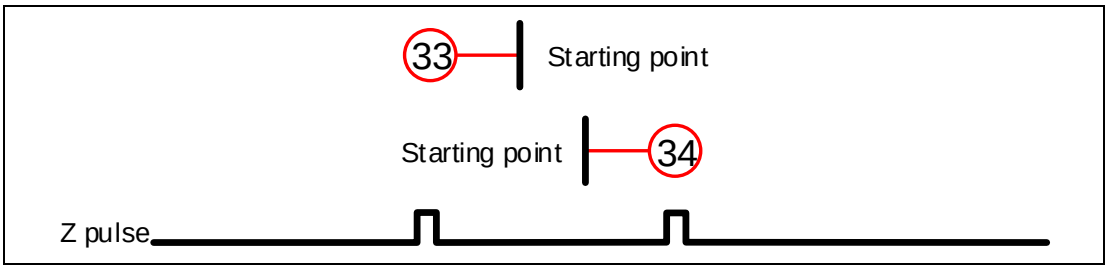


Methods 29 and 30: homing on the negative limit switch and home switch



Methods 31 and 32: reserved

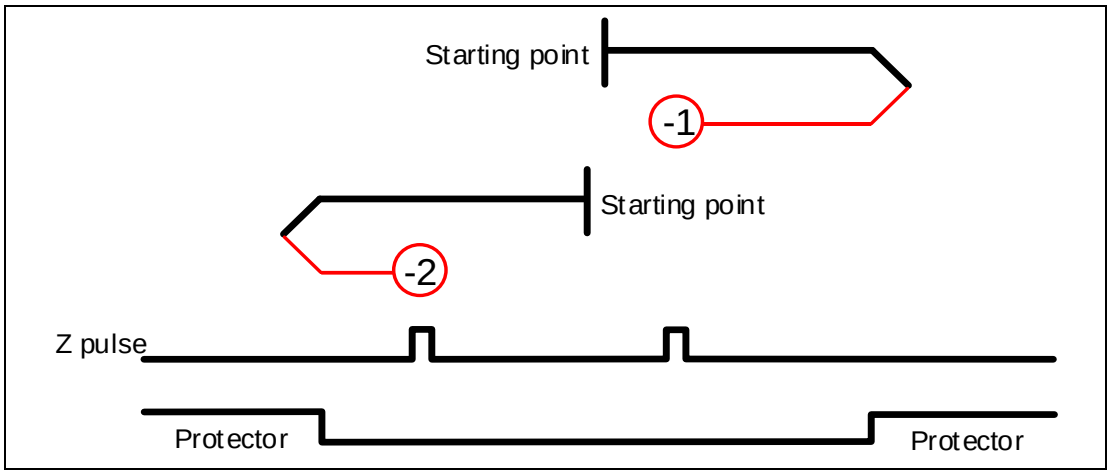
Methods 33 and 34: homing on the Z pulse



Method 35: defines the current feedback position as the origin

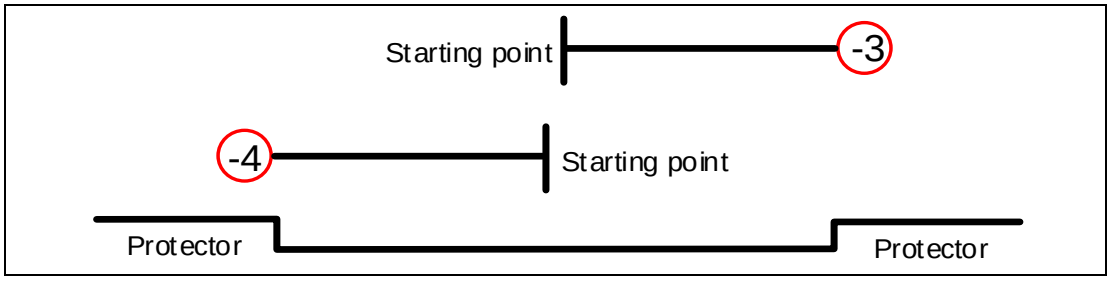
Methods 36 and 37:

When OD 6098h is set to -1 or -2: homing on the hard stop and Z pulse. Set the servo parameters P1.087 (torque level detection) and P1.088 (level reached timer) when using these homing methods.



Methods 38 and 39:

When OD 6098h is set to -3 or -4: homing on the hard stop. Set the servo parameters P1.087 (torque level detection) and P1.088 (level reached timer) when using these homing methods.



11

Object 6099h: Homing speeds

Index	6099h
Name	Homing speeds
Object code	ARRAY
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	Yes
Setting range	2
Default	2

Sub-index	1
Description	Speed during search for switch
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 20000
Default	100
Unit	0.1 rpm

Sub-index	2
Description	Speed during search for zero
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 5000
Default	20
Unit	0.1 rpm

Object 609Ah: Homing acceleration

Index	609Ah
Name	Homing acceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED32
Default	100
Unit	ms

Object function:

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

Object 60C0h: Interpolation sub mode select

Index	60C0h
Name	Interpolation sub mode select
Object code	VAR
Data type	INTEGER16
Access	RW
PDO mapping	Yes
Setting range	INTEGER16
Default	0

Object function:

No need to set this object.

Object 60C1h: Interpolation data record

Index	60C1h
Name	Interpolation data record
Object code	ARRAY
Data type	INTEGER32
Access	RW
PDO mapping	Yes

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	2
Default	2

Sub-index	1
Description	Command position Pos_Cmd
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object function:

The PDO sets OD 60C1h every T ms until the PDO receives the SYNC message. The value of T is determined by OD 60C2h sub1. This object only works in Interpolated Position mode. For more details, refer to Section 11.3.2.

Object 60C2h: Interpolation time period

Index	60C2h
Name	Interpolation time period
Object code	RECORD
Data type	UNSIGNED8
Access	RW
PDO mapping	Yes

Sub-index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	No
Setting range	2
Default	2

Sub-index	1
Description	Interpolation time units
Data type	UNSIGNED8
Access	RW
PDO mapping	Yes
Setting range	UNSIGNED8
Default	1

Sub-index	2
Description	Interpolation time index
Data type	INTEGER8
Access	RW
PDO mapping	Yes
Setting range	-128 to +63
Default	-3

Object function:

This object only works in Interpolated Position mode. The interpolation time period is calculated by OD 60C2h sub1 and OD 60C2h sub2. The calculation is as follows:

$$\text{Interpolation time period} = \text{OD 60C2h sub1} \times 10^{\text{OD 60C2h sub2}}$$

Example:

If you want to set the interpolation time period to 2 ms, set OD 60C2h sub1 to 2 and OD 60C2h sub2 to -3.

$$\text{Interpolation time period} = 2 \times 10^{-3} = 0.002 \text{ s} = 2 \text{ ms}$$

Object 60C5h: Max acceleration

Index	60C5h
Name	Max acceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 65500
Default	1
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to accelerate from 0 rpm to 3,000 rpm.

Object 60C6h: Max deceleration

Index	60C6h
Name	Max deceleration
Object code	VAR
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	1 - 65500
Default	1
Unit	ms

Object function:

The time slope set by this object is the time required for the motor to decelerate from 3,000 rpm to 0 rpm.

Object 60F4h: Following error actual value

Index	60F4h
Name	Following error actual value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	PUU

Object function:

The following error actual value is the difference between the position demand value (OD 6062h) and position actual value (OD 6064h). For more details, refer to the architecture diagrams in Section 11.3.

Object 60FCh: Position demand value

Index	60FCh
Name	Position demand value
Object code	VAR
Data type	INTEGER32
Access	RO
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	pulse

Object function:

This command is generated after being processed by the servo drive filter. For more details, refer to the architecture diagrams in Section 11.3.

11

Object 60FDh: Digital inputs

Index	60FDh
Name	Digital inputs
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED32
Default	0
Unit	-

Object function:

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function
Bit 0	Negative limit signal
Bit 1	Positive limit signal
Bit 2	Homing signal
Bit 3 - Bit 15	Reserved

Object 60FEh: Digital outputs

Index	60FEh
Name	Digital outputs
Object code	ARRAY
Data type	UNSIGNED32
Access	RW

Sub-Index	0
Description	Number of sub-index
Data type	UNSIGNED8
Access	RO
PDO mapping	Yes
Setting range	2
Default	2

Sub-Index	1
Description	Physical outputs
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	0x00000000 to 0xFFFFFFFF
Default	0

Sub-Index	2
Description	Bit mask
Data type	UNSIGNED32
Access	RW
PDO mapping	Yes
Setting range	0x00000000 to 0xFFFFFFFF
Default	0

Object function:

OD 60FEh sub1 (Physical outputs)

Bit	DO	Description
0 - 15	-	Reserved
16	DO1	0: off; 1: on
17	DO2	0: off; 1: on
18	DO3	0: off; 1: on
19	DO4	0: off; 1: on
20 - 31	-	Reserved

OD 60FEh sub2 (Bit mask)

Bit	DO	Description
0 - 15	-	Reserved
16	DO1	0: disable physical outputs; 1: enable
17	DO2	0: disable physical outputs; 1: enable
18	DO3	0: disable physical outputs; 1: enable
19	DO4	0: disable physical outputs; 1: enable
20 - 31	-	Reserved

- To use the software to control the DO output, you must first set the corresponding DO function code.
  When P2.018 = 0x0130, the output of DO1 is controlled by the software.
  When P2.019 = 0x0131, the output of DO2 is controlled by the software.
  When P2.020 = 0x0132, the output of DO3 is controlled by the software.
  When P2.021 = 0x0133, the output of DO4 is controlled by the software.

- DO output settings
  When the corresponding OD 60FEh sub2 bit of the DO is set to 1, the output status of this DO is determined by the corresponding bit of OD 60FEh sub1.
  When the corresponding OD 60FEh sub2 bit of the DO is set to 0, the output status of this DO is determined by P4.006.

- Example:
  1. Set P2.018 to 0x0130, which means the output of DO1 is controlled by the software.
  2. When OD 60FEh sub2 [Bit 16] is 1, the output status of DO1 is determined by OD 60FEh sub1 [Bit 16].
     When OD 60FEh sub2 [Bit 16] is 0, the output status of DO1 is determined by P4.006 [Bit 0].

Object 60FFh: Target velocity

Index	60FFh
Name	Target velocity
Object code	VAR
Data type	INTEGER32
Access	RW
PDO mapping	Yes
Setting range	INTEGER32
Default	0
Unit	0.1 rpm

Object function:

This object sets the target velocity. This object only works in Profile Velocity mode.

Object 6502h: Supported drive modes

Index	6502h
Name	Supported drive modes
Object code	VAR
Data type	UNSIGNED32
Access	RO
PDO mapping	Yes
Setting range	UNSIGNED32
Default	6Dh

Object function:

This object is read-only and provides the operation modes supported by Delta servo drives in CANopen mode.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Function
Bit 0	Profile Position mode
Bit 1	Reserved
Bit 2	Profile Velocity mode
Bit 3	Profile Torque mode
Bit 4	Reserved
Bit 5	Homing mode
Bit 6	Interpolated Position mode
Bit 7 - Bit 31	Reserved

11.5 Diagnostics and troubleshooting

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

1. The SYNC communication cycle of the controller and servo drive is different

   Since the jitter of each controller is different, the time the servo drive receives the SYNC differs from the SYNC communication cycle time. When this happens, adjust the value of P3.009.U to increase the error range and let the servo drive automatically correct the internal timer so it is consistent with the communication cycle of the controller.

2. Eliminate interference

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