Chapter 11 Communication

11.1 485 Communication

The servo drive's upper computer communication uses the standard Modbus protocol based on the 485 interface.

Modbus is a serial, asynchronous communication protocol with a common language for its application to PLCs or other controllers. This protocol defines a message structure that a controller can recognize and use, regardless of the network via which it is transmitted. The Modbus protocol does not require a dedicated interface, and the typical physical interface is RS485.

The function codes of the servo drives are divided into 16-bit and 32-bit according to the data length. The Modbus RTU protocol enables data reading and writing operations to the function codes, and the command codes differ according to the data length when writing the function codes.

Table 11- 1

Command code	Description
03h	Read 16/32-bit function codes
06h	Read 16-bit function codes
10h	Write 32-bit function codes

11.2 Canopen Communication

11.2.1 Canopen Performance Parameter

Table 11- 2 CAN performance parameter description

Designation	Description
Link layer protocol	CAN bus
Application layer protocol	Canopen protocol
CAN-ID type	11bit-CAN2.0A
Baud rate	1Mbit/s(default)、500Kbit/s、250 Kbit/s、125Kbit/s、100 Kbit/s、50 Kbit/s、20 Kbit/s
Max. node number	63
CAN frame length	0~8
Application layer CAN frame type	Standard frame
Terminal resistance	120Ω
Sub-protocol supported	CiA-301：Canopen application layer and communication protocols
Services supported	NMT: Network Management Terminal SDO: Service Data Object PDO: Process Data Object SYNC: Synchronization
PDO transmission type	Time & event trigger, synchronous trigger
PDO data supported	RPDO x4, TPDO x4
SDO transmission method	Accelerated SDO transmission
Servo operation mode supported	Contour position mode Contour speed mode Contour torque mode Homing method Interpolation mode

The Canopen communication function of the servo drive supports the following different baud rates. The communication distance is related to baud rate and the communication cable.

Table 11- 3 Supported baud rate description

Data transmission rate	Bus length
1 Mbit/s	25
500kbit/s	100
250kbit/s	250
125kbit/s	500
50kbit/s	1000
25kbit/s	2500

Table 11- 4 Relationship among CAN communication transmission distance, rate and nodes

No.	Transmission distance	Speed rate	Node	Wire diameter
①	25m	1Mbps	64	0.205mm^2
②	95m	500Kbps	64	0.34mm^2
③	560m	100Kbps	64	0.5mm^2
④	1100m	50Kbps	64	0.75mm^2

11.2.2 Communication Object

(1)SDO (Service Data Object)

① R-SDO (Receive- Service Data Object)and T-SDO (Transmit- Service Data Object);

② Customers access to the device object dictionary via SDO when using indexes and sub-indexes;

③ Each SDO request and response message contains 8 bytes;

④ SDO is implemented through the CMS object in CAL, which can transmit data of different byte lengths and will actively split into groups of messages when the data exceeds 4 bytes.

(2)PDO (Process Data Object)

① R-PDO (Receive- Service Data Object) and T-PDO(Transmit- Service Data Object);

② PDO data transmit 1 to 8 bytes real-time data to one or more receivers;

③ The communication parameters corresponding to the PDO determine synchronous or asynchronous transmission;

④ Each Canopen device contains four transmission PDO channels and four receiving PDO channels.

(3)SYNC (Synchronization)

The synchronizated object is a message broadcast periodically to the CAN bus by the Canopen master to implement the basic network clock signal. Each device can decide whether to use this event to synchronize communication with other network devices according to its own configuration.

(4)NMT(Network Management Terminal)

NMT includes boot-up messages, Heartbeat protocols and NMT messages. Based on master-slave communication mode, NMT is used to manage and monitor each node in the network mainly for three functions: node status control, error control and node startup.

(5)EMCY(Emergency Message)

Messages sent when inner device communication faulte or application failure error occures.

11.2.3 Network Parameter Configuration

11.2.3.1 Communication Object Identifier

The Communication Object Identifier (COB-ID) specifies object priority and object identification during communication. COB-ID corresponds to the 11-bit frame ID in CAN, and the 11-bit COB-ID consists of two parts, a 4-bit object function code and a 7-bit node address, as shown in Table 11-5.

Table 11- 5 COB-ID composition description

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

Each communication object of Canopen has a default COB-ID, which can be read by SDO and partially modified by SDO. The list of objects is shown in Table 11-6 below.

Table 11- 6 COB-ID

Communication object	Function code	Node address	COB-ID	Corresponding object index
NMT	0000b	0	0h	-
Synchronized objects	0001b	0	80h	1005h,1006h
Emergency message	0001b	0~127	80h+Node-ID	1014h
TPDO1	0011b	0~127	180h+Node-ID	1800h
RPDO1	0100b	0~127	200h+Node-ID	1400h
TPDO2	0101b	0~127	280h+Node-ID	1801h
RPDO2	0110b	0~127	300h+Node-ID	1401h
TPDO3	0111b	0~127	380h+Node-ID	1802h
RPDO3	1000b	0~127	400h+Node-ID	1402h
Communication object	**Function code**	**Node address**	**COB-ID**	**Corresponding object index**
TPDO4	1001b	0~127	480h+Node-ID	1803h
RPDO4	1010b	0~127	500h+Node-ID	1403h
T-SDO	1011b	0~127	580h+Node-ID	1200h
R-SDO	1100b	0~127	600h+Node-ID	1200h
NMT error	1110b	0~127	700h+Node-ID	1016h,1017h

Example: COB-ID of the R-SDO of No. 2 slave node is 600h+2h=602h

11.2.3.2 System Parameter Setting

In order to enable the servo drive to access the Canopen fieldbus network, the relevant function codes of the servo drive need to be set. It is necessary to set Pn000=0.

Table 11- 7 Function code table of system setting

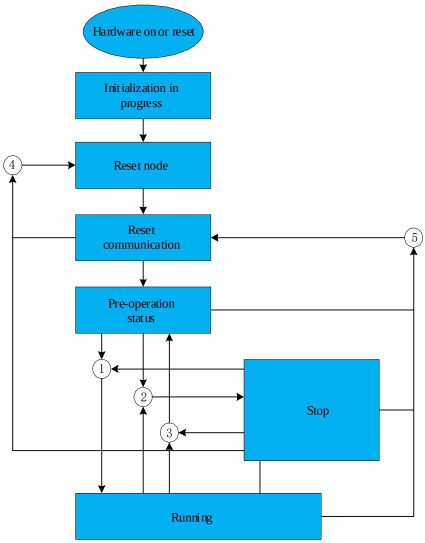
Function code	Designation	Setting range	Value
Pn208	Internal command reception selection	0: External pulse input 1: Internal position command 2: Electronic cam 3: Reserve	4

		4: CanOpen	
Pn080	Can Node-ID	1~127	1(Default)
Pn083	Can communication baud rate	0：20kbit/s；1：50kbit/s 2：100kbit/s；3：125kbit/s 4：250kbit/s；5：500kbit/s 6：1 Mbit/s	4(Default)

11.2.3.3 NMT Service

The Network Management System (NMT), part of the master-slave system, is responsible for initializing, starting and stopping the network. There is and only one Network Management System (NMT) host in the entire Canopen network that can configure the Canopen network including itself. Part of this conversion is automatically implemented internally and part of it must be implemented by the NMT messages sent from host.

Figure 11- 1 NMT status



The Network Management System (NMT) message format is shown in Table 11-8.

Table 11- 8 NMT message format

COB-ID	RTR	Data (bytes)	
		0	1
0x000	0	Command word	Node-ID

COB-ID of NMT message is fixed to "0x000".

Data area consists of two bytes, the first one is a command word indicating the control role of that frame, as shown in Table 11-9.

Table 11- 9 NMT message command

Command word	Function code	Description
01h	①	Run command (all networks are running)
02h	②	Stop command (only NMT works in the whole network)
80h	③	Pre-run command (only SDO, heartbeat, NMT work)
81h	④	Reset node command
82h	5	Reset communication command

The second byte is the node address of Canopen, when it is "0", it is a broadcast message, which is valid for all slave devices in the network.

Table 11- 10 Status table

	Initialize	Pre-run	Run	Stop
PDO			○	
SDO		○	○	
SYNC		○	○	
EMCY		○	○	
Boot-Up	○			
NMT		○	○	○

Note: ○ means vaild

Example: If the SDO operation of the drive is turned on (drive node address is 1), a command word of 80H can be sent.

Table 11- 11

Frame format	COB-ID	RTU	0	1	2	3	4	5	6	7
Data frame	**00**	**0**	**80**	**01**	**-**	**-**	**-**	**-**	**-**	**-**

11.2.3.4 NMT Error Control

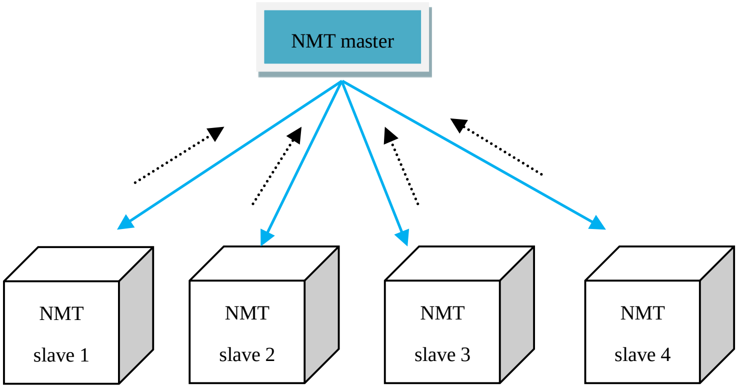
NMT error control is mainly used to detect whether the devices in the network are online and their status including node protection/life protection and heartbeat. In practice, simultaneous life protection and heartbeat are prohibited, and the time of node protection/life protection and heartbeat should not be set too short to avoid increased network load.

(1) Node/lifetime protection

Node protection is that the NMT master periodically checks the NMT slaves' status via remote frames; lifetime protection is that the slaves indirectly monitor the status of the master via the interval of remote frames which are received originally to monitor the slave. Node protection follows a master-slave model, where each remote frame must be answered.

The objects associated with node/lifetime protection include the protection time 100Ch and the lifetime factor 100Dh. The value of 100Ch is the node protection remote frame interval in ms under normal conditions, and the product of 100Ch and 100Dh determines the latest time for host queries. Under normal conditions, node protection is achieved. Lifetime protection is activated when both node 100Ch and 100Dh are not 0 and a node protection request frame is received.

Figure 11- 2



The NMT master sends the remote frame of node protection every 100Ch, and the slave must response, otherwise the slave is considered to be disconnected; if the slave does not receive the node protection remote frame within 100Ch × 100Dh, the master is considered to be disconnected. The NMT master sends remote frames in the format shown in Table 11-12.

Table 11- 12 Remote frame messages of node protection

COB-ID	RTR
0x700 + Node-ID	1

The response messages returned from the NMT slaves are shown in Table 11-13.

Table 11- 13 Node protection response messages

COB-ID	RTR	Data
0x700 + Node-ID	0	status word

Data segment is a one-byte status word with the data format shown in Table 11-14.

Table 11- 14 Data segment description

Data bit	Description
bit7	alternate "0" and "1" each time
bit6~0	4: in stop state 5: in running state 127: in pre-running state

(2) Heartbeat

Heartbeat is a producer-consumer model.

The Canopen device can send heartbeat messages according to the period set by the producer heartbeat interval object 1017h in ms. The node in the CAN network with the consumer heartbeat function monitors this producer according to the consumer time set by object 1016h and considers the node to be faulty once the producer heartbeat of the corresponding node is not received within the consumer heartbeat time range.

After configuring the producer heartbeat time interval 1017h, the node heartbeat function is activated and starts generating heartbeat messages. After configuring a valid subindex of consumer heartbeat 1016h, monitoring starts after a frame of heartbeat from the corresponding node is received.

Master sends heartbeat messages according to its producer time. If the slave of the monitoring master does not receive a heartbeat message within the object 1016h subindex time, the matster is considered disconnected. The object 1016h subindex time ≥ host producer time × 2, otherwise it causes the slave to mistakenly judge that the matster is disconnected.

Each object in 1017h time of the slave sends a heartbeat message to the master that monitors the slave, and if the heartbeat message is not received within the consumer time, the slave is considered to be disconnected. The heartbeat message format is shown in Table 11-18.

Table 11- 15 Heartbeat message format

COB-ID	RTR	Data
0x700 + Node-ID	0	status word

The data segment has only one byte, and the highest bit is fixed to "0".

Table 11- 16 Data segment description

Data bit	Description
bit7	fixed to "0"
bit6~bit0	4: in stop state 5： in running state 127: in pre-running state

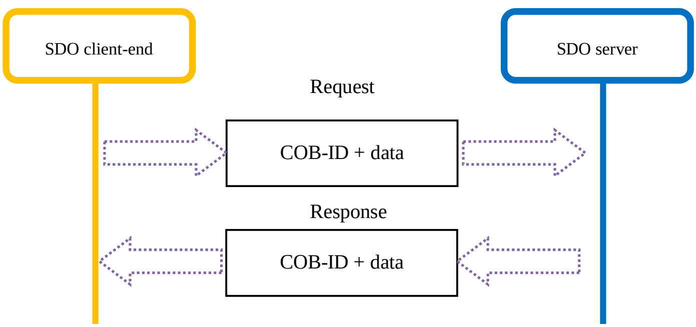
11.2.4 Service Data Object (SDO)

The Service Data Object (SDO) is linked to the object dictionary through object indexes and sub-indexes, through the object contents in the object dictionary can be read or partly modified if allowed via SDO.

11.2.4.1 SDO Transmission Mode

The SDO transmission follows the client-server mode, i.e., that is, the ask-and-answer mode, and SDO is initiated by the SDO client in the CAN bus network and answered by the SDO server; the data exchange between SDOs requires at least two CAN messages, and the CAN identifiers of the two CAN messages are not the same. The transmission mode is shown in the following figure:

Figure 11- 3 SDO client-end read and write the object dictionary in SDO server



11.2.4.2 SDO Transmission Format

SDO transmission is divided into object data transmission of no more than 4 bytes and higher than 4 bytes. The accelerated SDO transmission mode is used when it is not higher than 4 bytes, and the segmented transmission or block transmission mode is used when it is higher than 4 bytes. SD700 series drives only support accelerated SDO transmission mode. The SDO communication message is basically composed of COB-ID + command code + index + subindex + data. The data segments are arranged in the "little-endian ", mode where the lower bits are before the higher bits. SDO transmission message format is shown in Table 11-17.

Table 11- 17 Transmission messages

<table>
<tr><th>COB-ID</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>600h+Node-ID</td><td rowspan="2">command code</td><td colspan="2">index</td><td>subindex</td><td colspan="4">data area</td></tr>
<tr><td>580h+Node-ID</td><td colspan="2">index</td><td>subindex</td><td colspan="4">data area</td></tr>
</table>

For example, if the data area needs to send or receive data 32-bit 0x11223344, it is arranged as 44 33 22 11.

(1) SDO accelerated writing transmission message

For reads and writes not higher than 4 bytes, accelerated SDO transmission is used. The transmission messages vary according to the inconsistency of reading/writing method and data length. The format of the accelerated SDO write message is shown in Table 11-18.

Table 11- 18 Explanation of accelerated SDO message format

<table>
<tr><th></th><th>COB-ID</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td rowspan="3">client
→</td><td rowspan="3">600h+Node-ID</td><td>23H</td><td colspan="2" rowspan="3">index</td><td rowspan="3">subindex</td><td colspan="4">data</td></tr>
<tr><td>2BH</td><td colspan="2">data</td><td>-</td><td>-</td></tr>
<tr><td>2FH</td><td>data</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="2">server
←</td><td rowspan="2">580h+Node-ID</td><td>60H</td><td colspan="2" rowspan="2">index</td><td rowspan="2">subindex</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>80H</td><td colspan="4">stop code</td></tr>
</table>

Note: 1. "-" means data is available but not considered, and it is recommended to write 0 when writing data.

2. The servo drive currently supports the following command words

Table 11- 19 SDO write command word

Command word	Description
2Fh	write 1 byte
2Bh	write 2 byte
23h	write 4 byte

Example 1: If the slave Node-ID is 1 and use SDO to write the object 100Dh(00), which is 8 bits, and write data 64h to this object, the data command is sent as:

Table 11- 20

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	601	2F	0D	10	00	64	-	-	-

If the parameter is written successfully, the returned data frame is:

Table 11- 21

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	581	60	0D	10	00	-	-	-	-

Example 2: If the slave Node-ID is 1, and write the manufacturer parameter Pn500 [2003h(01)] with SDO, which is 16 bits, and the data 64h needs to be written to this object, the data command is sent:

Table 11- 22

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	601	2B	05	20	01	64	00	-	-

If the parameter is written successfully, the returned data frame is:

Table 11- 23

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	581	60	05	20	01	-	-	-	-

(2) SDO accelerated reading transmission messages

The SDO data reading is accelerated when the object message is not higher than 4 bytes. The format of the accelerated SDO reading message is shown in Table 11-24.

Table 11- 24 Accelerated SDO message format description

<table>
<tr><th>Frame</th><th>COB-ID</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>client→</td><td>600h+Node-ID</td><td>40H</td><td colspan="2">index</td><td>subindex</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td rowspan="4">server ←</td><td rowspan="4">580h+Node-ID</td><td>43H</td><td colspan="2" rowspan="4">index</td><td rowspan="4">subindex</td><td colspan="4">data</td></tr>
<tr><td>4BH</td><td colspan="2">data</td><td>-</td><td>-</td></tr>
<tr><td>4FH</td><td>data</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>80H</td><td colspan="4">stop code</td></tr>
</table>

Example 1: slave Node-ID 1, read object 100Dh(00) with SDO, sends the following command:

Table 11- 25

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	601	40	0D	10	-	-	-	-	-

In normal cases, the returned data frame is:

Table 11- 26

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	581	4F	0D	10	00	00	-	-	-

Example 2: slave Node-ID 1, read manufacturer parameter P204 [2002h(05)] with SDO, and send the following command:

Table 11- 27

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	601	40	02	20	05	-	-	-	-

If the drive electronic gear ratio is 16777216:10000, that is, Pn204=16777216, then the data frame returned under normal conditions is:

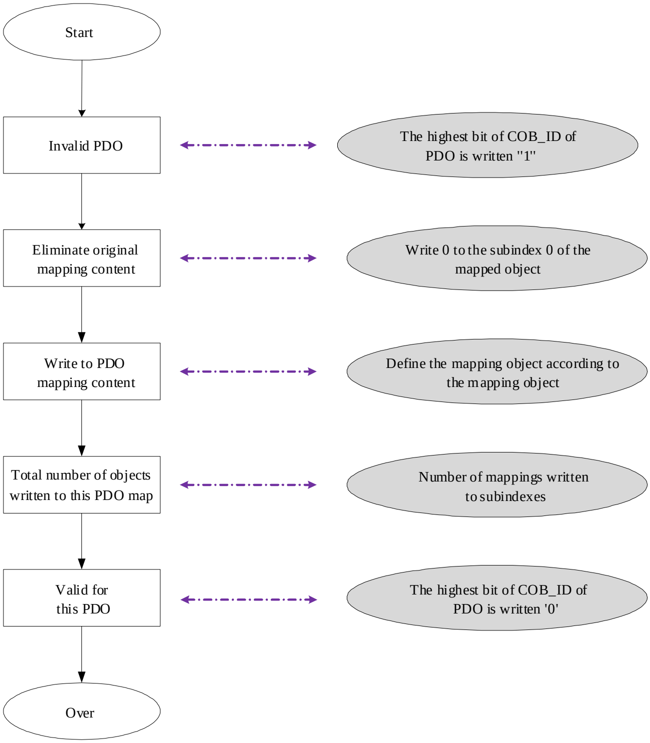
Table 11- 28

Frame format	COB-ID	0	1	2	3	4	5	6	7
Data frame	581	4B	02	20	05	00	00	00	01

11.2.5 Procedure Data Object (PDO)

Procedure Data Object (PDO) are used to transmit real-time data and are the main data transmission mode in Canopen. Since PDO transmission does not require a response, and the PDO must be no longer than 8 bytes in length, the transmission is quite fast. The PDO mapping configuration process is as follows:

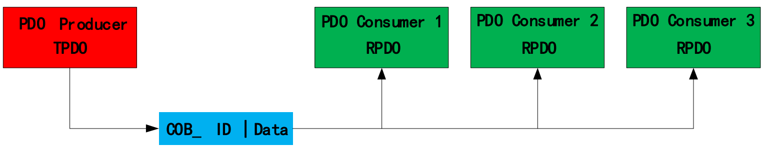
Figure 11- 4 PDO mapping configuration process



(1) PDO transmission mode

PDO uses a production-consumption-end mode, where each network node can listen to messages from the transmitting node and also determines whether a message needs to be processed after it is received. PDO data can be done on a one-to-one or one-to-many basis. Each PDO message contains a transmit PDO (TxPDO) and a receive PDO (RxPDO), and its transmission mode is defined in the PDO communication parameter index. The transmission mode is shown below:

Figure 11- 5 PDO transmission mode



(2) PDO object

PDO can be divided into the receive PDO (RPDO) and transmit PDO (TPDO). PDO is determined by communication parameters and mapping parameters simultaneously to decide the way and content of transmission. This servo drive is designed with 4 RPDOs and 4 TPDO to realize the data transmission of PDO, and the list of related objects is shown in Table 11-29.

Table 11- 29 PDO object list

Designation		COB-ID	Communication object	Mapping objects
RPDO	RPDO1	200h + Node-ID	1400h	1600h
	RPDO2	300h + Node-ID	1401h	1601h
	RPDO3	400h + Node-ID	1402h	1602h
	RPDO4	500h + Node-ID	1403h	1603h
TPDO	TPDO1	180h + Node-ID	1800h	1A00h
	TPDO2	280h + Node-ID	1801h	1A01h
	TPDO3	380h + Node-ID	1802h	1A02h
	TPDO4	480h + Node-ID	1803h	1A03h

(3) PDO Communication Parameter

The COB-ID of the PDO contains control bits and identification data to determine the bus priority of this PDO. COB-ID is located on sub-index 01 of the communication parameters (RPDO: 1400h to 1403h; TPDO: 1800h to 1803h) and the highest bit determines whether this PDO is valid or not.

Table 11- 30

MSB **LSB**

31	30 0
0： on 1： off	1400h~1403h + Node-ID 1800h~1803h + Node-ID

For example, for the node with Node-ID 1, COB-ID is "80000201h" when RPDO is invalid, and writing "00000201h" to this COB-ID will activate RPDO1.

(4) PDO Transmission Type

The transmission type of PDO is located on sub-index 02 of the communication parameters (RPDO: 1400h to 1403h; TPDO: 1800h to 1803h).

Table 11- 31 PDO transmission type

Communication type value	Synchronization		Asynchronization
	Cyclic	Non-cyclic	
0		○	
1~240	○	-	-
241~253	-		
254/255	-	-	○

When the transmission type of TPDO is 0, TPDO is sent if the mapping data is changed and a synchronization frame is received;

When the transmission type of TPDO is 1 to 240, TPDO is sent when the corresponding number of synchronization frames is received;

When the transmission type of TPDO is 254 or 255, TPDO is sent when the mappping data is changed or the event timer arrives;

When the output type of the RPDO is 0 to 240, update the latest data of this RPDO to the application whenever a synchronization frame is received;

When the transmission type of RPDO is 254 or 255, update the received data directly to the application.

(5) Inhibit time

The inhibit time is set for TPDO and stored in subindex 03 of the communication parameter (1800h to 1803h) to prevent the CAN network from being occupied by PDOs with lower priorities. The time unit of this parameter is 125us. After setting the value, the transmission interval of the same TPDO should not be shorter than the corresponding time of this parameter.

For example, if the inhibit time of TPDO1 is 16, the minimum transmission interval of TPDO1 is 2ms.

Table 11- 32

Cautions	
	● The inhibit time should not be too short, otherwise bus overload may be caused when the data keep changing. Please set the inhibit time properly.

(6) Event timer

For TPDO with asynchronous transmission (transmission type 254 or 255), define an event timer on sub-index 05 of the communication parameter (1800h to 1803h). The event timer can also be seen as a trigger time (timer) that triggers the corresponding TPDO when the set time is reached.

(7) PDO mapping parameter

All PDO transmission data must be mapped to the corresponding index area through the object dictionary. During mapping, users need to configure indexes, subindexes, and mapping object lengths in the corresponding format. Each PDO data length cannot exceed 8 bytes for mapping one or more objects simultaneously. Index 0 records the number of objects mapped to the PDO, and subindexes 1 to 4 indicate the mapping content. The mapping parameters are defined as follows:

Table 11- 33 PDO mapping parameter content definition

Bit	31	……	16	15	……	8	7	……	0
Definition	index			subindex			Object length / Bit length 08h / 8 bits 10h / 16 bits 20h / 32 bits		

For example:

Figure 11- 6 RPDO1 mapping object 6040h

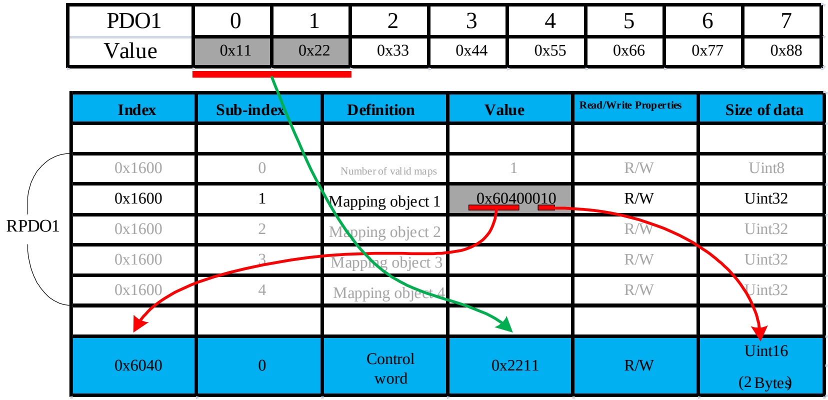
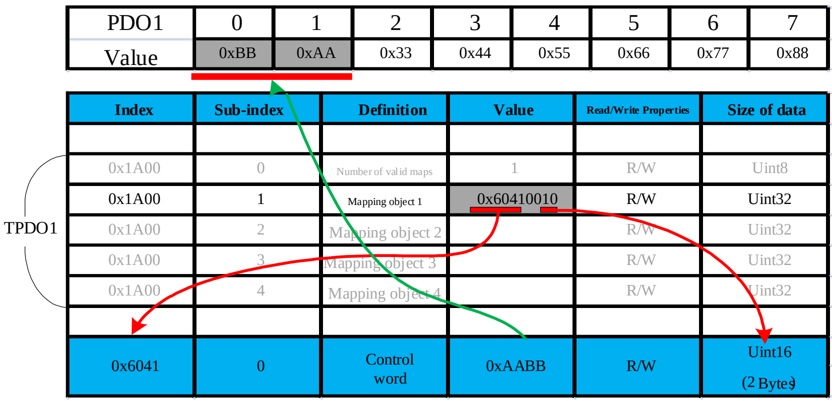


Figure 11- 7 RPDO1 mapping object 6041h



11.2.6 Synchronization (SYNC)

The servo drive can not only synchronize the consumer, but also the producer. The objects can be synchronizized are COB-ID (1005h) and cyclic period (1006h).

The second highest bit of the synchronization object COB-ID (1005h) determines whether the synchronization is activated or not:

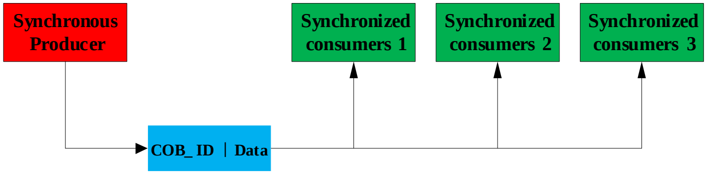
Table 11- 34

MSB **LSB**

31	30	29 0
0	0: off 1: on	0x80

Similar to PDO transmission, the output of synchronization objects follows the producer-consumer mode. In a Canopen network, only one sends the synchronization object (SYNC), and the sender is the producer while the receiver is the consumer, and the transmission framework is shown in Figure 11-8.

Figure 11- 8 Synchronization transmission mode



The synchronization in Canopen is realized by sending control data to each slave with PDO. Each slave that receives control commands from the master only saves the commands temporarily, and only after all the slave commands are sent will the master send out a synchronization (SYNC) broadcast message, and all slaves that support synchronization transmission mode will execute the previously received control commands at the same time after they have received the synchronization (SYNC) message.

PDO synchronization transmission is closely linked to synchronization frames and its specific application is shown below:

Table 11- 35 PDO trigger method

Communication type value	Synchronization		Asynchronization
	Cyclic	Non- cyclic	
0		○	
1~240	○	-	-
241~253	-		
254~255	-	-	○

When the transmission type of TPDO is 0, the TPDO is sent if the mapping data is changed and a synchronization frame is received;

When the transmission type of TPDO is 1 to 240, the TPDO is sent when the corresponding number of synchronization frames is received;

When the transmission type of TPDO is 254 or 255, the TPDO is sent when the mapping data is changed or the event timer arrives;

When the output type of the RPDO is 0 to 240, update the latest data of the RPDO to the application whenever a synchronization frame is received;

When the transmission type of RPDO is 254 or 255, the received data is updated directly to the application.

11.2.7 Emergency (EMCY)

When a Canopen node fails, it sends an emergency message according to the table conversion mechanism. Emergency messages follow the producer-consumer model. After a node fault is sent, other nodes in the CAN network can choose to handle the fault. This servo driver only acts as an emergency message producer and does not process emergency messages from other nodes.

When a node fails, the drive updates the error register (1001h) and predefined error field (1003h) regardless of whether emergency messages are activated.

Users need to activate the emergency messages for use.

Table 11- 36

MSB		LSB
31	**30**	**0**
0：on 1：off	0x80+Node-ID	

The format of the emergency message sent by the servo drive is:

Table 11- 37

COB-ID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0x80+Node-ID	error code		error register	NA	auxiliary byte			

Note: The error register is consistent with 1001h:

(1) The error code shall be consistent with the requirements of DS301, and the auxiliary byte shall be zero in case of abnormal communication.

(2) In case of an exception specified by the user, the error code is 0xFF00, and the auxiliary byte displays that specified code.

For example, enable emergency message on node 1(Pn080=1).

(1) Node pre-running (turning on SDO running is valid)

Table 11- 38

Frame format	**COB-ID**	**0**	**1**
Data frame	00	80	01

Note: Here frames mean remote frames.

(2) Acticate the emergency message object 1014h, in which Bit31 is used to turn on /off the emergency message. Accordingly, the data sent by the upper computer is: (Write data 0x00000081)

Table 11- 39

COB-ID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
601H	23	14	10	00	81	00	00	00

Note: here frames mean data frames.

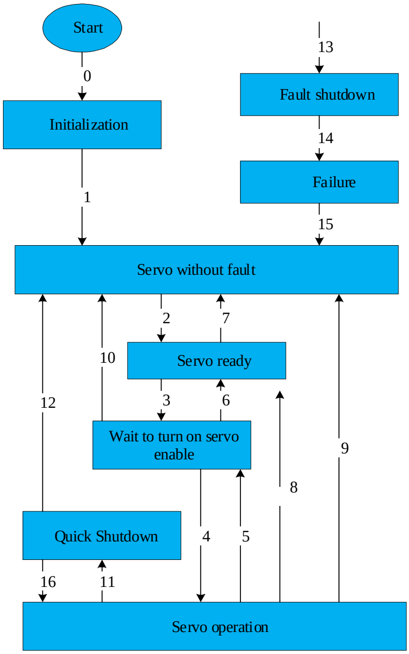
(3) Use the monitoring code Un031(the communication address is 0xE031) to check whether the drive has activated emergency messages.

11.2.8 Servo Stauts

11.2.8.1 Servo Stauts

The SD700 Canopen drive is controlled according to the standard CiA402 protocol. The overall running status diagram is as follows:

Figure 11- 9



States in the figure above are described as follows:

Table 11- 40 CiA States description table

CiA status	Description
Initialize	The drive is initialized and the internal self-inpection t is complete. Drive parameters cannot be set and drive does not run.
Servo no faults	The drive is fault free and the drive parameters can be set.
Switch On	The drive is ready and the drive parameters can be set.
Wait to enable servo	Drive is waiting for servo to be enabled and drive parameters can be set.
Enable Operation	The drive is running normally, a servo mode has been enabled, and the motor is powed. Drive parameters can be modified based on the specific mode.

CiA status	Description
Quick stop	Quick stop is activated and the drive is executing it. Drive parameters can be modified based on the specific mode.
Fault stop	Faults occur and drive is performing this function. Drive parameters can be modified based on the specific mode.
Fault	When the fault stop is completed and all functions of the drive are disabled, users can change the parameters of the corresponding drive to troubleshoot the fault. Example: For a resettable fault, run the control word 6040h=0x80 to reset the fault.

Control commands and status switching are shown in the following table:

Table 11- 41

CiA402 status switching		6040h (control word)	6041h (status word)bit 0~9[1]
0	Power-on → Initialize	Natural transition, no control commands required.	0x0000
1	Initialize → Servo no fault	Natural transition, no control commands required. If an error occurs during the initialization, jump to Step 13	0x0250
2	Servo no fault→Switch on	0x06	0x0231
3	Switch on→Wait to enable servo	0x07	0x0233
4	Wait to enable servo→Enable operation	0x0F	0x0237
5	Enable operation→Wait to enable servo	0x07	0x0233
6	Wait to enable servo→Switch on	0x06	0x0231
7	Switch on→No faults	0x00	0x0250

8	Enable operation→Switch on	0x06	0x0231
9	Enable operation→No faults	0x00	0x0250
10	Wait to enable servo→Servo no fault	0x00	0x0250
11	Enable operation→Quick stop	0x02	0x0217
12	Quick stop→No faults	No need any control commands, natural transition after the quick stop is complete.	0x0250
13	→Fault stop	No control command is required and the system switches to the fault stop state in face of faults	0x021F
CiA402 status switching		**6040h(control word)**	**6041h (status word) bit 0~9[1]**
14	Fault stop→Fault	No need any control command, natural transition and self-switching after the fault stop is completed.	0x0218
15	Fault→Servo no fault	0x80 fault reset	0x0250
16	Quick stop→Enable operation	Send 0x0F when stop is completed	0x0237

Note: [1] Bit10 ~15 of 6041h status word is related to the running state of each servo mode, so it is represented by "0".

11.2.8.2 Status Word 6041h

Table 11- 42

Object 6041h					**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041h**								
Designation	**Status Word**								
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535				
Mapping	Y	**Access**	RO	**Factory setting**	0				

Function description	Bit definition of a status word:

Bit	Designation	Bit definition
0	Switch on	1: valid; 0: invalid.
1	Wait to enable servo	1: valid; 0: invalid.
2	Enable operation	1: valid; 0: invalid.
3	Fault	0：no faluts；1：faults
4	Enable voltage	1: valid; 0: invalid.
5	Quick stop	0：vaild；1：invalid。
6	Power-on and running allowed	1: valid; 0: invalid.
7	Warning	1: valid; 0: invalid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	Target reached	Speed mode: 0: target speed is not reached; 1: target speed reached. Position mode: 0: target position is not reached; 1: target position is reached.
11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
12~13	Relate to control mode	
14	NA	
15	Home return completed	0: home return is not performed or not completed. 1: home return is completed and the reference point has been found.

11.2.8.3 Stop Mode

SD700 CANopen supports the following stop methods:

(1) Disable the servo to stop

When servo is disabled OFF, servo stops running.

(2) Servo fault stop

When servo fault or warning occurs, servo automatically enters stop state.

(3) Quick stop

In the non-fault state, if the control word 6040h: bit2=1, quick stop function is performed and stop method is selected via 605Ah.

Table 11- 43

Object 605Ah				PP	PV	PT	HM	IP
Index	605Ah							
Designation	Quick stop method selection							
Object structure	VAR	Data type	Int16	Data range	0~2			
Mapping	NO	Access	RW	Factory setting	2			

Function description:

Displayed value	Control mode display
0	Free stop,and free runnining after free stop is completed
1	Ramp stop at deceleration speed set at 6084h(hm：609Ah), and free running after stop is completed;
2	Ramp stop at deceleration speed set at 6085h, and free running after stop is completed;

(4) Halt stop (Not supported yet)

When the control word 6040h:bit8=1 in the non-fault state, stop will be halted, and the stop mode is selected via 605D.

Table 11- 44

Object 605Dh				PP	PV	PT	HM	IP
Index	605Dh							
Designation	Halt method selection							
Object structure	VAR	Data type	Int16	Data range	1~3			
Mapping	NO	Access	RW	Factory setting	1			

Function description:

Displayed value	Control mode display
1	Ramp stop as setting at 6084h/6087h(hm：609Ah), and position is locked after stop is completed;
2	Ramp stop as setting at 6085h/6087h, and position is locked after stop is completed;
3	Emergency torque stop, and position is locked after stop is completed;

11.2.8.4 Servo Running Mode

SD700 CANopn supports 5 servo running modes.

Servo operation modes can be set by object dictionary 6060h. The current running mode of the servo can be viewed through object dictionary 6061h.

(1)Mode selection at 6060h

Table 11- 45

Object 6060h		PP	PV	PT	HM	IP
Index	6060h					
Designation	Running modes selection					

Object structure	VAR	**Data type**	Int8	**Data range**	0~7
Mapping	Y	**Access**	RW	**Factory setting**	1
Function description	Set servo running mode:				

Displayed value	**Control mode display**
0	NA
1	Profile position mode (PP)
3	Profile velocity mode (PV)
4	Profile torque mode (PT)
6	Homing method (HM)
7	Interpolation mode (IP)

(2)Mode display 6061h

Table 11- 46

Object 6061h				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6061h**							
Designation	Running mode display							
Object structure	VAR	**Data type**	Int8	**Data range**	0~7			
Mapping	Y	**Access**	RO	**Factory setting**	0			
Function description								

Displayed value	**Control mode display**
0	NA
1	Profile position mode (PP)
3	Profile velocity mode (PV)
4	Profile torque mode (PT)
6	Homing method (HM)
7	Interpolation mode (IP)

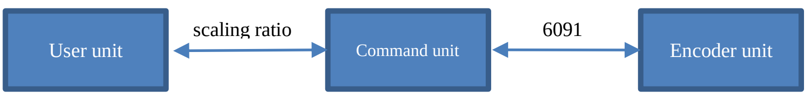
11.2.8.5 Conversion Factor Setting

■ Encoder unit: drive drives the motor directly, and position feedback of the motor is pulse quantity, and the encoder unit is the pulse unit.

■ Command unit: control commands of the drive and the commands from CanOpen are usually used as command units. The command units and encoder units are converted by the servo-end gear ratio 6091h.

■ User unit: users usually use the actual load displacement, velocity, and acceleration units for convenience. User units and command units are converted by the user scaling ratio.

Figure 11- 10



When the encoder unit, command unit and user unit do not match, it will cause the motor to run incorrectly. Therefore, before running the servo drive, the conversion factor must be set correctly, and the ratio between encoder units and user units must be decided by the conversion factor.

■ When using a 23-bit motor in profile position mode and gear ratio is set to 1:1, if the motor requires 10 turns of motion, motor speed is 600 rpm, user acceleration is 600 rpm/s, and user deceleration is 300 rpm/s.

Target position: 607Ah =10*8388608 ；

Profile velocity: 6081 = 600*8388608/60；

Profile acceleration: 6083 = 600*8388608/60；

Profile deceleration: 6084 = 300*8388608/60 ；

And so:

$$\text{Acceleration time} = \frac{6081}{6083} = 1s; \qquad \text{Acceleration time} = \frac{6081}{6084} = 2s;。$$

(1) Gear ratio: 6091(Pn204：Pn206)

Gear ratio is essentially the motor displacement (in encoder units) corresponding to a load shaft displacement of 1 command unit.

The gear ratio consists of 6091-01 (Pn204 electronic gear ratio numerator) and 6091-02 (Pn206 electronic gear ratio denominator), through which the proportional relationship between load shaft displacement (command unit) and motor displacement (encoder unit) can be established as follows:

The motor is connected to the load by means of a gearbox and other mechanical transmission mechanisms. Therefore, the gear ratio is related to the mechanical reduction ratio, mechanical dimension-related parameters, and motor resolution. The calculation is as follows:

$$\text{gear ratio} = \frac{\text{motor resolution}}{\text{load resolution}}$$

Table 11- 47

Object 6091h					**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6091h**								
Designation	Gear Ratio								
Object structure	ARR	**Data type**	Uint32	**Data range**	Uint32				
Mapping	Y	**Access**	RW	**Factory setting**	-				
Function description	Position factor is used to establish a user-specified proportional relationship between load displacement and motor displacement: Motor displacement (motor unit) = load displacement (user unit) × gear ratio (6091) Motor speed (rpm) VS load shaft speed (p/s): $\text{motor speed(rpm)} = \frac{\text{load shaft speed} * \text{gear ratio}}{\text{encoder resolution}} * 60$								

11 Communication

	Motor acceleration (rpm/ms) VS load shaft acceleration (command unit $/s^2$): $$\text{Motor acceleration (rpm/ms)} = \frac{\text{load shaft speed} * \text{gear ratio}}{\text{encoder resolution}} * 1000/60$$

Table 11- 48

Sub-index	**00h**				
Designation	**Number of sub-indexes**				
Object structure	VAR	**Data type**	Uint8	**Data range**	2
Mapping	Y	**Access**	RO	**Factory setting**	2

Table 11- 49

Sub-index	**01h**				
Designation	**Motor revolutions**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	Y	**Access**	RW	**Factory setting**	1

Table 11- 50

Sub-index	**02h**				
Designation	**Shaft revolution**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	Y	**Access**	RW	**Factory setting**	1

(2) Scaling ratio (User ratio)

The scaling ratio is set by the user of the upper computer, through which the proportional relationship between the load shaft displacement (user unit) and the motor displacement (command unit) can be established:

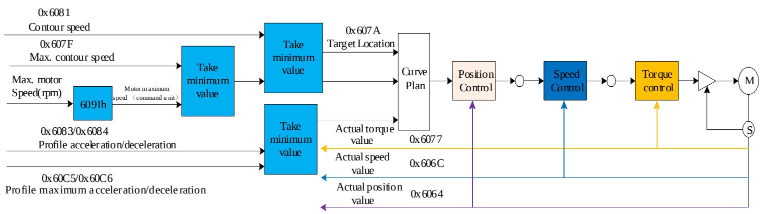
Motor displacement (command unit) = load shaft displacement (user unit) ×scaling ratio

11.2.9 Control Mode

11.2.9.1 Profile Position Mode (PP)

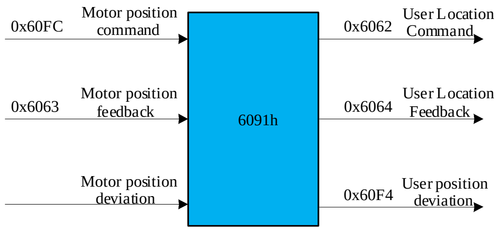
When in profile position mode, the master sends a dictionary of relevant objects such as the required target position (absolute or relative), velocity, acceleration and deceleration of the position profile to the servo drive, which generates the target profile command based on the relevant data and commands received.

Figure 11- 11 Profile position mode control diagram



The conversion of user and encoder unit in profile position mode via 0x6091 is illustrated as follows:

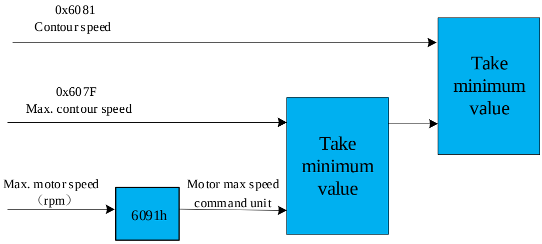
Figure 11- 12



0x6091(gear ratio)= $\frac{\text{motor resolution}}{\text{load resolution}}$. 0x6063(motor position feedback) and 0x6064(user location feedback) relationship is : 0x6063(encoder unit)=0x6064(command unit)×gear ratio.

Relationship of 0x6081 (profile speed), 0x607F (user maximum speed) and motor maximum speed after conversion is as follows:

Figure 11- 13

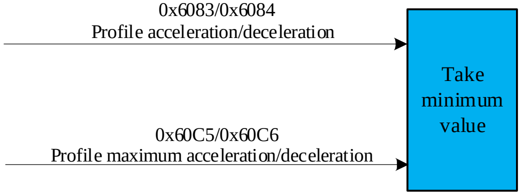


Motor speed (rpm) versus load shaft speed (command unit/s):

$$\text{motor speed (rpm)} = \frac{\text{load shaft speed} \times 6091\text{h}}{\text{encoder resolution}} \times 60$$

Relationship between 0x6083/0x6084 (profile acceleration/deceleration) and 0x60C5/0x60C6 (profile maximum acceleration/deceleration) is as follows:

Figure 11- 14



Example: If a 23-bit motor is used and the gear ratio is set to 1:1, if the motor is required to move 10 turns, then the motor speed is 600 rpm, the user acceleration is 600 rpm/s, and the user deceleration is 300 rpm/s.

Target position：607Ah =10*8388608 ；

Profile speed:6081 = 600*8388608/60；

Profile acceleration:6083 = 600*8388608/60；

Profile deceleration:6084 = 300*8388608/60 。

Related object dictionaries:

Table 11- 51

Control word 6040h		
Bit	**Designation**	**Description**
0	Servo ready (Switch on)	0: invalid; 1: valid.
1	Power on the main circuit (Enable voltage)	0: invalid; 1: valid.
2	Quick stop	0: valid; 1: invalid.
3	Servo running (Enable operation)	0: invalid; 1: valid.
4	New target posotopm (New set-point)	Rising edge triggers a new target position.
5	Change setting immediately	0: non-immediate change; 1: immediate change.
6	Abs/Rel	0: target position is an absolute position command; 1: target position is a relative position command.

Table 11- 52

Status word 6041h		
Bit	**Designation**	**Description**
10	Target reached	0: target position not reached; 1: target position reached;
12	Change target position(Set point acknowledge)	0: target position changeable; 1: target position unchangeable；
13	Following error	0: no excessive position deviation fault; 1: excessive position deviation fault;

15	Home return (Home find)	0: home return not completed; 1: home return completed;

Table 11- 53

Index	Sub-index	Designation	Read/ Write	Data type	Unit	Range
0x603F	00	Error code	RO	UINT16	-	0~65535
0x6040	00	Control word	RW	UINT16	-	0~65535
0x6041	00	Status word	RO	UINT16	-	0~65535
0x6060	00	Running Mode	RW	INT8	-	0~7
0x6061	00	Mode display	RO	INT8	-	0~7
0x6062	00	Position command	RO	INT32	Command unit	-231 ~(231-1)
0x6063	00	Motor position feedback	RO	INT32	Encoder unit	-2^{31} ~(2^{31}-1)
0x6064	00	User position feedback	RO	INT32	Command unit	-2^{31} ~(2^{31}-1)
0x606C	00	Real speed feedback	RO	INT32	Command unit/s	-2^{31} ~(2^{31}-1)
0x607A	00	Target position	RW	INT32	Command unit	-2^{31} ~(2^{31}-1)
0x6081	00	Profile speed	RW	UINT32	Command unit/s	0~(2^{32}-1)
0x6083	00	Acceleration	RW	UINT32	Command unit/s^2	0~(2^{32}-1)
0x6084	00	Deceleration	RW	UINT32	Command unit/s^2	0~(2^{32}-1)

The following table shows the steps for setting up the profile position running mode:

Table 11- 54

Item	Step	Parameter input	Status word display (6041h)
Profile position parameter assignment	0	607Ah = 10000	0x0250
	1	6081h = 1000	0x0250
	2	6083h = 200	0x0250
	3	6084h = 200	0x0250

Control mode switching	4	6060h = 0x01	0x0250
Servo enabling	5	6040h = 0x06	0x0231
	6	6040h = 0x07	0x0233
	7	6040h = 0x0F	0x0637
Absolute/relative position selection	8	6040h Bit6 set 1 (relative position)	0x0637
Position command triggering	9	6040hBit4 set 1 (rising edge)	0x1237
Positioning completed	10	6041h Bit10 set 1	0x0637
Bit reset triggering for next use	11	6040hBit4 reset	0x0637

Description of control word 6040h and status word 6041h in profile position mode:

Table 11- 55

Object 6040h				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6040h**							
Designation	**Control Word**							
Object structure	VAR	**Data type**	Uint 16	**Data range**	0~65535			
Mapping	Y	**Access**	RW	**Factory setting**	0			
Function description	Bit definition of the control word:							

Bit	**Designation**	**Description**
0	Servo ready	0: invalid; 1: valid.
1	Turn on the main circuit electricity	0: invalid; 1: valid.
2	Quick stop	1: invalid; 0: valid.
3	Enable operation	0: invalid; 1: valid.
4	Enable the new position command	0→1: when there is a new segment of position instruction to be changed, whether it is valid or not is determined by the servo status; 1→0: change 6041h: bit12 from 1to 0, whether success is determined by servo status.
5	Position command (change mode)	0: non-immediate change; 1: immediate update.
6	Position command (type)	0: 607Ah indicates an absolute position command; 1: 607Ah indicates a relative position command.
7	Fault reset	bit7 rising edge is valid; bit7 is held to 1. All other control commands are invalid.
8	Halt	0: invalid; 1: valid.
9~10	NA	

	11~15	Factory-defined	
	Note: each bit in the control word needs to be used together with other bits to form a control command.		

Table 11- 56

Object 6041h						**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041h**									
Designation	**Status Word**									
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535					
Mapping	Y	**Access**	RO	**Factory setting**	0					

Function description

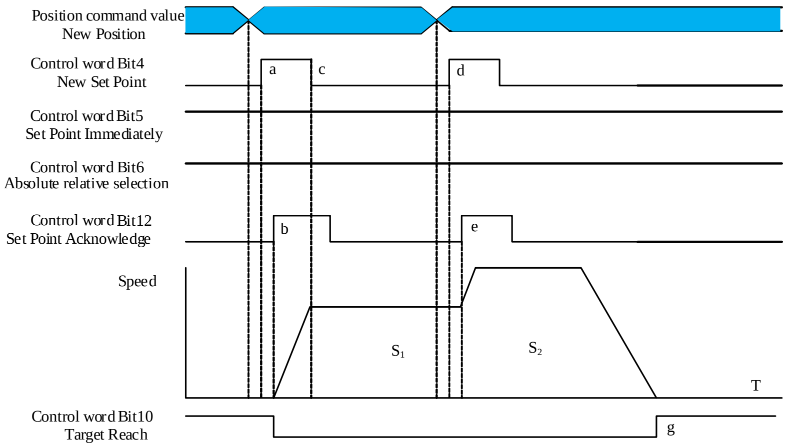
Bit definition of status word:

Bit	Designation	Description
0	Servo ready	0: invalid; 1: valid.
1	Waiting to turn on servo enable	0: invalid; 1: valid.
2	Servo operation	0: invalid; 1: valid.
3	Faults	0:no faults;1: faults.
4	Turn on the main circuit power	0: invalid; 1: valid.
5	Quick stop	0: valid; 1: invalid.
6	Power-on and running allowed	0: invalid; 1: valid.
7	Warning	0: invalid; 1: valid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	Target reached	0: target position not reached; 1: target position reached.
11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
12	Position command change signal	0: new positions allowed; 1: new position not allowed;
13	Position deviation status	0: position deviation within 6065h range; 1: position deviation outside 6065h range.
14	NA	
15	Home return completed	0: home return not performed or not completed; 1: home return completed and reference point found.

When running in the profile position mode, there are two ways to change the commands, namely, immediate change and non-immediate change. The specific process of implementing these two ways is explained below.

(1) Absolute position command or relative position command, immediate change

Figure 11- 15 Timing sequence of relative position command value in immediate change mode



0x6040： bit5=1 immediate change mode ,run the 1st stage S1 position command, and before its completion, 0x6040: bit4 and then bit12 are changed into 0 from 1, this means new position command S2 needs to be changed. When 0x6040:bit4 and then bit 12 are changed into 1 from 0, that means new position command is changed and will be performed immediately.

0x6040: bit6=1 relative position command, when the 2nd segment position command is completed, the total displacement command = 1st segment 0x607A target position + 2nd segment 0x607A target position.

0x6040: bit6=0 absolute position command, when the 2nd segment position command is completed, the total displacement command = 2nd segment 0x607A target position.

Running steps corresponding to the order shown in Figure 11.15 are shown in the following table

Table 11- 57

Step	Control word (6040h)	Status word (6041h)	Description (relative position mode)
1	0x0006	0x1231	No new commands can be received, servo ready.
2	0x0007	0x1233	No new commands can be received, the servo is ready and waiting to enable the servo.
3	0x006F	0x0637	New command can be received, servo enabled. (Note 1: 6040h: bit5=1 means the position command is changed immediately, bit6=1 means it is a relative position command) (Note 2: At this time 6041h: bit10=1 since the initial target position is 0, target position is

			reached by default)
4	0x007F	0x1237	The servo has received the profile target position (607Ah), the profile target running speed (6081h), the profile acceleration (6083h), and the profile deceleration (6084h), and runs them immediately.
5	If there are no new position commands to be changed immediately, proceed to step 6 and wait for the end. If there is a new position command that needs to be changed immediately, proceed to step 7.		
6	0x007F	0x1637	6041h:bit10=1 target position is reached and the running is over.
Step	**Control word (6040h)**	**Status word (6041h)**	**Description (relative position mode)**
7	0x006F	0x0237	6040h: bit10 changed into 0 from 1, and then 6041h: bit12 changed into 0 from 1, a new position command can be received.
8	0x007F	0x1237	The servo has received a new position command, and immediately changes and runs the relevant position command, cycling from step 5. (Note: If there are only two segment commands: relative position target = 1st segment relative position + 2nd segment relative position)

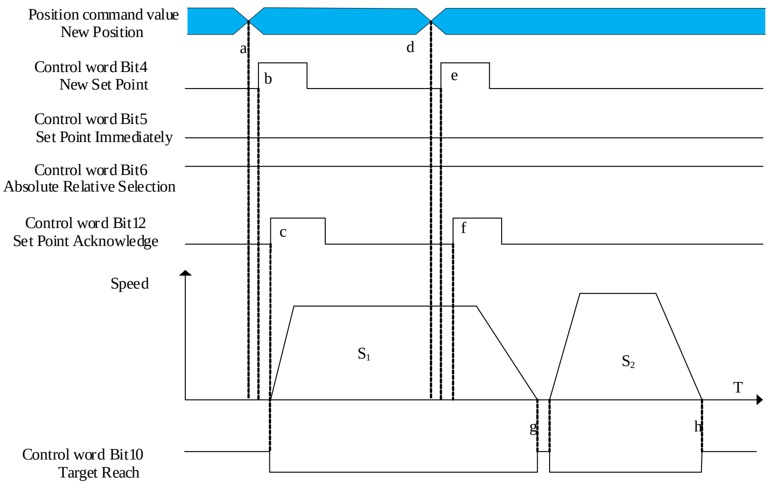
Table 11- 58

Step	**Control word (6040h)**	**Status word (6041h)**	**Description (absolute position mode)**
1	0x0006	0x1231	No new commands can be received, servo ready.
2	0x0007	0x1233	No new commands can be received, the servo is ready and waiting to enable the servo.
3	0x002F	0x0637	New command can be received, servo enabled. (Note 1: 6040h: bit5=1 means the position command is changed immediately, bit6=0 means it is an absolute position command) (Note 2: at this time 6041h: bit10=1 since the initial target position is 0, target is reached by default)
4	0x003F	0x1237	The servo has received the relevant commands for profile target position (607Ah), profile target running speed (6081h), profile acceleration (6083h), and profile deceleration (6084h), and runs them immediately.
5	If there are no new position commands to be changed immediately, proceed to step 6 and wait for the end.		

	If there is a new position command that needs to be changed immediately, proceed to step 7.		
6	0x003F	0x1637	6041h:bit10=1 target position is reached and the running is over.
7	0x002F	0x0237	6040h: bit10 changed into 0 from 1, and then 6041h: bit12 changed into 0 from 1, a new position command can be received.
8	0x003F	0x1237	The servo has received a new position command, and immediately changes and runs the relevant position command, cycling from step 5. (Note: If there are only two segment commands: the absolute position target =2nd segment absolute position)

(2) Absolute position command or relative position command, non-immediate change

Figure 11- 16 Timing sequence for non-immediate change mode of relative position command value



When 0x6040: bit5=1 immediate change mode, run the 1st segment S1 position command, 0x6040: bit4 and then bit12 are changed into 0 from 1 before the command is not finished, it means that there is a new position command S2 needs to be changed. When 0x6040: bit4 and then bit12 are changed into 1 from 0, it means that the new position command has been changed, but it is necessary to wait until the 1st segment position command is finished before running the 2nd segment position command.

Corresponding running steps to Figure11.16 are shown in the following table.

Table 11- 59

Step	Control word (6040h)	Status word (6041h)	Description (relative position mode)
1	0x0006	0x1231	No new commands can be received, servo ready.

2	0x0007	0x1233	No new commands can be received, the servo is ready and waiting to enable the servo.
3	0x004F	0x0637	New command can be received, servo enabled. (Note 1: 6040h: bit5=0 means the position command is not changed immediately, bit6=1 means it is a relative position command) (Note 2: At this time 6041h: bit10=1 since the initial target position is 0, target position is reached by default)
4	0x005F	0x1237	The servo has received the relevant commands for profile target position (607Ah), profile target running speed (6081h), profile acceleration (6083h), and profile deceleration (6084h), and runs them immediately.
5	If there are no new position commands to be changed, proceed to step 6 and wait for the end. If there is a new position command, proceed to step 7.		
6	0x005F	0x1637	6041h:bit10=1 target position is reached and the running is over.
7	0x004F	0x0237	6040h :bit10 changed into 0 from 1, and then 6041h: bit12 changed into 0 from 1, a new position command can be received.
8	0x005F	0x1237	The servo has received a new position command, and runs the next command after the 1st segment command is over, cycling from step 5. (Note: If there are only two segment commands: relative position target = 1st segment relative position + 2nd segment relative position)

Table 11- 60

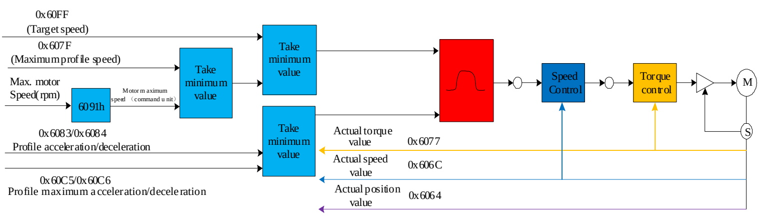
Step	Control word (6040h)	Status word (6041h)	Description (absolute position mode)
1	0x0006	0x1231	No new commands can be received, servo ready.
2	0x0007	0x1233	No new commands can be received, the servo is ready and waiting to enable the servo.
3	0x000F	0x0637	New command can be received, servo enabled. (Note 1: 6040h: bit5=0 means the position command is not changed immediately, bit6=0 means it is an absolute position command) (Note 2: At this time 6041h: bit10=1 since the initial target position is 0, target position is reached by default)
4	0x001F	0x1237	The servo has received the relevant commands for profile target position (607Ah), profile

			target running speed (6081h), profile acceleration (6083h), and profile deceleration (6084h), and runs them immediately.
5	If there are no new position commands to be changed, proceed to step 6 and wait for the end. If there is a new position command, proceed to step 7.		
6	0x001F	0x1637	6041h:bit10=1 target position is reached and the running is over.
7	0x000F	0x0237	6040h :bit10 changed into 0 from 1, and then 6041h: bit12 changed into 0 from 1, a new position command can be received.
8	0x001F	0x1237	The servo has received a new position command, and runs the next command after the 1st segment command is over, cycling from step 5. (Note: If there are only two segment commands: absolute position target = 1st segment absolute position + 2nd segment absolute position)

11.2.9.2 Profile Velocity Mode (PV)

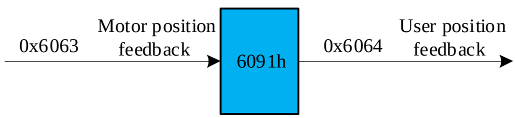
In profile velocity mode, the master transmits the required target velocity, acceleration time and deceleration time to the servo drive, which performs the speed and torque adjustment.

Figure 11- 17 Profile velocity mode control diagram



The conversion of user units and encoder units in profile velocity mode via 0x6091 is illustrated below:

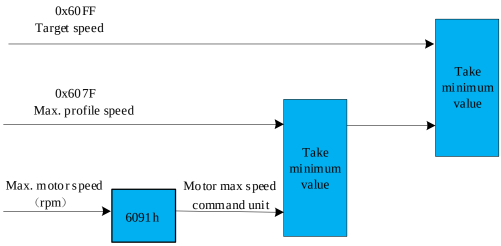
Figure 11- 18



0x6091(gear ratio)= $\frac{\text{motor resolution}}{\text{load resolution}}$. Relationship between 0x6063 (motor position feedback) and 0x6064 (user position feedback): 0x6063 (encoder unit) = 0x6064 (command unit) × gear ratio.

Relationshop of 0x60FF (target speed), 0x607F (user maximum speed) and the corresponding motor maximum speed after conversion as follows:

Figure 11- 19



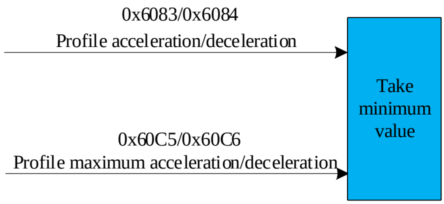
Motor speed (rpm) versus load shaft speed (command unit/s):

$$\text{motor speed(rpm)} = \frac{\text{load shaft speed} \times 6091\text{h}}{\text{encoder resolution}} \times 60$$

Example: gear ratio = 1:1, with 23-bit encoder. Motor speed = 500rpm (corresponding to 0x60FF (load shaft speed)) = 500*8388608/60 = 69905066 (command unit/s).

Relationship between 0x6083/0x6084 (profile acceleration/deceleration) and 0x60C5/0x60C6 (profile maximum acceleration/deceleration) is as follows:

Figure 11- 20



Example: gear ratio = 1:1, with 23-bit encoder. Motor speed = 600rpm, motor acceleration/deceleration = 1200rpm/s.

Target speed: 0x60FF = 600*8388608/60;

Profile acceleration: 0x6083 = 1200*8388608/60;

Profile deceleration: 0x6084 = 1200*8388608/60;

Table 11- 61 Related object dictionaries:

Index	Sub-index	Designation	Read/ Write	Data type	Unit	Range
0x603F	00	Error code	RO	UINT16	-	0~65535
0x6040	00	Control word	RW	UINT16	-	0~65535
0x6041	00	Status word	RO	UINT16	-	0~65535
Index	Sub-	Designation	Read/	Data	Unit	Range

	index		Write	type		
0x6060	00	Running mode	RW	INT8	-	0~7
0x6061	00	Mode display	RO	INT8	-	0~7
0x606C	00	Real speed feedback	RO	INT32	Command unit/s	-
0x607F	00	Maximum profile speed	RW	UINT32	Command unit/s	0~(2^{32}-1)
0x6083	00	Acceleration	RW	UINT32	Command unit /s^2	0~(2^{32}-1)
0x6084	00	Deceleration	RW	UINT32	Command unit /s^2	0~(2^{32}-1)
0x60FF	00	Target speed	RW	INT32	Command unit /s	-2^{31}~(2^{31}-1)

Note: The speed limit value is determined by the smaller of 0x607F and the maximum motor speed.

The operating procedure for the profile velocity mode is shown in the following table:

Table 11- 62

Item	Step	Parameter input	Status word display (6041h)
Profile velocity parameter assignment	1	6083h = 200	0x1250
	2	6084h = 200	0x1250
	3	60FFh = 10000	0x1250
Control mode selection	4	6060h = 0x03	0x1250
Servo enabling	5	6040h = 0x06	0x1231
	6	6040h = 0x07	0x1233
	7	6040h = 0x0F	0x0637

Description of control word 6040h and status word 6041h in the profile velocity mode:

Table 11- 63

Object 6040h					**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6040_h**								
Designation	**Control Word**								
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535				
Mapping	Y	**Access**	RW	**Factory setting**	0				

Function description

Bit definition of the control word:

Bit	Designation	Description
0	Servo ready	0:invalid; 1:valid.
1	Turn on the main circuit electricity	0:invalid; 1:valid.
2	Quick stop	1:invalid；0:valid。
3	Enable operation	0:invalid; 1:valid.
4~6	NA	
7	Fault reset	bit7 rising edge valid; bit7 is held to 1, and all other control commands are invalid.
8	Halt	0:invalid; 1:valid.
9~10	NA	
11~15	Factory-defined	

Note: each bit in the control word needs to be used together with other bits to form a control command.

Table 11- 64

Object 6041h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041h**					
Designation	**Status Word**					
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535	
Mapping	Y	**Access**	RO	**Factory setting**	0	

Function description

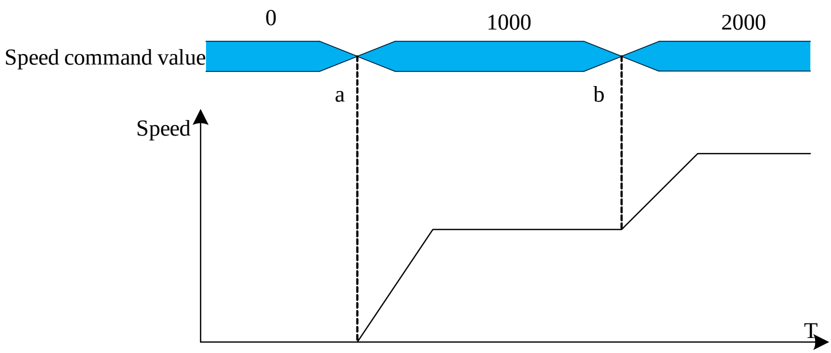
Bit definition of status word:

Bit	Designation	Description
0	Servo ready	0:invalid; 1:valid.
1	Wait to enable servo	0:invalid; 1:valid.
2	Enable operation	0:invalid; 1:valid.
3	Faults	0: no faults; 1: faults。
4	Enable voltage	0:invalid; 1:valid.
5	Quick stop	0:valid；1：invalid
6	Power-on and running allowed	0:invalid; 1:valid.
7	Warning	0:invalid; 1:valid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	Target reached	0: target position not reached; 1: target position reached.

	11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
	12	Zero-speed signal	0: user speed is non- zero; 1: user speed is zero.
	13~14	NA	
	15	Home return completed	0: home return not performed or not completed; 1: home return completed and reference point found.

In profile velocity mode, the velocity command is changed immediately, and its timing sequence diagram is shown in Figure 11.21.

Figure 11- 21 Profile speed mode running time sequence diagram



The timing diagram shown in Figure 11.21 corresponds to the operation steps shown below:

Table 11- 65

Step	Item	Operation
1	Speed command giving	After the speed command is given, the servo-controlled motor runs at the set speed
2	Speed command change	After the speed command changes, the servo-controlled motor changes speed to the set speed.

11.2.9.3 Profile Torque Mode (PT)

In profile torque mode, the master sends the target torque command 6071h and torque ramp constant 6087h to the servo drive, and the torque regulator is performed internally by the servo drive. When the speed reaches the maximum speed limit, it will enter the speed regulation phase.

Figure 11- 22 Timing sequence for profile torque mode operation

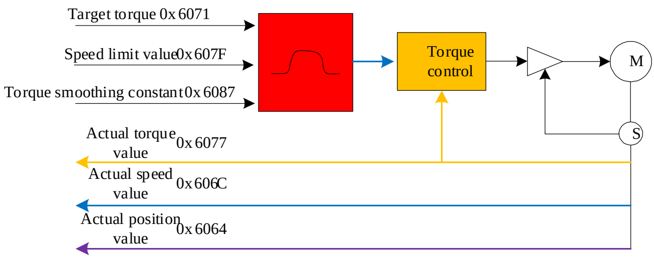


Table 11- 66 Related object dictionaries

Index	Sub-index	Designation	Read/ Write	Data type	Unit	Range
0x603F	0x00	Error code	RO	UINT16	-	0~65535
0x6040	0x00	Control word	RW	UINT16	-	0~65535
0x6041	0x00	Status word	RO	UINT16	-	0~65535
0x6060	0x00	Running Mode	RW	INT8	-	0~7
0x6061	0x00	Mode display	RO	INT8	-	0~7
0x606C	0x00	Real speed feedback	RO	INT32	Command unit /s	-
0x6071	0x00	Target torque	RW	INT16	0.1%	-3000~3000
0x6072	0x00	Maximum torque	RW	UINT16	0.1%	0~3000
0x6074	0x00	Torque command	RO	INT16	0.1%	-
0x6077	0x00	Real torque	RO	UINT16	0.1%	-
0x6087	0x00	Torque ramp time	RW	UINT32	ms	0~(2^{32}-1)

The steps for the profile torque mode are shown in the following table:

Table 11- 67

Item	Step	Parameter input	Status word display (6041h)
Profile torque parameter assignment	1	6071h = 50	0x0250
	2	6087h = 50	0x0250
Control mode switching	3	6060h = 0x04	0x0250
Servo enabling	4	6040h = 0x06	0x0231
	5	6040h = 0x07	0x0233
	6	6040h = 0x0F	0x0637

Description of control word 6040h and status word 6041h in the profile torque mode:

Table 11- 68

<table>
<tr><th>Object 6040h</th><th colspan="3"></th><th>PP</th><th>PV</th><th>PT</th><th>HM</th><th>IP</th></tr>
<tr><td>Index</td><td>6040h</td><td colspan="7"></td></tr>
<tr><td>Designation</td><td colspan="8">Control Word</td></tr>
<tr><td>Object structure</td><td>VAR</td><td>Data type</td><td>Uint16</td><td colspan="3">Data range</td><td colspan="2">0~65535</td></tr>
<tr><td>Mapping</td><td>Y</td><td>Access</td><td>RW</td><td colspan="3">Factory setting</td><td colspan="2">0</td></tr>
<tr><td>Function description</td><td colspan="8">Bit definition of the control word:
<table><tr><th>Bit</th><th>Designation</th><th>Description</th></tr><tr><td>0</td><td>Servo ready</td><td>0: invalid; 1: valid.</td></tr><tr><td>1</td><td>Turn on the main circuit electricity</td><td>0: invalid; 1: valid.</td></tr><tr><td>2</td><td>Quick stop</td><td>1: invalid; 0: valid.</td></tr><tr><td>3</td><td>Enable operation</td><td>0: invalid; 1: valid.</td></tr><tr><td>4~6</td><td>NA</td><td></td></tr><tr><td>7</td><td>Fault reset</td><td>bit7 rising edge is valid;
bit7 is held to 1. All other control commands are invalid.</td></tr><tr><td>8</td><td>Halt</td><td>0: invalid; 1: valid.</td></tr><tr><td>9~10</td><td>NA</td><td></td></tr><tr><td>11~15</td><td>Factory-defined</td><td></td></tr></table>Note: each bit in the control word needs to be used together with other bits to form a control command.</td></tr>
</table>

Table 11- 69

<table>
<tr><th>Object 6041h</th><th colspan="3"></th><th>PP</th><th>PV</th><th>PT</th><th>HM</th><th>IP</th></tr>
<tr><td>Index</td><td>6041h</td><td colspan="7"></td></tr>
<tr><td>Designation</td><td colspan="8">Status Word</td></tr>
<tr><td>Object structure</td><td>VAR</td><td>Data type</td><td>Uint16</td><td colspan="3">Data range</td><td colspan="2">0~65535</td></tr>
<tr><td>Mapping</td><td>Y</td><td>Access</td><td>RO</td><td colspan="3">Factory setting</td><td colspan="2">0</td></tr>
</table>

Function description

Bit definition of status word:

Bit	Designation	Description
0	Servo ready	0: invalid; 1: valid.
1	Wait to enable servo	0: invalid; 1: valid.
2	Servo operation	0: invalid; 1: valid.
3	Faults	0:no faults;1: faults.
4	Enable voltage	0: invalid; 1: valid.
5	Quick stop	0: valid； 1: invalid.
6	Power-on and running allowed	0: invalid; 1: valid.
7	Warning	0: invalid; 1: valid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	NA	
11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
12~14	NA	
15	Home return completed	0: home return not performed or not completed; 1: home return completed and reference point found.

The running steps of the profile torque mode are shown in the following table:

Table 11- 70

Item	Step	Parameter input	Status word display (6041h)
Profile torque parameter assignment	0	6087h = 100	0x0250
	1	6071h = 500	0x0250
Control mode switching	2	6060h = 0x04	0x0250
Servo enabling	3	6040h = 0x06	0x0231
	4	6040h = 0x07	0x0233
	5	6040h = 0x0F	0x0637

11.2.9.4 Homing method (HM)

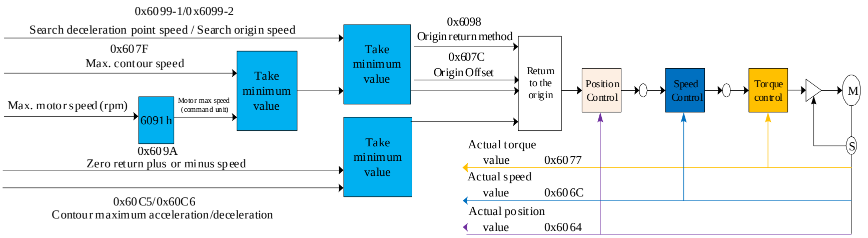
The homing method is used to find the mechanical home point and the position relationship between the mechanical home point and mechanical zero point.

Mechanical home: a fixed position on the machinery, corresponding to a certain determined home position signal switch.

Mechanical zero: mechanical zero point = mechanical home + 0x607C (home offset), if 0x607C = 0, the mechanical zero point is equal to the mechanical home point.

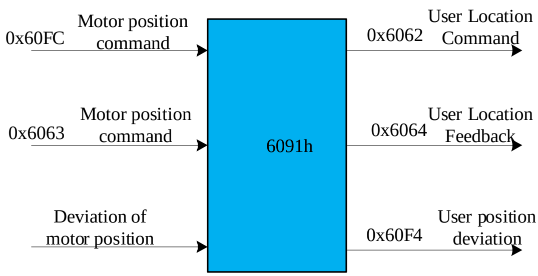
The servo drive will stop at the mechanical zero point after the home return return is completed, and adjust the position relationship between the mechanical home point and the mechanical zero point by setting the value of 0x607C in the object dictionary.

Figure 11- 23 Homing method control diagram



The conversion of user unit and encoder unit in home return mode via 0x6091 is illustrated as follows:

Figure 11- 24



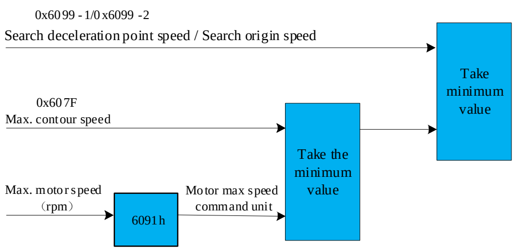
$$0x6091(\text{gear ratio}) = \frac{\text{motor resolution}}{\text{load resolution}}.$$

Relationship between 0x6063(motor position feedback) and 0x6064(user position feedback):

0x6063(encoder unit)=0x6064(command unit)×gear ratio.

The relationship between 0x6099-01 (search deceleration point speed) , 0x6099-02 (search home speed) and the corresponding maximum speed of the motor after transformation exists as follows:

Figure 11- 25



Motor speed (rpm) versus load shaft speed (command unit/s):

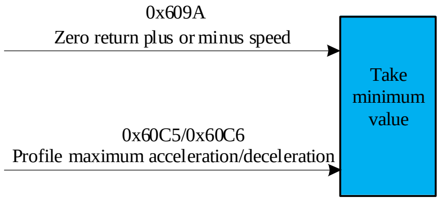
$$\text{Motor speed(rpm)} = \frac{\text{load shaft speed} \times \text{0x6091h}}{\text{encoder resolution}} \times 60$$

Example: gear ratio = 1:1 with 23-bit encoder. Motor speed = 500rpm (corresponding to 0x6099 (load shaft speed)) = 500*8388608/60 = 69905066 (command unit/s).

The following relationship exists between 0x609A (home return acceleration/deceleration) and

0x60C5/0x60C6 (maximum profile acceleration/deceleration):

Figure 11- 26



Example: gear ratio = 1:1 with 23-bit encoder. Motor deceleration = 500rpm/s (corresponding to 0x609A (load axis deceleration)) = 500*8388608/60 = 69905066 (command unit/s2).

Table 11- 71 Related Object Dictionaries

Index	Sub-index	Designation	Read/ Write	Data type	Unit	Range
0x603F	0x00	Error code	RO	UINT16	-	0~65535
0x6040	00	Control word	RW	UINT16	-	0~65535
0x6041	00	Status word	RO	UINT16	-	0~65535
0x6060	00	Running Mode	RW	INT8	-	0~7
0x6061	00	Mode display	RO	INT8	-	0~7
0x6064	00	Real position feedback	RO	INT32	Command unit	-
0x606C	00	Real speed feedback	RO	INT32	Command unit /s	-
0x6098	00	Home return method	RW	INT8	-	1~35
0x6099	01	High-speed search for deceleration points	RW	UINT32	Command unit /s	0~65535
	02	Low speed search for home signal	RW	UINT32	Command unit /s	1~500
0x609A	00	Home return Acceleration/deceleration	RW	UINT32	Command unit /s^2	0~(2^{32}-1)

Description of control word 6040h and status word 6041h in the homing method:

Table 11- 72

Object 6040h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6040_h**					
Designation	**Control Word**					

Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535
Mapping	Y	**Access**	RW	**Factory setting**	0

Function description

Bit definition of the control word:

Bit	Designation	Description
0	Servo ready	0: invalid; 1: valid.
1	Turn on main circuit power	0: invalid; 1: valid.
2	Quick stop	0: valid; 1: invalid.
3	Enable operation	0: invalid; 1: valid.
4	Enable home return	0: home return not enabled; 0→1: home return enabled; 1: home return in operation; 1→0: halt home return;
5~6	NA	
7	Fault reset	bit7 rising edge is valid; bit7 is held to 1. All other control instructions are invalid.
8	Halt	0: invalid; 1: valid.
9~10	NA	
11~15	Factory-defined	

Table 11- 73

Object 6041h				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041h**							
Designation	**Status Word**							
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535			
Mapping	Y	**Access**	RO	**Factory setting**	0			

Function description

Bit definition of status word:

Bit	Designation	Description
0	Servo ready	0: invalid; 1: valid.
1	Wait to enable servo	0: invalid; 1: valid.
2	Servo operation	0: invalid; 1: valid.
3	Faults	0:no faults;1: faults.
4	Turn on the main circuit power	0: invalid; 1: valid.
5	Quick stop	0: valid; 1: invalid.
6	Power-on and running allowed	0: invalid; 1: valid.
7	Warning	0: invalid; 1: valid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	Target reached	0: target position not reached; 1: target position reached.

	11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
	12	Home return completed	0: home return not completed; 1： home return completed;
	13	Home return error	0: no error occurred; 1: home return error at the origin.
	14	NA	
	15	Home return completed	0: home return not performed or not completed; 1: home return completed and reference point found.

The steps to turn on the home return mode are shown below:

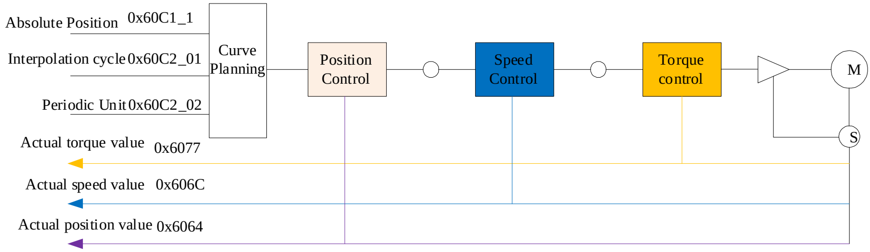
Table 11- 74

Item	Step	Parameter input	Status word display (6041h)
Home return parameter assignment	0	609Ah = 1000	0x0250
	1	6099_01h =1000	0x0250
	2	6099_02h = 100	0x0250
	3	6098h=0x01	0x0250
Control mode selection	4	6060h =0x06	0x0250
Servo enabling	5	6040h = 0x06	0x0231
	6	6040h = 0x07	0x0233
	7	6040h = 0x0F	0x0637
	8	6040h = 0x1F	0x0237
Home found	10	6040h = 0x1F	0x9637

11.2.9.5 Interpolation (IP)

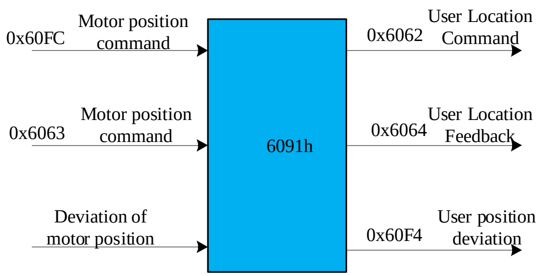
In interpolation position mode, the upper computer sends a position value (corresponding to the object dictionary [0x60C1]) during every synchronization cycle, which takes the value of the object dictionary 0x60C1 as the absolute position. For example, if the value of 0x60C1 is 0 at the beginning, then that’s the starting point of the absolute positiont. After the servo drive receives the interpolated position value in the first cycle, it starts to plan the curve path; when the second cycle comes and a new position value is sent, the path curve planned in the previous cycle is sent to the servo unit for running, and at the same time, it starts to plan a new position curve.

Figure 11- 27 Interpolation mode control block diagram



The conversion of user unit and encoder unit in interpolation mode via 0x6091 is illustrated below:

Figure 11- 28



As shown in Figure 11.29, at the moment t0, the upper computer sends an interpolated position command value, and the servo drive plans the motion trajectory POS0 according to the received interpolated position value and sends the motion trajectory POS0 to the execution unit at the moment t1, and at the same time plans the motion trajectory POS1 according to the new interpolated position value. At t2 the motion trajectory POS1 is executed again and at the same time the motion trajectory POS2 is planned and so on. The drive always plans the motion trajectory at the current moment for the next moment to ensure the smooth operation of the servo motor.

Figure 11- 29 Interpolation position diagram

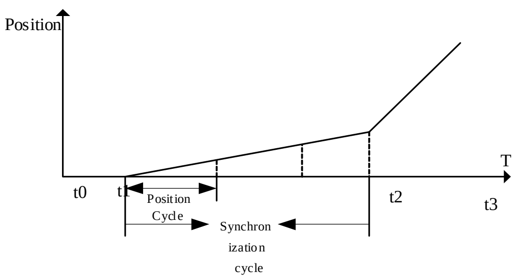


Table 11- 75 Related object dictionaries

Index	Sub-index	Designation	Read/ Write	Data type	Unit	Range
0x603F	00	Error code	RO	UINT16	—	0~65535
0x6040	00	Control word	RW	UINT16	—	0~65535
0x6041	00	Status word	RO	UINT16	—	0~65535
0x6060	00	Running mode	RW	INT8	—	0~7
0x6061	00	Operation mode display	RO	INT8	—	0~7
0x6064	00	Real position value	RO	INT32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x6065	00	Excessive position deviation threshold	RW	UINT32	Command unit	$0\sim(2^{32}-1)$
0x6067	00	Position reach threshold	RW	UINT32	Command unit	$0\sim(2^{32}-1)$
0x6068	00	Position reach time	RW	UINT16	ms	0~65535
0x607A	00	Target position value	RW	INT32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x607D	01	Min. software limit	RW	INT32	Command unit	$-2^{31}\sim(2^{31}-1)$
	02	Max. software limit	RW	INT32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x60C1	01	Absolute interpolation position value	RW	INT32	Command unit	$-2^{31}\sim(2^{31}-1)$
0x60C2	01	Interpolation cycle value	RW	UINT8	—	1~20
	02	Interpolation cycle unit	RW	INT8	—	-3

Description of control word 6040h and status word 6041h in interpolation mode:

Table 11- 76

Object 6040h			PP	PV	PT	HM	IP
Index	**6040h**						
Designation	**Control Word**						
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535		
Mapping	Y	**Access**	RW	**Factory setting**	0		
Function description	Bit definition of the control word:						

Bit	Designation	Description
0	Servo ready	0: invalid; 1: valid.
1	Turn on the main circuit electricity	0: invalid; 1: valid.
2	Quick stop	1:invalid；0:valid
3	Enable operation	0: invalid; 1: valid.

	4	Enable interpolation mode	0: halt interpolation; 1: enable interpolation.
	5~6	NA	
	7	Fault reset	bit7 rising edge is valid; bit7 is held to 1. All other control instructions are invalid.
	8	Halt	0: invalid; 1: valid.
	9~10	NA	
	11~15	Factory-defined	
	Note: each bit in the control word needs to be used together with other bits to form a control command.		

Table 11- 77

Object 6041h					**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041$_h$**								
Designation	**Status Word**								
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535				
Mapping	Y	**Access**	RO	**Factory setting**	0				

Function description

Bit definition of status word:

Bit	**Designation**	**Description**
0	Turn on the main circuit power	0: invalid; 1: valid.
1	Wait to enable servo	0: invalid; 1: valid.
2	Servo operation	0: invalid; 1: valid.
3	Faults	0:no faults;1: faults.
4	Turn on the main circuit power	0: invalid; 1: valid.
5	Quick stop	0: valid; 1: invalid.
6	Power-on and running allowed	0: invalid; 1: valid.
7	Warning	0: invalid; 1: valid.
8	Factory-defined	
9	Remote control	0: non-Canopen mode; 1: Canopen remote control mode.
10	Target reached	0: target position is not reached; 1: target position is reached.
11	Software internal position exceeds the limit	0: position command or feedback does not reach the internal position limit of the software; 1: position command or feedback reaches the internal position limit of the software.
12	Enable interpolation mode	0：interpolation mode not enabled； 1：interpolation mode enabled.

	13~14	NA	
	15	Home return completed	0: home return is not performed or not completed. 1: home return is completed and the reference point has been found.

The interpolation command value is planned by the upper computer planning during each synchronization cycle, and sends it through PDO to the servo driver to control the motor running. The interpolation mode is shown in the following table:

Table 11- 78

Item	Step	Parameter input	Status word display (6041h)
Interpolation cycle assignment	0	60C2_01h = 200(or 0xC8)	0x0250
	1	60C2_02h = -3(or 0xFD)	0x0250
Interpolation position assignment	2	60C1h = 10000	0x0250
Control mode selection	3	6060h = 0x07	0x0250
Servo enabling	4	6040h = 0x06	0x0231
	5	6040h = 0x07	0x0233
	6	6040h = 0x0F	0x0637
	7	6040h = 0x1F	0x0237
Positioning completed	8	6040h = 0x1F	0x0637

11.2.10 Object Dictionary

11.2.10.1 Description of Object Properties

Explanation of terms

Index: specifies the position of each object in the object dictionary, in hexadecimal (h).

Data type: See Table 11-79 for details.

Table 11- 79 Data type description

Data type	Range	Data length	DS301 value
Int8	-128~127	1 byte	2
Uint8	0~255	1 byte	5
Int16	-32768~+32767	2 bytes	3
Uint16	0~65535	2 bytes	6
Int32	-2147483648~+2147483647	4 bytes	4
Uint32	0~4294967295	4 bytes	7
String	ASCII	-	9

"Read/Write Type": Please refer to Table 11-80 for details.

Table 11- 80 Read and write type description

Read/Write	Description

RW	Read and write
WO	Write only
RO	Read only
CONST	Constant, read only

"Object structure": please refer to Table 11-81 for details.

Table 11- 81 Description of the object structure

Object structure	Description	DS301 value
VAR	Simple values containing the data types in Table 3-1	7
ARR	Data blocks of the same type	8
REC	Data blocks of different types	9

11.2.10.2 1000h Group Object List

Table 11- 82

Index	Sub-index	Designation	Object structure	Data type	Read/ Write	Mapping
1000h	-	Device type	VAR	Uint32	RO	N
1001h	-	Error register	VAR	Uint8	RO	N
1003h	-	Predefined error field	ARR	Uint32	RO	N
	$1\sim4_h$	Error field	-	Uint32	RW	N
1005h	-	COB-ID SYNC message	VAR	Uint32	RW	N
1006h	-	SYNC cycle	VAR	Uint32	RW	N
100Ch	-	Node guarding time	VAR	Uint16	RW	N
100Dh	-	Lifetime factor	VAR	Uint8	RW	N
1010h	-	Save parameters	ARR	Uint32	RW	N
	1_h	Save all object parameter	-	Uint32	RW	N
1011h	-	Restore default parameter	ARR	Uint32	RW	N
	1_h	Save all object parameter	-	Uint32	RW	N
1014h	-	Emergency message COB-ID	VAR	Uint32	RO	N
1016h	-	Consumer heartbeat time	ARR	-	-	-
	0_h	Maximum subindex supported	-	Uint8	RO	N
	1_h	Consumer heartbeat time	-	Uint32	RW	N
	2_h	Consumer heartbeat time	-	Uint32	RW	N
	3_h	Consumer heartbeat time	-	Uint32	RW	N
	4_h	Consumer heartbeat time	-	Uint32	RW	N
1017h	-	Producer heartbeat time	VAR	Uint16	RW	N
1018h	-	Device object description	REC	-	-	-
	0_h	Maximum subindex supported	-	Uint8	RO	N
	1_h	Manufacturer ID	-	Uint32	RO	N
	2_h	Device code	-	Uint32	RO	N
	3_h	Device revision number	-	Uint32	RO	N
1029h	-	Error behavior object	ARR	-	-	-

	0_h	Maximum subindex supported	-	Uint8	RO	N
	1_h	Communication error	-	Uint8	RW	N
1200h	-	SDO server parameter	REC	-	-	-
	0_h	Maximum subindex supported	-	Uint8	RO	N
	1_h	Client-to-Server COB-ID	-	Uint32	RO	N
	2_h	Server-to-Client COB-ID	-	Uint32	RO	N
1400h	-	RPDO1 mapping parameter	REC	-	-	-
	0_h	RPDO1 maximum subindex	-	Uint8	RO	N
	1_h	RPDO1 COB-ID	-	Uint32	RW	N
	2_h	RPDO1 transmission type	-	Uint8	RW	N
	3_h	Inhibit time (not supported)	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer (not supported)	-	Uint16	RW	N
1401h	-	RPDO2 mapping parameter	REC	-	-	-
	0_h	RPDO2 maximum subindex	-	Uint8	RO	N
	1_h	RPDO2 COB-ID	-	Uint32	RW	N
	2_h	RPDO2 transmission type	-	Uint8	RW	N
	3_h	Inhibit time (not supported)	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer (not supported)	-	Uint16	RW	N
1402h	-	RPDO3 mapping parameter	REC	-	-	-
	0_h	RPDO3 maximum subindex	-	Uint8	RO	N
	1_h	RPDO3 COB-ID	-	Uint32	RW	N
	2_h	RPDO3 transmission type	-	Uint8	RW	N
	3_h	Inhibit time (not supported)	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer (not supported)	-	Uint16	RW	N
1403h	-	RPDO4 mapping parameter	REC	-	-	-
	0_h	RPDO4 maximum subindex	-	Uint8	RO	N
	1_h	RPDO4 COB-ID	-	Uint32	RW	N
	2_h	RPDO4 transmission type	-	Uint8	RW	N
	3_h	Inhibit time (not supported)	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer (not supported)	-	Uint16	RW	N
1600h	-	RPDO1 mapping parameter	REC	-	-	-
	0_h	RPDO1valid mapping number	-	Uint8	RW	N
	1_h	RPDO1 mapping object 1	-	Uint32	RW	N
	2_h	RPDO1 mapping object 2	-	Uint32	RW	N
	3_h	RPDO1 mapping object 3	-	Uint32	RW	N
	4_h	RPDO1 mapping object 4	-	Uint32	RW	N
1601h	-	RPDO2 mapping parameter	REC	-	-	-
	0_h	RPDO2 valid mapping number	-	Uint8	RW	N
	1_h	RPDO2 mapping object 1	-	Uint32	RW	N
	2_h	RPDO2 mapping object 2	-	Uint32	RW	N

	3_h	RPDO2 mapping object 3	-	Uint32	RW	N
	4_h	RPDO2 mapping object 4	-	Uint32	RW	N
1602h	-	RPDO3 mapping parameter	REC	-	-	-
	0_h	RPDO3 valid mapping number	-	Uint8	RW	N
	1_h	RPDO3 mapping object 1	-	Uint32	RW	N
	2_h	RPDO3 mapping object 2	-	Uint32	RW	N
	3_h	RPDO3 mapping object 3	-	Uint32	RW	N
	4_h	RPDO3 mapping object 4	-	Uint32	RW	N
1603h	-	RPDO4 mapping parameter	REC	-	-	-
	0_h	RPDO4 valid mapping number	-	Uint8	RW	N
	1_h	RPDO4 mapping object 1	-	Uint32	RW	N
	2_h	RPDO4 mapping object 2	-	Uint32	RW	N
	3_h	RPDO4 mapping object 3	-	Uint32	RW	N
	4_h	RPDO4 mapping object 4	-	Uint32	RW	N
1800h	-	TPDO1 parameter	REC	-	-	-
	0_h	TPDO1 maximum subindex	-	Uint8	RO	N
	1_h	TPDO1 COB-ID	-	Uint32	RW	N
	2_h	TPDO1 transmission type	-	Uint8	RW	N
	3_h	Inhibit time	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer	-	Uint16	RW	N
1801h	-	TPDO2 parameter	REC	-	-	-
	0_h	TPDO2 maximum subindex	-	Uint8	RO	N
	1_h	TPDO2 COB-ID	-	Uint32	RW	N
	2_h	TPDO2 transmission type	-	Uint8	RW	N
	3_h	Inhibit time	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer	-	Uint16	RW	N
1802h	-	TPDO3 parameter	REC	-	-	-
	0_h	TPDO3 maximum subindex	-	Uint8	RO	N
	1_h	TPDO3 COB-ID	-	Uint32	RW	N
	2_h	TPDO3 transmission type	-	Uint8	RW	N
	3_h	Inhibit time	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer	-	Uint16	RW	N
1803h	-	TPDO4 parameter	REC	-	-	-
	0_h	TPDO1 maximum subindex	-	Uint8	RO	N
	1_h	TPDO4 COB-ID	-	Uint32	RW	N
	2_h	TPDO4 transmission type	-	Uint8	RW	N
	3_h	Inhibit time	-	Uint16	RW	N
	4_h	NA	-	Uint8	RW	N
	5_h	Event timer	-	Uint16	RW	N
1A00h	-	TPDO1 mapping parameter	REC	-	-	-
	0_h	TPDO1 valid mapping number	-	Uint8	RW	N

	1_h	TPDO1 mapping object 1	-	Uint32	RW	N
	2_h	TPDO1 mapping object 2	-	Uint32	RW	N
	3_h	TPDO1 mapping object 3	-	Uint32	RW	N
	4_h	TPDO1 mapping object 4	-	Uint32	RW	N
1A01h	-	TPDO4 mapping parameter	REC	-	-	-
	0_h	TPDO2 valid mapping number	-	Uint8	RW	N
	1_h	TPDO2 mapping object 1	-	Uint32	RW	N
	2_h	TPDO2 mapping object 2	-	Uint32	RW	N
	3_h	TPDO2 mapping object 3	-	Uint32	RW	N
	4_h	TPDO2 mapping object 4	-	Uint32	RW	N
1A02h	-	TPDO3 mapping parameter	REC	-	-	-
	0_h	TPDO3 valid mapping number	-	Uint8	RW	N
	1_h	TPDO3 mapping object 1	-	Uint32	RW	N
	2_h	TPDO3 mapping object 2	-	Uint32	RW	N
	3_h	TPDO3 mapping object 3	-	Uint32	RW	N
	4_h	TPDO3 mapping object 4	-	Uint32	RW	N
1A03h	-	TPDO4 mapping parameter	REC	-	-	-
	0_h	TPDO4 valid mapping number	-	Uint8	RW	N
	1_h	TPDO4 mapping object 1	-	Uint32	RW	N
	2_h	TPDO4 mapping object 2	-	Uint32	RW	N
	3_h	TPDO4 mapping object 3	-	Uint32	RW	N
	4_h	TPDO4 mapping object 4	-	Uint32	RW	N

11.2.10.3 2000h Group Object List

2000h group object dictionary is the mapping of internal parameters of the drive. The object dictionaries 2000h~2007h correspond to the parameter groups of Pn0xx~Pn7xx respectively; 2E00h~2E03h correspond to the monitoring parameters of Un0xx~Un3xx. The specific function code of the drive corresponds to the sub-index of the object dictionary of the 2000h group, and the specific correspondence rule is that the last two digits of the function code plus 1 is the corresponding object dictionary sub-index.

The following table shows the correspondence between the 2000h object dictionary index number and the function code of the drive, the specific meaning of the function code is detailed in "Chapter 9 Parameter Description"and"Chapter 8 Monitoring Parameters".

Table 11- 83

Cautions	
	● The last two digits of the function code correspond to the subindex. The function code is a hexadecimal number, and so is the subindex. **Example**: When reading or writing function code Pn299, the corresponding object

	dictionary is 2002_9Ah.

11.2.10.4 6000h Group Object List

The Canopen6000h group object dictionary assignment is shown in the following table:

Table 11- 84

Index	Sub-index	Designation	Access	Map	Data type	Unit	Range
603Fh	00	Error code	RO	Y	UINT16	-	UINT16
6040h	00	Control word	RW	Y	UINT16	-	UINT16
6041h	00	Status word	RO	Y	UINT16	-	UINT16
605Ah	00	Quick stop method	RO	Y	INT16		INT16
605Dh	00	Halt stop method	RO	Y	INT16		INT16
6060h	00	Running mode	RW	Y	INT8	-	INT8
6061h	00	Mode display	RO	Y	INT8	-	INT8
6062h	00	User position command	RO	Y	INT32	Command unit	INT32
6063h	00	Motor position feedback	RO	Y	INT32	Encoder units	INT32
6064h	00	User position feedback	RO	Y	INT32	Command unit	INT32
6065h	00	Excessive position deviation threshold	RW	Y	UINT32	Command unit	UINT32
6067h	00	Position reach threshold	RW	Y	UINT32	Command unit	UINT32
6068h	00	Position reach time	RW	Y	UINT16	ms	UINT16
606Bh	00	Speed command value	RO	Y	INT32	Command unit /s	INT32
606Ch	00	Real speed feedback value	RO	Y	INT32	Command unit /s	INT32
606Dh	00	Speed reach threshold	RW	Y	UINT16	0.1rpm	UINT16
606Eh	00	Speed reach time window	RW	Y	UINT16	ms	UINT16
606Fh	00	Zero-speed threshold	RW	Y	UINT16	0.1rpm	UINT16

6070h	00	Zero-speed time window	RW	Y	UINT16	ms	UINT16
6071h	00	Target torque	RW	Y	INT16	0.1%	INT16
6072h	00	Maximum torque	RW	Y	UINT16	0.1%	UINT16
6074h	00	Torque command	RO	Y	INT16	0.1%	INT16
6075h	00	Rated current	RO	Y	UINT32	mA	UINT32
6076h	00	Rated torque	RO	Y	UINT32	mNm	UINT32
6077h	00	Real torque	RO	Y	INT16	0.1%	INT16
6078h	00	Real current	RO	Y	INT16	0.1%	INT16
607Ah	00	Target position	RW	Y	INT32	Command unit	INT32
607Ch	00	Home return bias	RW	Y	INT32	Command unit	INT32
607Dh	01	Min. software limit	RW	Y	INT32	Command unit	INT32
	02	Max. software limit	RW	Y	INT32	Command unit	INT32
607F	00	Max speed limit	RW	Y	UINT32	Command unit /s	UINT32
6080h	00	Max. motor speed	RO	Y	UINT32	rpm	UINT32
6081h	00	Profile position target speed	RW	Y	UINT32	Command unit /s	UINT32
6083h	00	Profile acceleration	RW	Y	UINT32	Command unit /s^2	UINT32
6084h	00	Profile deceleration	RW	Y	UINT32	Command unit /s^2	UINT32
6085h	00	Profile emergency stop deceleration	RW	Y	UINT32	Command unit /s^2	UINT32
6086h	00	Motor running profile type	RW	Y	UINT16		UINT16
6087h	00	Torque smoothing time	RW	Y	UINT16	ms	UINT16
6091h	01	Electronic gear numerator	RW	Y	UINT32	-	UINT32
	02	Electronic gear denominator	RW	Y	UINT32	-	UINT32
6098h	00	Home return method	RW	Y	UINT8	-	UINT8
6099h	01	Home return high speed setting	RW	Y	UINT32	Command unit/s	UINT32
	02	Home return low speed setting	RW	Y	UINT32	Command unit/s	UINT32

609Ah	00	Zero return acceleration/deceleration	RW	Y	UINT32	Command unit/s^2	UINT32
60C1h	01	Absolute interpolation position value	RW	Y	INT32	Command unit	INT32
60C2h	01	Interpolation cycle	RW	Y	UINT8	-	UINT8
	02	Interpolation cycle unit	RW	Y	INT8	-	INT8
60C5h	00	Max. profile acceleration	RW	Y	UINT32	Command unit/s^2	UINT32
60C6h	00	Max. profile deceleration	RW	Y	UINT32	Command unit/s^2	UINT32
60E0h	00	Forward torque limit	RW	Y	UINT16	0.1%	UINT16
60E1h	00	Reverse torque limit	RW	Y	UINT16	0.1%	UINT16
60F4h	00	User position deviation	RO	Y	INT32	Command unit	INT32
60FCh	00	Motor position command	RO	Y	INT32	编码器单位	INT32
60FDh	00	Digital input status	RO	Y	UINT16	-	UINT16
60FEh	00	No. of digital output	RO	N	UINT8	-	1
	01	Digital output status	RO	Y	UINT16	-	UINT16
60FFh	00	Profile speed target value	RW	Y	UINT32	Command unit/s	UINT32
6502h	00	Running mode of servo drive	RO	Y	UINT16	-	UINT16

11.2.10.5 1000h Detailed Object Description

Table 11- 85

Object 1000h					
Index	**1000_h**				
Designation	**Device Type**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	0x20192

<table>
<tr><td>Function description</td><td>The Device type parameter is used to describe the device subprotocol or application specification.
<table><tr><th>Bit</th><th>Designation</th><th>Description</th></tr><tr><td>0~15</td><td>Device sub-protocol</td><td>402(0x192): device sub-protocol</td></tr><tr><td>16~23</td><td>Type</td><td>02: servo drive</td></tr><tr><td>25~31</td><td>Mode</td><td>Factory-defined</td></tr></table></td></tr>
</table>

<table>
<tr><td colspan="6">Object 1001h</td></tr>
<tr><td>Index</td><td colspan="5">1001h</td></tr>
<tr><td>Designation</td><td colspan="5">Error Register</td></tr>
<tr><td>Object structure</td><td>VAR</td><td>Data type</td><td>Uint8</td><td>Data range</td><td>Uint8</td></tr>
<tr><td>Mapping</td><td>NO</td><td>Access</td><td>RO</td><td>Factory setting</td><td>0x0</td></tr>
<tr><td>Function description</td><td colspan="5">Contain error type information by bit, as shown in the following table:
<table><tr><th>Bit</th><th>Designation</th><th>Bit</th><th>Designation</th></tr><tr><td>0</td><td>General</td><td>4</td><td>Communication</td></tr><tr><td>1</td><td>Current</td><td>5</td><td>Sub-protocol</td></tr><tr><td>2</td><td>Voltage</td><td>6</td><td>NA</td></tr><tr><td>3</td><td>Temperature</td><td>7</td><td>Factory-defined</td></tr></table>When an error occurs, the corresponding bit of the error is "1", and bit 0 must be "1".</td></tr>
<tr><td colspan="6">Object 1003h</td></tr>
<tr><td>Index</td><td colspan="5">1003h</td></tr>
<tr><td>Designation</td><td colspan="5">Pre-defined Error Field</td></tr>
<tr><td>Object structure</td><td>ARR</td><td>Data type</td><td>Uint32</td><td>Data range</td><td>Uint32</td></tr>
<tr><td>Mapping</td><td>NO</td><td>Access</td><td>RO</td><td>Factory setting</td><td>-</td></tr>
<tr><td>Sub-index</td><td colspan="5">00h</td></tr>
<tr><td>Designation</td><td colspan="5">Number of Errors</td></tr>
<tr><td>Object structure</td><td>-</td><td>Data type</td><td>Uint32</td><td>Data range</td><td>0~4</td></tr>
<tr><td>Mapping</td><td>NO</td><td>Access</td><td>RW</td><td>Factory setting</td><td>0x0</td></tr>
<tr><td>Function description</td><td colspan="5">Only 0 can be written here, and all error records are cleared.</td></tr>
<tr><td>Sub-index</td><td colspan="5">1~4h</td></tr>
<tr><td>Designation</td><td colspan="5">Standard Error Field</td></tr>
<tr><td>Object structure</td><td>-</td><td>Data type</td><td>Uint32</td><td>Data range</td><td>Uint32</td></tr>
<tr><td>Mapping</td><td>NO</td><td>Access</td><td>RW</td><td>Factory setting</td><td>0x0</td></tr>
<tr><td>Function description</td><td colspan="5">When the subindex is 0, it is not readable; when there is an error, the error is stored in the following format:
MSB LSB
<table><tr><td>31 16</td><td>15 0</td></tr><tr><td>Factory error code</td><td>Standard error code</td></tr></table></td></tr>
<tr><td colspan="6">Object 1005h</td></tr>
<tr><td>Index</td><td colspan="5">1005h</td></tr>
<tr><td>Designation</td><td colspan="5">COB-ID SYNC Message</td></tr>
</table>

Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	0x80
Function description	Only 0x80 and 0x40000080 can be written. When 0x80 is written, synchronization is off; When 0x40000080 is written, synchronization is on. The synchronization cycle 1006h must be configured to be non-zero before activating synchronization.				
Object 1006h					
Index	**1006h**				
Designation	**SYNC Cycle**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	0x0
Function description	Cycle time in units of 125us for synchronization.				
Object 1008h					
Index	**1008h**				
Designation	**Manufacturer Device Name**				
Object structure	REC	**Data type**	Uint8	**Data range**	-
Mapping	NO	**Access**	RO	**Factory setting**	Servo Device
Object 100Ah					
Index	**100Ah**				
Designation	**Software Version**				
Object structure	REC	**Data type**	Uint8	**Data range**	-
Mapping	NO	**Access**	RO	**Factory setting**	Determined by model
Object 100Ch					
Index	**100Ch**				
Designation	**Guard Time**				
Object structure	VAR	**Data type**	Uint16	**Data range**	Uint16
Mapping	NO	**Access**	RW	**Factory setting**	0x0
Function description	For SYNC only in ms. Used in conjunction with lifetime factor for node protection.				
Object 100Dh					
Index	**100Dh**				
Designation	**Lifetime Factor**				
Object structure	VAR	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RW	**Factory setting**	0x0
Function description	Must be larger than 1 when used.				

Table 11- 86

Object 1010h	
Index	**1010h**
Designation	**Save Parameters**

Object structure	ARR	Data type	Uint32	Data range	Uint32
Mapping	NO	Access	RW	Factory setting	0x0
Function description	Save parameter is to save the current value to EEPROM, and the next time the EEPROM is loaded (re-powered, node or communication reset), the saved value will be loaded. When users need to save a parameter, write "save" according to ASCII code in addition to specifying the subindex corresponding to the save area. Other values written will not save the parameters successfully. The correspondence of writing is as follows: MSB LSB				

ASCII	e	v	a	s
Hexadecimal	65h	76	61h	73h

The corresponding subindex read return value indicates how the parameter is saved in the subindex. Return value format and meaning is as follows: LSB

MSB

31 2	1	0
NA	0/1	0/1

Value	Description
0	No automatic saving of parameters, and no saving of parameters by command
1	Save parameters by command only, no automatic saving
2	Only save parameters automatically, no saving of parameters by command
3	Save parameters by command and automatically

Object 1011h

Index	1011h				
Designation	Restore Default Parameters				
Object structure	ARR	Data type	Uint32	Data range	-
Mapping	NO	Access	RW	Factory setting	

11 Communication

Function description (continued)

Restoring default parameters is restoring the default parameters to the EEPROM and does not take effect immediately. Next time when EEPROM is loaded (power-on, node or communication reset), the default values (factory settings) are loaded. To restore the default parameters, in addition to specifying the sub-index of the recovery area, users need to write "load" according to ASCII code, and writing other values will not restore the default values successfully.

The correspondence of writing is as follows:

MSB LSB

ASCII	d	a	o	l
Hexadecimal	64h	61h	6Fh	6Ch

The corresponding sub-index read return value indicates the way the sub-index saves the parameters. Return format and meaning are as follows:

MSB LSB

31 … 1	0
NA	0/1

Value	Description
0	Default parameters cannot be restored
1	Default parameters can be restored

Object 1014h

Index	1014h				
Designation	COB-ID Emergency Message				
Object structure	VAR	Data type	Uint32	Data range	Uint32
Mapping	NO	Access	RW	Factory setting	0x80+Node-ID

Function description

0 on Bit31 means Emergency (EMCY) function is on (servo will send EMCY command);

1 on Bit31 means Emergency(EMCY) function is off (servo will not send EMCY command).

MSB

LSB

31	30 … 11	10 … 0
0/1	0x0	11-bits verified COB-ID

When an emergency message takes effect, its COB-ID must be consistent with this object.

Object 1016h

Index	1016h				
Designation	Consumer Heartbeat Time				
Object structure	ARR	Data type	Uint32	Data range	Uint32
Mapping	NO	Access	RW	Factory setting	

Function description

The parameters include the address of the monitored node and the actual consumer time, and this time must be greater than the heartbeat producer time (in ms) of the corresponding node. It is not possible to set two consumer times for the same node.
The parameters are as follows:
MSB
LSB

31 24	23 16	15 0
NA	Monitored address	Monitored time

The corresponding sub-index read return value indicates which way the sub-index restores the default parameters.

Sub-index	**00h**				
Designation	**Entry number**				
Object structure	-	**Data type**	Uint8	**Data range**	1
Mapping	NO	**Access**	RO	**Factory setting**	1
Function description	Only 0 can be written, and all error records are cleared.				
Sub-index	**01h**				
Designation	**Consumer Heartbeat Time**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	0
Function description	Save all parameters of the object dictionary list.				
Object 1017h					
Index	**1017h**				
Designation	**Producer Heartbeat Time**				
Object structure	VAR	**Data type**	Uint16	**Data range**	Uint16
Mapping	NO	**Access**	RW	**Factory setting**	
Function description	Unit (ms).				
Object 1018h					
Index	**1018h**				
Designation	**Device Object Description**				
Object structure	REC	**Data type**	Uint16	**Data range**	-
Mapping	NO	**Access**	RO	**Factory setting**	
Sub-index	**00h**				
Designation	**Entry number**				
Object structure	-	**Data type**	Uint8	**Data range**	3
Mapping	NO	**Access**	RO	**Factory setting**	3
Sub-index	**01h**				
Designation	**Manufacturer ID**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	0x3B9

Function description	A unique number assigned by the CiA.				
Sub-index	**02_h**				
Designation	**Device Code**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	-
Function description	The equipment code corresponds to the product series and product model on the electronic label, and the correspondence is as follows: MSB LSB **31 16** | **15 0** Product Series | Product Model				
Sub-index	**03_h**				
Designation	**Device Revision Number**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	-
Function description	Correspond to the software version number 100Ah, the specific meaning is as follows: MSB LSB **31 16** | **15 0** Main revision version | Sub-revision version				

Table 11- 87

Object 1029h					
Index	**1029_h**				
Designation	**Error Behavior Object**				
Object structure	ARR	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	The state control to which the NMT of Canopen communication needs to automatically shift when different types of errors occur. NMT shifts to different states according to different values. **Value** | **Description** 0 | Turns to the pre-running state from the current running state. 1 | Keep the current state. 2 | Enter the stop state. Others | NA。				
Sub-index	**00_h**				
Designation	**Largest Sub-index Supported**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RO	**Factory setting**	1
Sub-index	**01_h**				
Designation	**Communication Error**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8

Mapping	NO	**Access**	RW	**Factory setting**	0
Function description	Communication errors included include: NMT error control timeout, PDO length error, bus detachment, etc.				
Object 1200h					
Index	**1200_h**				
Designation	**SDO Server Parameter**				
Object structure	REC	**Data type**	-	**Data range**	-
Mapping	NO	**Access**	RO	**Factory setting**	-
Function description	The highest bit is "0" to indicate that the SDO is valid, and the highest bit is "1" to indicate that the SDO is invalid. The default SDO is always present and is a read-only constant. MSB LSB				

31	30 11	10 0
0/1	0x0	11-bits verified COB-ID

Sub-index	**00_h**				
Designation	**Entry number**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RO	**Factory setting**	2
Sub-index	**01_h**				
Designation	**Client-to- Server(rx) COB-ID**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	0x600+Node-ID
Sub-index	**02_h**				
Designation	**Server-to-Client(tx) COB-ID**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RO	**Factory setting**	0x580+Node-ID

Table 11- 88

Object1400h: RPDO1 Communication Parameter **Object1402h: RPDO2 Communication Parameter** **Object1403h: RPDO3 Communication Parameter** **Object1404h: RPDO4 Communication Parameter**					
Index	**1400_h~1403_h**				
Designation	**RPDO Message COB-ID**				
Object structure	REC	**Data type**	-	**Data range**	-
Mapping	NO	**Access**	RW	**Factory setting**	-
Sub-index	**00_h**				
Designation	**Largest Sub-index Supported**				
Object structure	-	**Data type**	Uint8	**Data range**	0~2
Mapping	NO	**Access**	RO	**Factory setting**	2
Sub-index	**01_h**				
Designation	**RPDO COB-ID**				

Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	See Function description
Function description	Only the highest bit can be changed. A "0" indicates that the PDO is valid, and a "1" indicates that the PDO is invalid. MSB LSB 31 | 30 … 11 | 10 … 0 0/1 | 0 | 11-bits verified COB-ID Factory setting: (Node-ID default value is 1): 1400h：0x80000200 + Node-ID 1401h：0x80000300 + Node-ID 1402h：0x80000400 + Node-ID 1403h：0x80000500 + Node-ID				
Sub-index	**02h**				
Designation	**RPDO Reception type**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RW	**Factory setting**	0
Function description	This value can only be modified when the PDO is invalid. Different values represent different PDO transmission types, as in the following table: **Value** | **Designation** 0 | Synchronous non-cycle 1~240 | Synchronous cycle 254,255 | Asynchronous non-cycle				

Table 11- 89

Object 1600h: RPDO1 Mapping Parameter **Object 1601h: RPDO2 Mapping Parameter** **Object 1602h: RPDO3 Mapping Parameter** **Object 1603h: RPDO4 Mapping Parameter**					
Sub-index	**1600h~1603h**				
Designation	**RPDO Mapping Parameter**				
Object structure	REC	**Data type**	-	**Data range**	-
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	This object can be modified only when PDO is off. The total bit length of the mapped object must not exceed 64 bits, and only per-byte mapping is supported, not per-bit mapping.				
Sub-index	**00h**				
Designation	**Number of valid mapped objects in PDO**				
Object structure	-	**Data type**	Uint8	**Data range**	0~4
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	When writing 0, other sub-index mapping object is invalid.				
Sub-index	**1h~4h**				

Designation	RPDO Mapped Object				
Object structure	-	Data type	Uint32	Data range	Uint32
Mapping	NO	Access	RW	Factory setting	-
Function description	The mapped object content, index and sub-index must exist in the object dictionary list, in writable state and be mappable. Write the corresponding sub-index in the following format: MSB LSB				

31 16	15 8	7 0
Index	Sub-index	Object length

RPDOdefault mapping content:

(1)RPDO1(**1600h**)

Table 11- 90

Sub-index	Value	Description
0	1	Map one object
1	0x60400010	Command word

(2)RPDO2(**1601h**)

Table 11- 91

Sub-index	Value	Description
0	2	Map two objects
1	0x60410010	Control word
2	0x60600008	Running mode selection

(3)RPDO3(**1602h**)

Table 11- 92

Sub-index	Value	Description
0	2	Map two objects
1	0x60410010	Control word
2	0x607A0020	Target position (position command)

(4)RPDO4(**1603h**)

Table 11- 93

Sub-index	Value	Description
0	2	Map two objects
1	0x60410010	Control word
2	0x60FF0020	Target speed (speed command)

Table 11- 94

Object1800h: TPDO1 Communication Parameter Object1801h: TPDO2 Communication Parameter Object1802h: TPDO3 Communication Parameter Object1803h: TPDO4 Communication Parameter		
Index	1800h~1803h	
Designation	TPDO Communication Parameter	

Object structure	REC	**Data type**	-	**Data range**	-
Mapping	NO	**Access**	RW	**Factory setting**	-
Sub-index	**00h**				
Designation	**Largest Sub-index Supported**				
Object structure	-	**Data type**	Uint8	**Data range**	0~4
Mapping	NO	**Access**	RO	**Factory setting**	5
Sub-index	**01h**				
Designation	**TPDO COB-ID**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	See Function description

Function description

Only the highest bit can be changed. A "0" indicates that the TPDO is valid, and a "1" indicates that the PDO is not valid.

MSB LSB

31	30 11	10 0
0/1	0	11-bits verified COB-ID

Factory setting: (Node-ID default is 1):
1800h：0x80000180 + Node-ID
1801h：0x80000280 + Node-ID
1802h：0x80000380 + Node-ID
1803h：0x80000480 + Node-ID

Sub-index	**02h**				
Designation	**TPDO Transmission type**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RW	**Factory setting**	255

Function description

This value can only be modified if the PDO is invalid. Different values represent different PDO transmission types, as shown in the following table:

Value	Description
0	Synchronous non-cycle
1~240	Synchronous cycle
255	Asynchronous cycle

Sub-index	**03h**				
Designation	**Inhibit Time**				
Object structure	-	**Data type**	Uint16	**Data range**	Uint16
Mapping	NO	**Access**	RW	**Factory setting**	8
Function description	This object can only be modified when the PDO is invalid. unit is 125us. Note: The inhibit time is invalid when set to 0.				
Sub-index	**04h**				
Designation	**Reserved**				
Object structure	-	**Data type**	Uint8	**Data range**	Uint8
Mapping	NO	**Access**	RW	**Factory setting**	0
Sub-index	**05h**				
Designation	**Event Timer**				
Object structure	-	**Data type**	Uint16	**Data range**	Uint16

Mapping	NO	**Access**	RW	**Factory setting**	2
Function description	This object can be modified only when the PDO is invalid. unit is 1ms. Note: when set to 0, the time timer is invalid.				

Table 11- 95

Object 1A00h: TPDO1 Mapping Parameter **Object 1A01h: TPDO2 Mapping Parameter** **Object 1A02h: TPDO3 Mapping Parameter** **Object 1A03h: TPDO4 Mapping Parameter**					
Index	**$1A00_h$~$1A03_h$**				
Designation	**TPDO Mapping Parameter**				
Object structure	REC	**Data type**	-	**Data range**	-
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	This object can be modified only when the PDO is invalid. The total bit length of the mapped object must not exceed 64 bits, and only per-byte mapping is supported, not per-bit mapping.				
Sub-index	**00_h**				
Designation	**Number of valid mapped objects in PDO**				
Object structure	-	**Data type**	Uint8	**Data range**	0~4
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	When written 0, the sub-index mapping object is invalid.				
Sub-index	**1_h~4_h**				
Designation	**TPDO Mapped Object(Application Object)**				
Object structure	-	**Data type**	Uint32	**Data range**	Uint32
Mapping	NO	**Access**	RW	**Factory setting**	-
Function description	The mapped object content, index and subindex, must exist in the object dictionary list, writable mappable. Write the corresponding mapped Object in the following format: MSB LSB **31 16** | **15 8** | **7 0** Index | Sub-index | Object length				

TPDOdefault mapping content：

(1)TPDO1(**$1A00_h$**)

Table 11- 96

Sub-index	**Value**	**Description**
0	1	Map one object
1	0x60410010	Status word

(2)TPDO2(**$1A01_h$**)

Table 11- 97

Sub-index	**Value**	**Description**
0	2	Map two objects

1	0x60410010	Status word
2	0x60610008	Current running mode

(3)TPDO3(**1A02**$_h$)

Table 11- 98

Sub-index	**Value**	**Description**
0	2	Map two objects
1	0x60410010	Status word
2	0x60640020	Current position

(4)TPDO4(**1A03**$_h$)

Table 11- 99

Sub-index	**Value**	**Description**
0	2	Map two objects
1	0x60410010	Status word
2	0x606C0020	Current speed

11.2.10.6 6000h Detailed Object Description

Table 11- 100

Object 603Fh				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**603F**$_h$							
Designation	**Error Code**							
Object structure	VAR	**Data type**	Uint16	**Data range**		0~65535		
Mapping	Y	**Access**	RO	**Factory setting**		-		
Function description	The fault code is the error that occurred the last time. See the fault list for details.							
Object 6040h				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6040**$_h$							
Designation	**Control Word**							
Object structure	VAR	**Data type**	Uint16	**Data range**		0~65535		
Mapping	Y	**Access**	RW	**Factory setting**		0		

<table>
<tr><td rowspan="14">Function description</td><td colspan="3">Bit definition of control word:</td></tr>
<tr><td>Bit</td><td>Designation</td><td>Description</td></tr>
<tr><td>0</td><td>Servo ready</td><td>0: invalid; 1: valid.</td></tr>
<tr><td>1</td><td>Turn on the main circuit electricity</td><td>0: invalid; 1: valid.</td></tr>
<tr><td>2</td><td>Quick Shutdown</td><td>1： invalid； 0: valid.。</td></tr>
<tr><td>3</td><td>Servo operation</td><td>0: invalid; 1: valid.</td></tr>
<tr><td>4~6</td><td>Mode-related</td><td>
<table>
<tr><td rowspan="2">Bit</td><td colspan="4">Running mode</td></tr>
<tr><td>PP</td><td>PV</td><td>PT</td><td>HM</td></tr>
<tr><td>4</td><td>New position rising edge triggered</td><td>NA</td><td>NA</td><td>Home return on</td></tr>
<tr><td>5</td><td>0:non-immediate update
1:immediate update</td><td>NA</td><td>NA</td><td>NA</td></tr>
<tr><td>6</td><td>0:absolute position
1:relative position</td><td>NA</td><td>NA</td><td>NA</td></tr>
</table>
</td></tr>
<tr><td>7</td><td>Fault reset</td><td>bit7 rising edge is valid;
bit7 is held to 1, all other control instructions are invalid</td></tr>
<tr><td>8</td><td>Halt</td><td>0: invalid; 1: valid.</td></tr>
<tr><td>9~10</td><td>NA</td><td></td></tr>
<tr><td>11~15</td><td>Factory-defined</td><td></td></tr>
<tr><td colspan="3">Note: each bit in the control word needs to be used together with other bits to form a control command.</td></tr>
</table>

Object 6041h				**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6041h**							
Designation	**Status Word**							
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535			
Mapping	Y	**Access**	RO	**Factory setting**	0			

<table>
<tr><td rowspan="2">Function description</td><td>Bit definition of status word:</td></tr>
<tr><td>

Bit	Designation	Bit definition
0	Servo ready	0: invalid;1: valid.
1	Wait to enable servo	0: invalid;1: valid.
2	Servo operation	0: invalid;1: valid.
3	Fault	0: no faults; 1:faults。
4	Turn on the main circuit electricity	0: invalid;1: valid.
5	Quick stop	0: valid; 1: invalid。
6	Power-on and running allowed	0: invalid;1: valid.
7	Warning	0: invalid;1: valid.
8	Factory-defined	
9	Remote control	0： non-Canopen mode； 1： Canopen remote control mode.
10	Target reached	Speed mode: 0: target speed not reached; 1: target speed reached. Position mode: 0: target position not reached; 1: target position reached.
11	Software internal position exceeds the limit	0: position command or feedback does not reach the software internal position limit; 1: position command or feedback reaches the software internal position limit.
12~13	Control mode-related	
14	NA	
15	Home return completed	0: home return not performed or not completed. 1: home return completed and the reference point found.

</td></tr>
</table>

Table 11- 101

Object 605Ah		PP	PV	PT	HM	IP
Index	**605Ah**					
Designation	**Quick Stop Selection**					
Object structure	VAR	**Data type**	Int16	**Data range**	0~2	

Mapping	NO	Access	RW	Factory setting	2
Function description	**Value**	**Control mode display**			
	0	Free stop, keep free running after the stop is completed			
	1	Stop at the set deceleration ramp of 6084h (hm: 609Ah) and keep free running after stopping is completed.			
	2	Stop at the deceleration ramp set at 6085h and keep free running after the stop is completed.			

Object 605Dh		PP	PV	PT	HM	IP
Index	605Dh					
Designation	Halt Stop Selection					
Object structure	VAR	Data type	Int16	Data range	1~3	
Mapping	NO	Access	RW	Factory setting	1	
Function description	**Value**	**Control mode display**				
	1	Stop at the set deceleration ramp of6084h/6087h(hm： 609Ah) and lock the position after stopping is completed.				
	2	Stop at the set deceleration ramp of6084h/6087h(hm： 609Ah) and lock the position after stopping is completed.				
	3	Stop via emergency torque and lock the position after stopping is completed.				

Table 11- 102

Object 6060h		PP	PV	PT	HM	IP
Index	6060h					
Designation	Running Mode Selection					
Object structure	VAR	Data type	Int8	Data range	0~7	
Mapping	Y	Access	RW	Factory setting	1	
Function description	Set running mode:					
	Value	**Description**				
	0	NA				
	1	Profile position mode (PP)				
	3	Profile velocity mode (PV)				
	4	Profile torque mode (PT)				
	6	Homing mode (HM)				
	7	Interpolation mode (IP)				

Object 6061h		PP	PV	PT	HM	IP
Index	6061h					
Designation	Modes Display					
Object structure	VAR	Data type	Int8	Data range	0~7	

Mapping	Y	**Access**	RO	**Factory setting**	0
Function description					

Value	**Control mode display**
0	NA
1	Profile position mode (PP)
3	Profile velocity mode (PV)
4	Profile torque mode (PT)
6	Homing mode (HM)
7	Interpolation mode (IP)

Object 6062h				**PP**	**HM**	**IP**
Index	**6062h**					
Designation	**Position Command**					
Object structure	VAR	**Data type**	Int 32	**Data range**	-2^{31}~$(2^{31}-1)$	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Position command value (Unit: Command unit).					

Object 6063h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6063h**					
Designation	**Motor Position Feedback**					
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{31}~$(2^{31}-1)$	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Reflect real-time motor absolute position feedback (in encoder units).					

Object 6064h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6064h**					
Designation	**User Position Feedback**					
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{31}~$(2^{31}-1)$	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Real-time absolute motor position feedback (Unit: Command unit). User position feedback 6064h × Gear ratio (6091h) = Motor position feedback 6063h.					

Object 6065h				**PP**	**HM**	**IP**
Index	**6065h**					
Designation	**Excessive Position Deviation Threshold**					
Object structure	VAR	**Data type**	Uint32	**Data range**	0~$(2^{31}-1)$	
Mapping	Y	**Access**	RW	**Factory setting**	3840000	
Function description	Set the excessive position deviation threshold (Unit: Command unit). If the difference between user position command 6062h and user position feedback 6064h exceeds ±6065h, an excessive position deviation fault (ER.d00) occurs.					

	When 6065h is set to 4294967295, the servo does not monitor excessive position deviation.					
Object 6067h				**PP**	**HM**	**IP**
Index	**6067h**					
Designation	**Position Reach Threshold**					
Object structure	VAR	**Data type**	Uint32	**Data range**	$0 \sim (2^{31}-1)$	
Mapping	Y	**Access**	RW	**Factory setting**	100	
Function description	Set the threshold value for position reaching (unit: Command unit). The difference between the user position command 6062h and the actual user position feedback 6064h is within ±6067h, and the position is considered to be reached when the time reaches 6068h, and status word 6041 bit10=1 of in profile position mode.					
Object 6068h				**PP**	**HM**	**IP**
Index	**6068h**					
Designation	**Position Reach Time Window**					
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535	
Mapping	Y	**Access**	RW	**Factory setting**	0	
Function description	Set the time window (unit: 2ms) for judging the validity of the position arrival. The difference between the user position command 6062h and the actual user position feedback 6064h is within ±6067h, and the position is considered to have arrived when the time reaches 6068h, and the status word 6041h bit10=1 in the profile position mode.					

Object 606Bh		**PP**	**PV**	**PT**	**HM**	**IP**	
Index	**606Bh**						
Designation	**User Actual Speed Command**						
Object structure	VAR	**Data type**	Int32	**Data range**	$-2^{31} \sim (2^{31}-1)$		
Mapping	Y	**Access**	RO	**Factory setting**	-		
Function description	Reflect the actual user speed command (unit: Command unit/s). In position-related modes, it reflects the speed command corresponding to the output of the position regulator; In speed-related modes, it reflects the input command of the speed regulator.						
Object 606Ch		**PP**	**PV**	**PT**	**HM**	**IP**	
Index	**606Ch**						
Designation	**User Actual Velocity Feedback**						
Object structure	VAR	**Data type**	Int32	**Data range**	$-2^{31} \sim (2^{31}-1)$		
Mapping	Y	**Access**	RO	**Factory setting**	-		
Function description	reflecting the actual user speed feedback value (unit: Command unit/s).						
Object 606Dh						**PV**	
Index	**606Dh**						
Designation	**Velocity Reach Threshold**						
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535		
Mapping	Y	**Access**	RW	**Factory setting**	100		

Function description	Set the threshold value for speed reaching (unit: 0.1rpm). When the difference between the target speed 60FFh and the actual user speed 606Ch is within ±606Dh and the time reaches 606Eh, the speed is considered to be reached and status word 6041h bit10 = 1 in the profile speed mode. Conversely, status word 6061h bit10 = 0.					
Object 606Eh						**PV**
Index	**606Eh**					
Designation	**Velocity Reach Window Time**					
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535	
Mapping	Y	**Access**	RW	**Factory setting**	0	
Function description	Set the time window (unit: ms) for judging the speed arrival validity. If the difference between the target speed 60FFh and the actual user speed 606Ch is within ±606Dh and the time reaches 606Eh, the speed is considered to be reached, and status word 6041h bit 10 =1 in the profile speed mode. Otherwise, status word 6061h bit 10 =0.					
Object 606Fh						**PV**
Index	**606Fh**					
Designation	**Zero-speed threshold**					
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535	
Mapping	Y	**Access**	RW	**Factory setting**	10	
Function description	Set the time window used to judge whether the user speed is 0 (unit: 2ms). User speed feedback 606Ch within ±606Fh, and the time reaching 6070h set value means that the user speed is 0, at this time the status word 6041h bit12 = 1; either of the two conditions nor met means that the user speed is not 0, at this time the status word 6041h bit12 of = 0.					

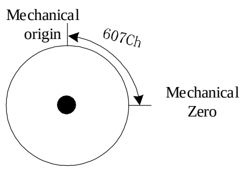
Table 11- 103

Object 6070h						**PV**
Index	**6070h**					
Designation	**Zero-speed Window Time**					
Object structure	VAR	**Data type**	Uint16	**Data range**	0~65535	
Mapping	Y	**Access**	RW	**Factory setting**	0	
Function description	Set the time window used to judge whether the user speed is 0 (unit: 2ms). User speed feedback 606Ch within ±606Fh, and the time reaching 6070h set value means that the user speed is 0, at this time the status word 6041h bit12 = 1; either of the two conditions nor met means that the user speed is not 0, at this time the status word 6041h bit12 of = 0.					
Object 6071h						**PT**
Index	**6071h**					
Designation	**Target torque**					
Object structure	VAR	**Data type**	Int16	**Data range**	-5000~5000	
Mapping	Y	**Access**	RW	**Factory setting**	0	

Function description	For commanding the target value (unit: 0.1%) in profile torque mode and cycle synchronous torque mode.					
Object 6072h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6072_h**					
Designation	**Maximum torque**					
Object structure	VAR	**Data type**	Uint16	**Data range**	-5000~5000	
Mapping	Y	**Access**	RW	**Factory setting**	3000	
Function description	Set the maximum output torque value of the servo (unit: 0.1%).					
Object 6074h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6074_h**					
Designation	**Torque command**					
Object structure	VAR	**Data type**	Uint16	**Data range**	-5000~5000	
Mapping	Y	**Access**	RO	**Factory setting**	-	
Function description	Display the current torque command (unit: 0.1%).					
Object 6075h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6075_h**					
Designation	**Motor rated current**					
Object structure	VAR	**Data type**	Uint 32	**Data range**	Uint 32	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	The rated current (in mA) on the motor nameplate. All current-related parameters are related to this parameter.					
Object 6076h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6076_h**					
Designation	**Motor rated torque**					
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	The rated torque (in mNm) on the motor nameplate. All torque related parameters are related to this parameter.					
Object 6077h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6077_h**					
Designation	**Motor feedback torque**					
Object structure	VAR	**Data type**	Int16	**Data range**	Int16	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Reflect the instantaneous torque output of the servo motor (unit: 0.1%).					
Object 6078h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6078_h**					
Designation	**Instantaneous current output**					
Object structure	VAR	**Data type**	Int16	**Data range**	Int16	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Reflect the instantaneous current output of the servo motor (unit: 0.1%).					

Object 607Ah					**PP**
Index	**607A$_h$**				
Designation	**Target Position**				
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{31}~$(2^{31}-1)$
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	Set the servo target position (unit: Command unit) in profile position mode. When control word 6040h bit 6 is 0, 607Ah is the absolute target position of the current segment; When control word 6040h bit 6 is 1, 607Ah is the target incremental displacement of the current segment.				

Object 607Ch					**HM**
Index	**607C$_h$**				
Designation	**Home Offset**				
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{31}~$(2^{31}-1)$
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	In the position-related control mode, the mechanical zero point deviates from the physical position of the motor origin (unit: Command unit). Mechanical zero point = mechanical home position + 607Ch (home offset). When set to 0, the home point is not offset.				



Object 607Dh		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**607D$_h$**					
Designation	**Software absolute position limit**					
Object structure	ARR	**Data type**	Int32	**Data range**	Int32	
Mapping	Y	**Access**	RW	**Factory setting**	0	
Function description	Set the minimum and maximum value of the software absolute position limit. Minimum absolute position limit = (607D-01h) Maximum absolute position limit = (607D-02h) Software absolute position limit setting: 1.When both (607D-01h) and (607D-02h) are set to the default value, the software limit is invalid. 2.When the minimum absolute position limit (607D-01h) is greater than the maximum absolute position limit (607D-02h), the software internal automatically adjust its value. 3.When the position command or position feedback reaches the software limit value, the servo will take the position limit as the target posion in position mode, and stop when it reaches the position limit, and then prompt the overtravel warning. Input reverse command to exit position exceeding state of motor.					

	4. Absolute position limit is relative to the motor feedback position 6064h (user unit).					
Sub-index	**00_h**					
Designation	**Sub-index Number**					
Object structure	VAR	**Data type**	Uint8	**Data range**	2	
Mapping	Y	**Access**	RO	**Factory setting**	2	
Sub-index	**01_h**					
Designation	**Minimum Software Absolute Position Limit**					
Object structure	VAR	**Data type**	Int32	**Data range**	Int32	
Mapping	Y	**Access**	RW	**Factory setting**	-231	
Sub-index	**02_h**					
Designation	**Maximum Software Absolute Position Limit**					
Object structure	VAR	**Data type**	Int32	**Data range**	Int32	
Mapping	Y	**Access**	RW	**Factory setting**	231	
Object 607Eh		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**$607E_h$**					
Designation	**Command Polarity**					
Object structure	VAR	**Data type**	Uint8	**Data range**	Int8	
Mapping	Y	**Access**	RW	**Factory setting**	0	
Function description	Set the polarity of position command, speed command and torque command. MSB LSB					

7	6	5	4 0
position polarity	speed polarity	torque polarity	NA

Bit7 = 1, standard position mode, reverses the motor as the position command × (-1) .
In profile position mode and synchronous cycle position mode, the position command and target position are reversed.
Bit6 = 1, speed mode, speed command (60FFh) ×(-1), reverse the motor.
Bit5 = 1, torque mode, torque command ×(-1).

Object 607Fh		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**$607F_h$**					
Designation	**Max Profile Velocity**					
Object structure	VAR	**Data type**	Uint32	**Data range**	$0\sim(2^{31}-1)$	
Mapping	Y	**Access**	RW	**Factory setting**	838860800	
Function description	Set the maximum running speed (unit: Command unit/s). The set value is valid when the slave speed command is changed.					

$$\text{Maximum profile speed(rpm)} = \frac{607\text{Fh} \times \frac{6091-01\text{h}}{6091-02\text{h}}}{\text{encoder resolution}} \times 60$$

Note: in various modes, the maximum running speed is limited by the function code Pn318 in addition to the 607Fh limit. The smaller of the two is taken as the limit.

Table 11- 104

Object 6080h				PP	PV	PT	HM	IP
Index	6080_h							
Designation	Max Motor Speed							
Object structure	VAR	Data type	Uint32	Data range	Uint32			
Mapping	Y	Access	RO	Factory setting	Maximum speed limit			
Function description	The maximum allowable running speed of the motor can be obtained from the servo motor manual (unit: rpm).							
Object 6081h								PP
Index	6081_h							
Designation	Position Profile Speed							
Object structure	VAR	Data type	Uint32	Data range	$0\sim(2^{31}\text{-}1)$			
Mapping	Y	Access	RW	Factory setting	8388608			
Function description	The running speed (in command unit/s) reaches the speed of uniform section after the completion of acceleration section in profile position mode. $\text{motor speed(rpm)} = \frac{6081\text{h} \times \frac{6091-01\text{h}}{6091-02\text{h}}}{\text{encoder resolution}} \times 60$							
Object 6083h							PP	PV
Index	6083_h							
Designation	Profile acceleration time							
Object structure	VAR	Data type	Uint32	Data range	$0\sim(2^{31}\text{-}1)$			
Mapping	Y	Access	RW	Factory setting	83886080			
Function description	Set the acceleration (unit: Command unit/s2) during profile position mode and profile speed mode. In position profile mode, the change is effective before this segment command is triggered, and after this segment command is triggered, it is valid when the current segment is finished. In profile speed mode, it takes effect immediately. When the parameter is set to 0, it is forced to 1 internally by the software.							
Object 6084h							PP	PV
Index	6084_h							
Designation	Profile Deceleration Time							
Object structure	VAR	Data type	Uint32	Data range	$0\sim(2^{31}\text{-}1)$			
Mapping	Y	Access	RW	Factory setting	83886080			
Function description	Set the deceleration rate (unit: command unit/s2) during profile position mode and profile speed mode. Effective immediately in profile speed mode. When the parameter is set to 0, it is forced to 1 internally by the software.							
Object 6085h				PP	PV	PT	HM	IP
Index	6085_h							
Designation	Profile Quick Stop Deceleration							
Object structure	VAR	Data type	Uint32	Data range	$0\sim(2^{31}\text{-}1)$			
Mapping	Y	Access	RW	Factory setting	2147483647			

Function description	valid when quick stop 6040h: bit2=0 and when 605Ah(Quick stop mode)=2 , it runs at the speed of deceleration section.				
Object 6086h				**PP**	**PV**
Index	**6086_h**				
Designation	**Motion Running Profile Type**				
Object structure	VAR	**Data type**	Int16	**Data range**	Int16
Mapping	Y	**Access**	RW	**Factory setting**	-
Function description	Profile type of motor position command or speed command				

Object 6087h					**PT**
Index	**6087_h**				
Designation	**Torque ramp time**				
Object structure	VAR	**Data type**	Uint32	**Data range**	0~65535
Mapping	Y	**Access**	RW	**Factory setting**	1000
Function description	Sets the torque command acceleration in profile torque mode, which indicates the torque command increment per second (0.1%/s). The parameter will be forced to convert to 1 when set to 0.				

Table 11- 105

Object 6091h			**PP**	**PV**	**PT**	**HM**	**IP**
Index	**6091_h**						
Designation	Gear Ratio						
Object structure	ARR	**Data type**	Uint32	**Data range**	Uint32		
Mapping	Y	**Access**	RW	**Factory setting**	-		
Function description	The position factor is used to establish the proportional relationship between the user-specified load displacement and the motor displacement: Motor displacement (motor units) = load displacement (user units) x position factor The setting of the position factor is related to the mechanical reduction ratio, the parameters related to the mechanical dimensions and the motor resolution. The calculation is as follows: $\text{position factor} = \frac{\text{motor resolution} \times \text{gear ratio}}{\text{load feeding}}$						
Sub-index	**00_h**						
Designation	**Sub-index Number**						
Object structure	VAR	**Data type**	Uint8	**Data range**	2		
Mapping	Y	**Access**	RO	**Factory setting**	2		
Sub-index	**01_h**						
Designation	**Motor Revolution**						
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32		
Mapping	Y	**Access**	RW	**Factory setting**	1		

Sub-index	**02h**				
Designation	**Shaft Revolution**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	Y	**Access**	RW	**Factory setting**	1

Object6098h					**HM**
Index	**6098h**				
Designation	**Homing Mode**				
Object structure	VAR	**Data type**	Int8	**Data range**	0~35
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	Select the homing method:				

Value	**Description**
1	Home in face of reverse limit switches and Z pulse signals
2	Home in face of forward limit switches and Z-pulse signals
3,4	Home in face of forward home switches and Z pulse signals
5,6	Home in face of reverse home switches and Z pulse signals
7~14	Home in face of home switches and Z pulse signals
15~16	NA
17~30	Home without reference to the Z-pulse signals
31~32	NA
33~34	Home without reference to the Z-pulse signals
35	Current position as zero point

Cautions

●ER.E03 alarm occurs when setting data other than those above.

Table 11- 106

Object 6099h					**HM**
Index	**6099h**				
Designation	**Homing Speed**				
Object structure	ARR	**Data type**	Uint8	**Data range**	Uint32
Mapping	Y	**Access**	RW	**Factory setting**	-
Function description	The 2 speed value settings included in the home mode: 6099-01h search for deceleration point signal speed (unit: command unit/s); 6099-02h search for the home signal speed (unit: command unit/s).				
Sub-index	**00h**				
Designation	**Sub-index Number**				
Object structure	VAR	**Data type**	Uint8	**Data range**	2
Mapping	Y	**Access**	RO	**Factory setting**	2

Sub-index	**01_h**				
Designation	**Search For Deceleration Point Signal Speed**				
Object structure	VAR	**Data type**	Uint32	**Data range**	$0\sim2^{32}$-1
Mapping	Y	**Access**	RW	**Factory setting**	27962027
Function description	This subindex is used to set the search deceleration point signal speed, this speed can be set to a higher value to prevent too long homing time which may result in home return timeout faults.				
Sub-index	**02_h**				
Designation	**Search for home signal speed**				
Object structure	VAR	**Data type**	Uint32	**Data range**	1~500
Mapping	Y	**Access**	RW	**Factory setting**	5592405
Cautions					
	● When home returning, the slave station will decelerate running after finding the deceleration point; ● During deceleration, the slave station shields the change of the home signal, and in order to avoid meeting the home signal during deceleration, the switch position of the deceleration point signal should be set reasonably; such as leaving enough deceleration distance and increasing the acceleration of returning, etc.				

Object609Ah					**HM**
Index	**$609A_h$**				
Designation	**Homing Acceleration/ Homing Deceleration**				
Object structure	VAR	**Data type**	Uint32	**Data range**	$0\sim2^{32}$-1
Mapping	Y	**Access**	RW	**Factory setting**	83886080
Function description	Set the acceleration and deceleration in home return mode (unit: command unit/s2).				

Table 11- 107

Object 60C1h					**IP**
Index	**$60C1_h$**				
Designation	**Interpolation Data Record)**				
Object structure	ARR	**Data type**	Int32	**Data range**	Int32
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	Interpolation mode command parameter setting.				
Sub-index	**00_h**				
Designation	**Sub-index Number**				

Object structure	VAR	**Data type**	Uint8	**Data range**	3
Mapping	N	**Access**	RO	**Factory setting**	3
Sub-index	**01h**				
Designation	**Absolute position command**				
Object structure	VAR	**Data type**	Int32	**Data range**	Int32
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	Interpolation mode absolute position command value, unit: command unit.				
Object 60C2h					**IP**
Index	**60C2h**				
Designation	**Interpolation Cycle**				
Object structure	ARR	**Data type**	Uint8	**Data range**	Uint8
Mapping	Y	**Access**	RW	**Factory setting**	0
Sub-index	**00h**				
Designation	**Sub-index Number**				
Object structure	VAR	**Data type**	Uint8	**Data range**	2
Mapping	N	**Access**	RO	**Factory setting**	2
Function description	Number of subindexes of the Object dictionary for the interpolation cycle.				
Sub-index	01h				
Designation	**Interpolation Cycle Time Constant**				
Object structure	VAR	**Data type**	Uint8	**Data range**	Uint8
Mapping	Y	**Access**	RW	**Factory setting**	1
Function description	The interpolation cycle time unit is given by 60C2_02h. Example: If 60C2_02h is -3, and 60C2_01h is 1, it means the interpolation period currently set is 1ms. Note: The interpolation cycle and the synchronization cycle must be the same.				
Sub-index	**02h**				
Designation	**Interpolation Cycle Time Index**				
Object structure	VAR	**Data type**	Int8	**Data range**	Int8
Mapping	Y	**Access**	RW	**Factory setting**	-3
Function description	Set interpolation period unit. Give -3, the interpolation period unit is ms. Give -4, the interpolation period unit is 0.1ms. Give -2, the interpolation period unit is 10ms.				
Object 60C5h	**PP**	**PV**	**PT**	**HM**	**IP**
Index	**60C5h**				
Designation	**Max Profile Acceleration**				
Object structure	VAR	**Data type**	Uint32	**Data range**	$0\sim(2^{32}-1)$
Mapping	Y	**Access**	RW	**Factory setting**	16777216
Function description	Profile maximum acceleration (unit: Command unit/ s^2)				
Object 60C6h	**PP**	**PV**	**PT**	**HM**	**IP**

Index	**60C6h**					
Designation	**Max Profile Deceleration**					
Object structure	VAR	**Data type**	Uint32	**Data range**	0~(2^{32}-1)	
Mapping	Y	**Access**	RW	**Factory setting**	16777216	
Function description	Profile maximum deceleration (unit: command unit/s^2).					
Object 60E0h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**60E0h**					
Designation	**Forward torque limit value**					
Object structure	VAR	**Data type**	Uint16	**Data range**	Uint16	
Mapping	Y	**Access**	RW	**Factory setting**	3000	
Function description	Limit the maximum value of forward torque (unit: 0.1%).					
Object 60E1h		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**60E1h**					
Designation	**Negative torque limit**					
Object structure	VAR	**Data type**	Uint16	**Data range**	Uint16	
Mapping	Y	**Access**	RW	**Factory setting**	3000	
Function description	Limit the maximum value of negative torque (unit: 0.1%).					
Object60F4h				**PP**	**HM**	**IP**
Index	**60F4h**					
Designation	**User position deviation**					
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{32}~(2^{32}-1)	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Real-time position deviation (unit: user unit).					
Object 60FCh				**PP**	**HM**	**IP**
Index	**60FCh**					
Designation	**Motor position command**					
Object structure	VAR	**Data type**	Int32	**Data range**	-2^{32}~(2^{32}-1)	
Mapping	Y	**Access**	RO	**Factory setting**	0	
Function description	Real-time motor position command (unit: encoder unit). User position command (6062h) × Electronic gear ratio = Motor position command (60FCh)					
Object 60FDh		**PP**	**PV**	**PT**	**HM**	**IP**
Index	**60FDh**					
Designation	**Digital Input**					
Object structure	VAR	**Data type**	Uint32	**Data range**	0~(2^{32}-1)	
Mapping	Y	**Access**	RO	**Factory setting**	0	

Function description	Reflects the current DI terminal logic of the drive, 0 means invalid, 1 means valid The DI signals indicated by each of them are as follows: MSB LSB

31 16	15 4	3	2	1	0
Factory-defined	NA	Undefined	Undefined	Forward overtravel switch	Reverse overtravel switch

Object 60FEh			**PP**	**PV**	**PT**	**HM**	**IP**

Index	**60FE$_h$**				
Designation	**Digital Output**				
Object structure	ARR	**Data type**	Uint32	**Data range**	Uint32
Mapping	Y	**Access**	RO	**Factory setting**	0
Sub-index	**00$_h$**				
Designation	**Sub-index Number**				
Object structure	VAR	**Data type**	Uint8	**Data range**	1
Mapping	N	**Access**	RO	**Factory setting**	1
Sub-index	**01$_h$**				
Designation	**Physical Output**				
Object structure	VAR	**Data type**	Uint32	**Data range**	Uint32
Mapping	Y	**Access**	RO	**Factory setting**	0
Function description	Reflects the drive's current DO terminal logic, 0 indicating invalid and 1 indicating valid. **MSB** **LSB**				

31 16	**15 1**	**0**
Factory-defined	NA	Holding brake output

Object 60FFh	**PV**

Index	**60FF$_h$**				
Designation	**Target Velocity**				
Object structure	VAR	**Data type**	Int16	**Data range**	-2^{32}~$(2^{32}-1)$
Mapping	Y	**Access**	RW	**Factory setting**	0
Function description	User speed command (unit:command unit/s) in profile speed mode.				

Table 11- 108

Object 6502h: Running Mode Supported					
Index	**6502$_h$**				
Designation	**Running Mode Supported**				
Object structure	VAR	**Data type**	Uint16	**Data range**	Uint16
Mapping	Y	**Access**	RO	**Factory setting**	1B$_h$

Function description	Running Mode Supported, 0 means not supported, 1 means supported.

Bit	Description	Value
0	Profile position mode	1
1	NA	0
2	Profile speed mode	1
3	Profile torque mode	1
4	NA	
5	Homing mode	1
6	Interpolation mode	1
7~15	NA	0

11.2.11 Canopen Transmission Halt Code

Table 11- 109

Halt Code	Description
0x05040001	Control commands are invalid (SDO only supports 0x40, 0x2F, 0x2B, 0x23 commands)
0x06010002	Attempting to write a read-only object
0x06020000	The object in the object dictionary does not exist
0x06040041	The object cannot be mapped to PDO
0x06040042	Number and length of mapped objects exceed the PDO length
0x06070010	Inconsistent written length (the length of the object dictionary definition does not match that of the written object)
0x06070012	Inconsistent data type, Inconsistent service parameter length
0x06090011	Sub-index does not exist
0x06090031	Written parameter value is too large
0x06090032	Written parameter value is too small

11.3 Canopen Troubleshooting Information

Table 11- 110

Display	Designation	Error code	Auxiliary code
Er.020	Abnormal user function code parameters and parity	0x6000	0x00000020
Er.021	Abnormal function code parameter formatting	0x6001	0x00000021
Er.022	Abnormal manufacturer parameters and parity	0x6002	0x00000022
Er.023	Abnormal communication between MCU and FPGA	0x6003	0x00000023
Er.030	FPGA backup program	0x6004	0x00000030
Er.040	Abnormal function code parameter setting	0x6005	0x00000040
Er.042	Abnormal combination of parameters	0x6007	0x00000042
Er.050	Inconsistent drive and motor voltage or power difference of more than 4 times	0x6009	0x00000050
Er.0B0	Invalid servo ON command	0x600D	0x000000B0
Er.100	Drive overcurrent (software)	0x600E	0x00000100

Er.101	Drive overcurrent (hardware)	0x600F	0x00000101
Er.320	Regenerative overload	0x6010	0x00000320
Er.400	Overvoltage	0x6012	0x00000400
Er.410	Undervoltage	0x6013	0x00000410
Er.42A	KTY type temperature sensor over-temperature	0x6014	0x0000042A
Er.450	Repeated digital input terminal X function assignment	0x6015	0x00000450
Er.451	Repeated digital input terminal Y function assignment	0x6016	0x00000451
Er.452	Abnormal distribution of analog signal AI in torque mode	0x6017	0x00000452
Er.520	Vibration fault	0x6018	0x00000520
Er.521	Vibration in adjustment-free mode	0x6019	0x00000521
Er.710	Instantaneous drive overload	0x601A	0x00000710
Er.711	Instantaneous motor overload	0x601B	0x00000711
Er.720	Drive continuous overload	0x601C	0x00000720
Er.721	Motor continuous overload	0x601D	0x00000721
Er.730	DB overload	0x601E	0x00000730
Er.7A0	Drive overtemperature	0x6020	0x000007A0
Er.810	Abnormal multi-turn data in absolute encoder	0x6023	0x00000810
Er.820	Abnormal data parity in absolute encoder	0x6024	0x00000820
Er.830	Abnormal battery of absolute encoder	0x6025	0x00000830
Er.840	Abnormal direction at the upper limit of encoder turns	0x6026	0x00000830
Er.860	Over temperature in absolute encoder	0x6028	0x00000860
Er.890	Motor code does not exist	0x6029	0x00000890
Er.8A1	Home return timeout	0x602C	0x000008A1
Er.B31	Abnormal U-phase circuit	0x6034	0x00000B31
Er.B32	Abnormal V-phase circuit	0x6035	0x00000B32
Er.B33	STO input protection	0x6036	0x00000B33
Er.BF0	Abnormal system running	0x6039	0x00000BF0
Er.BF2	MCU data writing to FPGA exception	0x603B	0x00000BF2
Er.BF3	Abnormal pulse command source selection	0x603C	0x00000BF3
Er.C10	Stall detected	0x603E	0x00000C10
Er.C21	Absolute encoder multi-turn overflow	0x6040	0x00000C21
Display	**Designation**	**Error code**	**Auxiliary code**
Er.C80	abnormal incremental encoder frequency division setting	0x6047	0x00000C80
Er.C90	Encoder disconnected	0x6048	0x00000C90
Er.C91	Abnormal encoder acceleration	0x6049	0x00000C91
Er.C92	Incremental encoder Z signal loss	0x604A	0x00000C92
Er.C95	Abnormal encoder UVW signal	0x604B	0x00000C95
Er.D00	Excessive position deviation	0x6050	0x00000D00
Er.D01	Excessive position deviation during servo is ON	0x6051	0x00000D01
Er.D02	Excessive position deviation due to speed limit when servo is ON	0x6052	0x00000D02
Er.D03	Excessive mixing deviation (motor feedback	0x6053	0x00000D03

	position and optical scale deviation are too large)		
Er.D04	Electronic gear ratio setting over limit	0x6054	0x00000D04
Er.E03	Abnormal home return	0x6058	0x00000E03
Er.E05	Running modes not supported by the drive	0x605A	0x00000E05
Er.E20	CAN master disconnected (lifetime factor)	0x6064	0x00000E20
Er.E21	CAN master disconnected (consumer time)	0x6065	0x00000E21

11.4 Homing Mode Description

11.4.1 Mode 1(6098h = 1)

Home signal: Z signal

Deceleration point signal: N-OT (reverse overtravel) signal

(1) The deceleration point signal is OFF during homing

Trajectory: N-OT=0, homing starts at reverse high speed until the rising edge of N-OT, and then decelerates → reverses → forwards at low speed, and stop at the first Z signal after the falling edge of N-OT.

NOTE

NOTE: "H" in the figure is search for high speed of deceleration point =6099-01h, "L" represents the search for low speed of home signal =6099-02h。

$6099-01h=\frac{H\times M}{60}(p/s)$，$6099-02h=\frac{L\times M}{60}(p/s)$，$609Ah=\frac{J\times M}{60}(p/s^2)$。

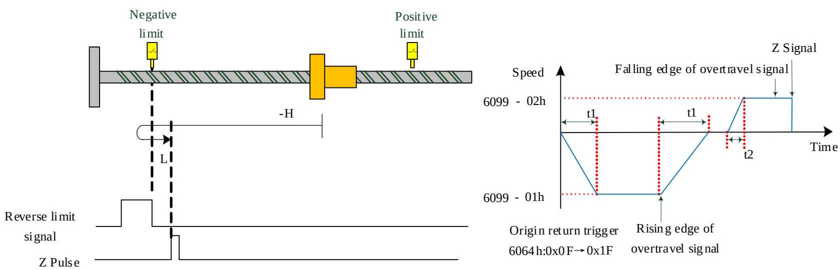
H：homing high speed；L：homing low speed；J：acceleration；M：Determined by encoder bits。

Example: H =100（rpm），L=10（rpm），J=100（rpm/s），M=17 for 17-bit encoder.

$6099-01h=\frac{H\times M}{60}=218453(p/s)$，$6099-02h=\frac{L\times M}{60}=21845(p/s)$，$609Ah=\frac{J\times M}{60}=$

$218453(p/s^2)$。

Figure 11- 30



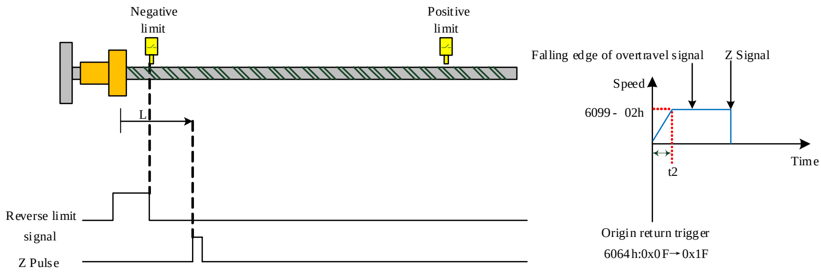
$$t_1=\frac{6099-01h}{609Ah}(s),t_2=\frac{6099-02h}{609Ah}(s)$$

a.6098h=1,initial deceleration point signal =OFF

(2) The deceleration point signal is ON during homing

Trajectory: N-OT=1 when homing, it starts directly at forward low speed, and stops at the first Z signal after the falling edge of N-OT.

Figure 11- 31



$$t_2 = \frac{6099\text{ - }02h}{609Ah}(\mathrm{s})$$

b.6098h=1,initial deceleration point signal =ON

11.4.2 Mode 2(6098h = 2)

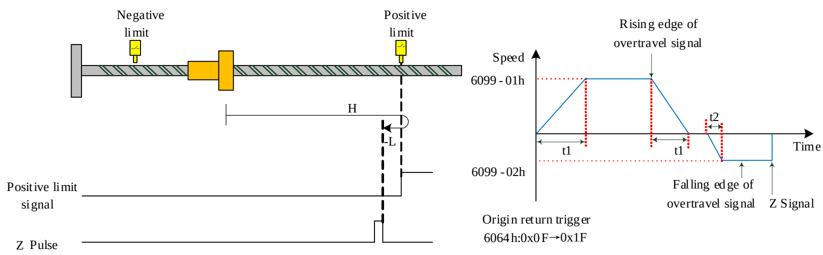
Home signal: Z signal

Deceleration point signal: P-OT (forward overtravel) signal

(1) The deceleration point signal is OFF when homing

Trajectory: P-OT=0 when homing starts at forward high speed until the rising edge of P-OT, and then decelerates→reverses→reverses at low speed, and stops at the first Z signal after the falling edge of P-OT.

Figure 11- 32



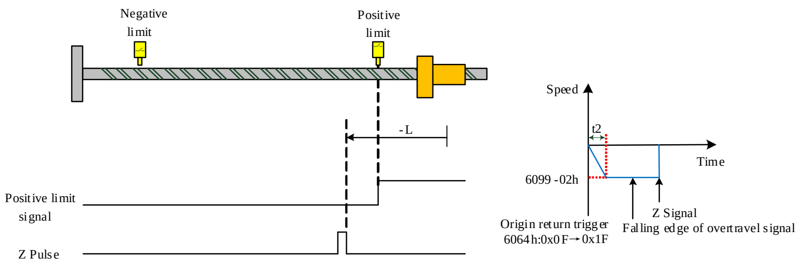
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(\mathrm{s}), t_2 = \frac{6099\text{ - }02h}{609Ah}(\mathrm{s})$$

a.6098h=2,initial deceleration point signal=OFF

(2) The deceleration point signal is ON when homing

P-OT=1 when homing, it starts directly at reverse low speed, and stops at the first Z signal after the falling edge of P-OT.

Figure 11- 33



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=2,initial deceleration point signal=ON

11.4.3 Mode 3(6098h = 3)

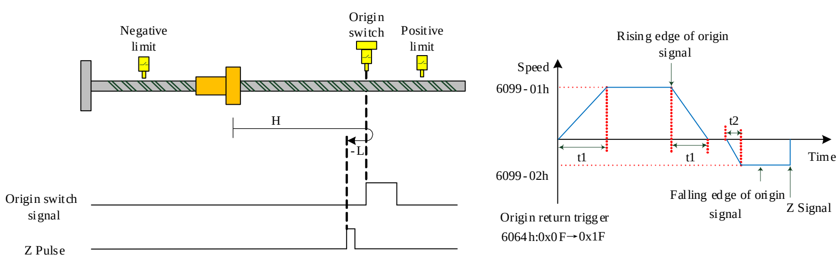
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1) The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW, decelerates → reverses → reverses at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 34



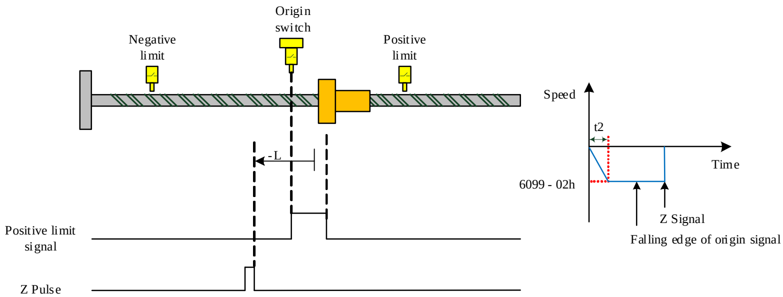
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=3, initial deceleration point signal=OFF

(2) The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts directly at reverse low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 35



$$t_2 = \frac{6099-02h}{609Ah}(s)$$

b.6098h=3,initial deceleration point signal=ON

11.4.4 Mode 4(6098h = 4)

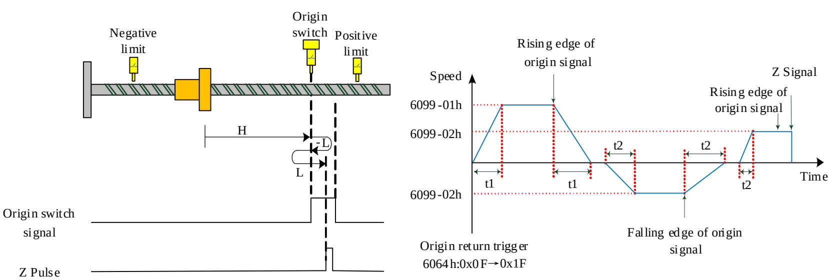
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory:HW=0 when homing starts at forward high speed until the rising edge of HW, and then decelerates → reverses → reverses at low speed until the falling edge of HW, decelerates → reverses → that is, resumes forward low speed running, and stops at the first Z signal after the rising edge of HW.

Figure 11- 36



11 Communication

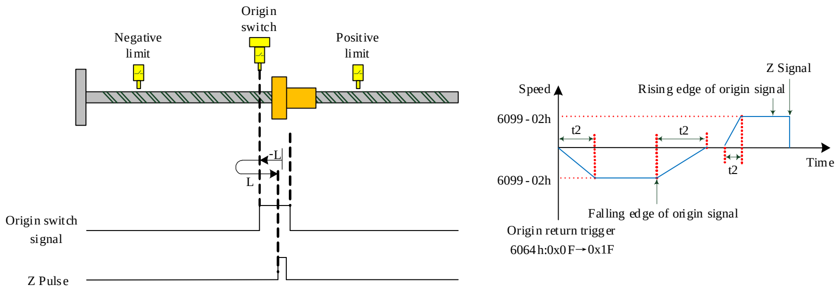
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=4,initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts at reverse low speed until the falling edge of HW, and decelerates → reverses → forwards at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 37



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=4, initial deceleration point signal=ON

11.4.5 Mode 5(6098h = 5)

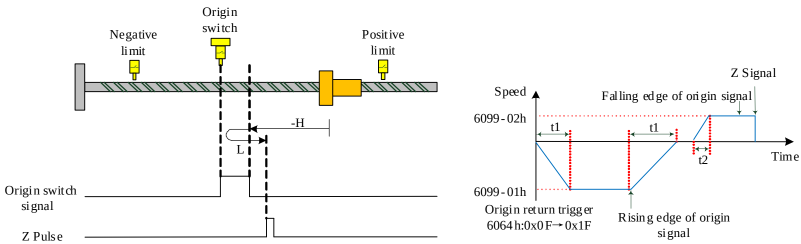
Home signal: Z signal

Deceleration point signal:H HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, and then decelerates → reverses → forwards at low speed until the rising edge of HW, decelerates → reverses →forwards at low speed and stops at the first Z signal after the falling edge of HW.

Figure 11- 38



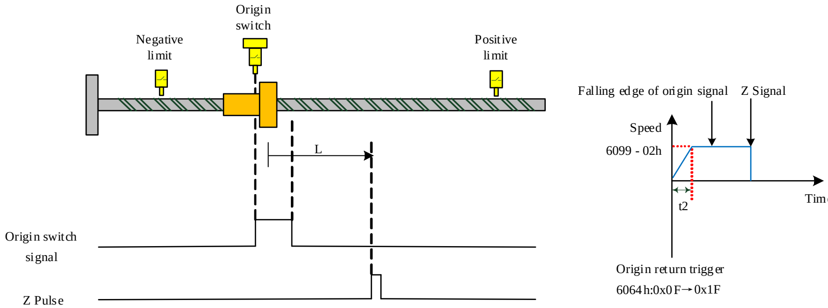
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=5, initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts directly at forward low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 39



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=5, initial deceleration point signal=ON

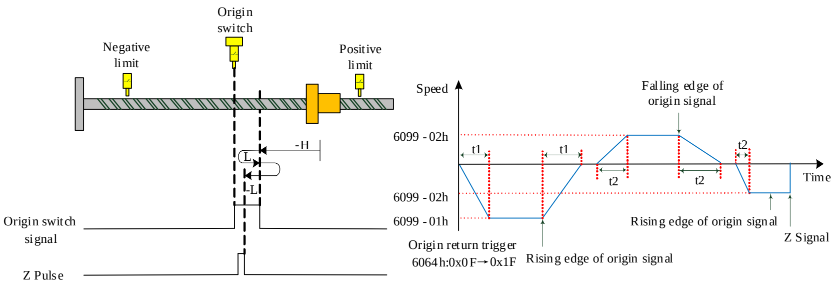
11.4.6 Mode 6(6098h = 6)

Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, and then decelerates → reverses → forwards at low speed until the falling edge of HW, decelerates → reverses →that is resumes reverse low speed running, and stops at the first Z signal after the rising edge of HW.



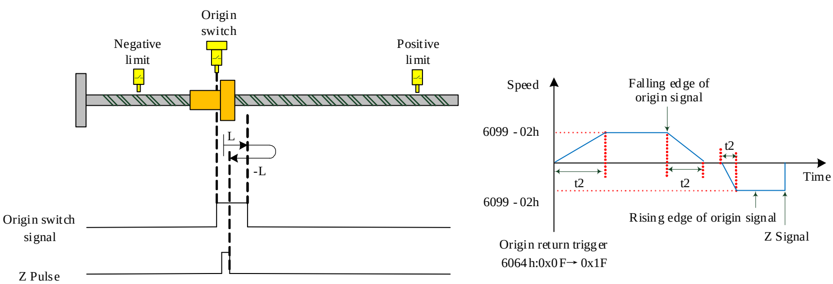
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=6, initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory:HW=1 when homing starts directly at forward low speed until the HW falling edge, and then decelerates → reverses → reverses at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 40



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=6, initial deceleration point signal=ON

11.4.7 Mode 7(6098h = 7)

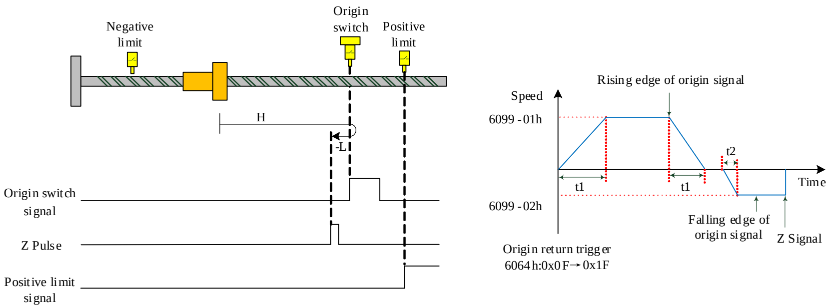
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch in-between, and then decelerates → reverses → reverses at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 41



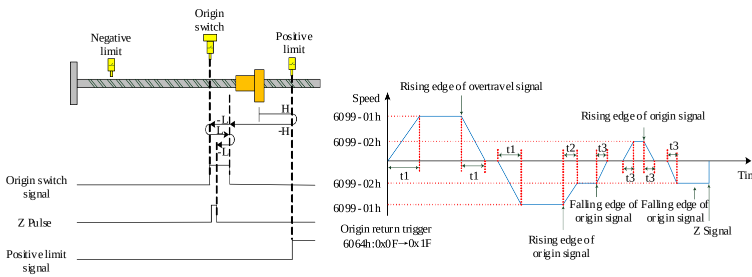
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=7,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing without forward limit signal

Trajectory: HW=0 when homing starts at forward high speed, and if there is a limit switch, reverses automatically at high speed until the rising edge of HW, and then decelerates and goes on reserse running at low speed until the falling edge of HW, decelerates again and reverses, goes on forward running at low speed until the HW rising edge, decelerates and reverses running until it stops at the first Z signal after the falling edge of HW.

Figure 11- 42



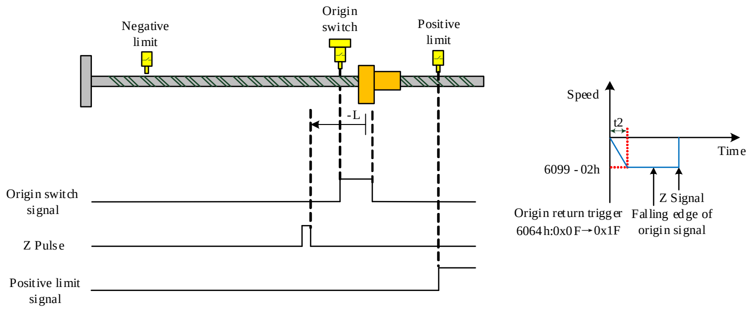
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=7,initial deceleration point signal=OFF without forward limit signal

(3)The deceleration point signal is ON during homing without forward limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed, and stops at the first Z signal after the falling edge of HW

Figure 11- 43



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=7, initial deceleration point signal=ON without forward limit signal

11.4.8 Mode 8(6098h = 8)

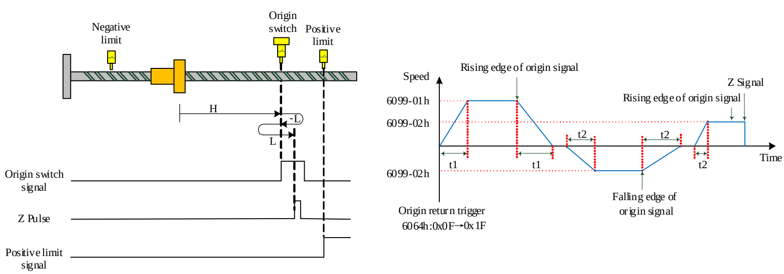
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch in-between, and then decelerates → reverses → reverses at low speed until the falling edge of HW, and then reverses→forwards at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 44



$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

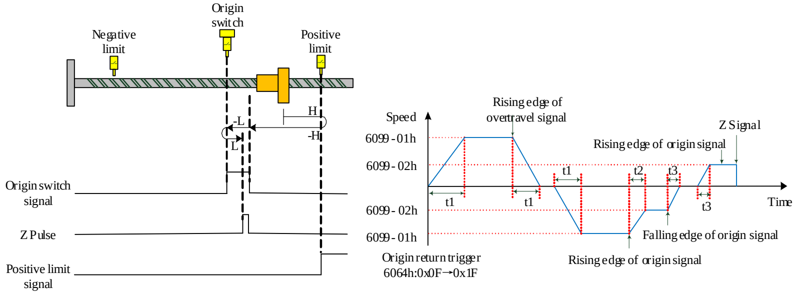
a.6098h=8, initial deceleration point signal=OFF without forward limit signal

11 Communication

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed, and if there is a limit switch, reverses automatically at high speed until the rising edge of HW, and then decelerates and goes on reserse running at low speed until the falling edge of HW, reverses again and goes on forward running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 45



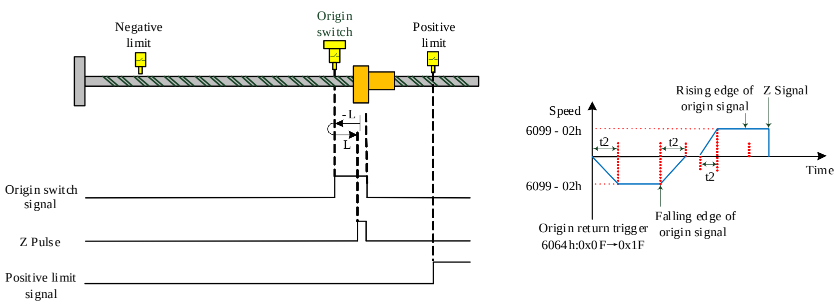
$$t_1=\frac{6099-01h}{609Ah}(s),t_2=\frac{(6099-01h)-(6099-02h)}{609Ah}(s),t_3=\frac{6099-02h}{609Ah}(s)$$

b.6098h=8,initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homing without forward limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed until the HW falling edge, and then reverses and goes on forward running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 46



$$t_2=\frac{6099-02h}{609Ah}(s)$$

c.6098h=8,initial deceleration point signal=ON without forward limit signal

11.4.9 Mode 9(6098h = 9)

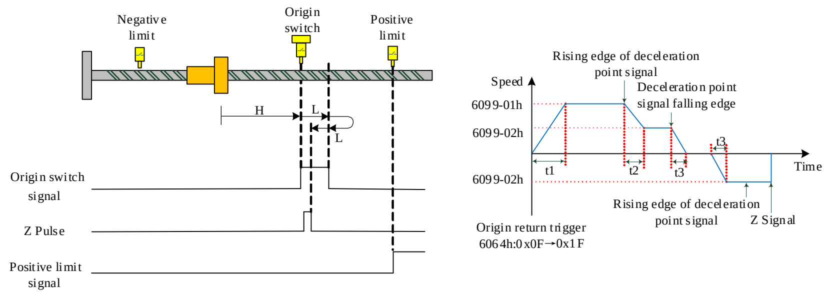
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch in-between, and then decelerates, goes on forward running at low speed until the HW falling edge, reserves and goes on reverse running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 47



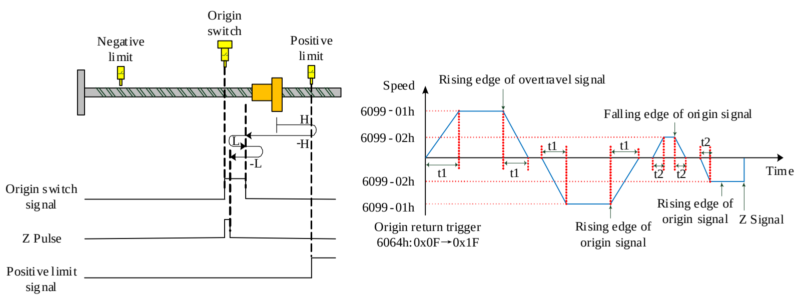
$$t_1 = \frac{6099\text{-}01h}{609Ah}(s), t_2 = \frac{(6099\text{-}01h) - (6099\text{-}02h)}{609Ah}(s), t_3 = \frac{6099\text{-}02h}{609Ah}(s)$$

a.6098h=9,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed, and if there is a limit switch, reverses automatically and goes on reverse running at high speed until the rising edge of HW, and then decelerates and reverses and resumes forward running at low speed until the falling edge of HW, reverses and goes on reverse running at low speed until it stops at the first Z signal after the rising edge of HW.

Figure 11- 48



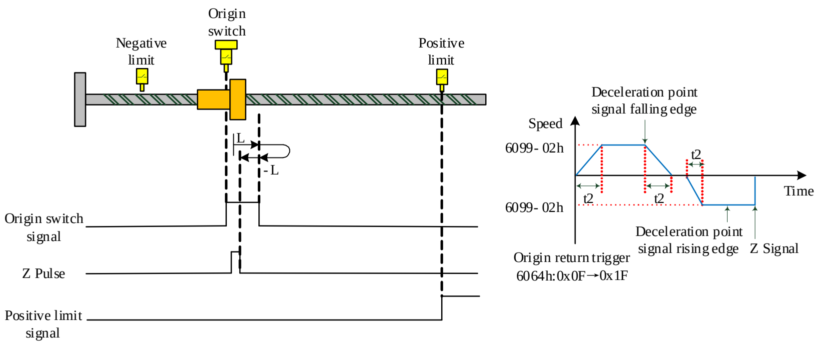
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=9,initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homing without forward limit signal

Trajectory: HW=1 when homing starts directly at forward high speed until the HW falling edge, and then decelerates, reverses and goes on reverse running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 49



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=9,initial deceleration point signal=ON without forward limit signal

11.4.10 Mode 10(6098h =10)

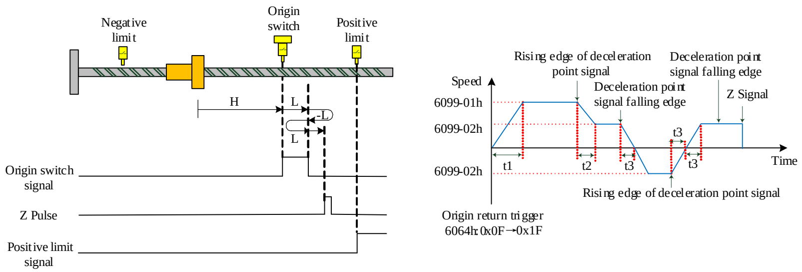
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW, and then decelerates and forwards at low speed until the falling edge of HW, and then decelerates and goes reverse at low speed to the rising edge of HW, decelerates and reverses again, runs forward at low speed and stops at the first Z signal after the falling edge of HW.

Figure 11- 50



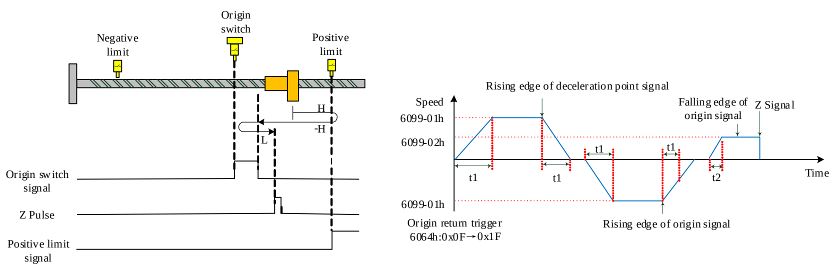
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(s), t_2 = \frac{(6099\text{ - }01h) - (6099\text{ - }02h)}{609Ah}(s), t_3 = \frac{6099\text{ - }02h}{609Ah}(s)$$

a.6098h=10,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory:HW=0 when homing starts at forward high speed, and reverses automatically if there is no limit switch in-between, goes on reverse running at high speed until the rising edge of HW, and then decelerates →reverses→that is resumes forward running at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 51



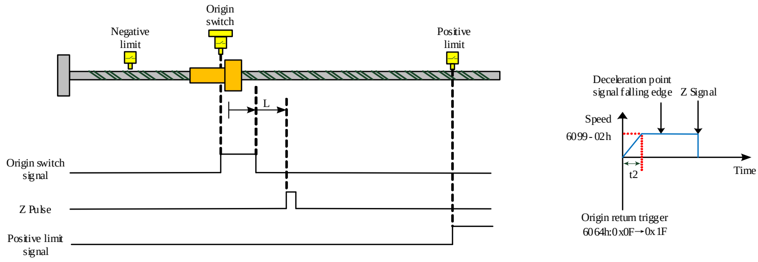
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(s), t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

b.6098h=10,initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homing without forward limit signal

Trajectory: HW=1 when homing starts directly at forward low speed, and stops at the first Z signal after the falling edge of HW

Figure 11- 52



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=10,initial deceleration point signal=ON without forward limit signal

11.4.11 Mode 11(6098h =11)

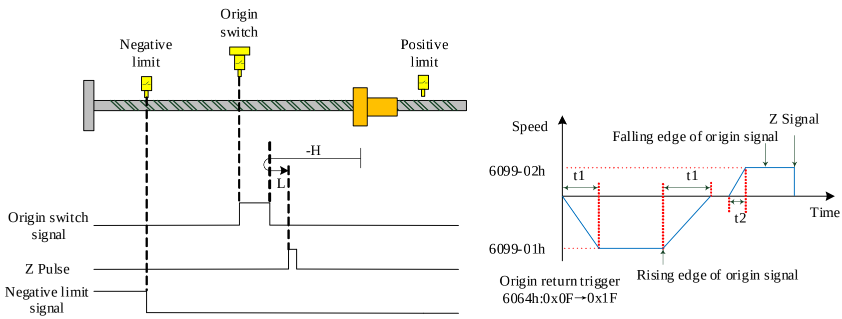
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW if there is no limit switch in-between, and then decelerates, goes on forward running at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 53



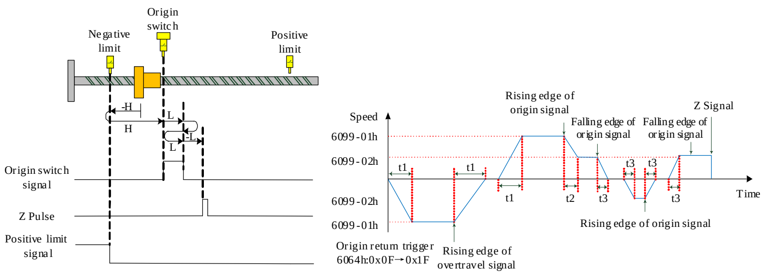
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=11,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, reverses automatically, forwards at high speed until the rising edge of HW, and then decelerates and goes on forward running at low speed until the falling edge of HW, decelerates again and reverses, goes on reverse running at low speed until the HW rising edge, decelerates and reverses to forward at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 54



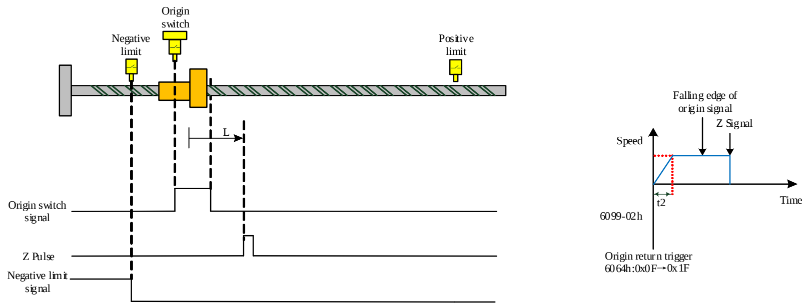
$$t_1=\frac{6099-01h}{609Ah}(s),t_2=\frac{(6099-01h)-(6099-02h)}{609Ah}(s),t_3=\frac{6099-02h}{609Ah}(s)$$

b.6098h=11,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at forward low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 55



$$t_2=\frac{6099-02h}{609Ah}(s)$$

c.6098h=11,initial deceleration point signal=ON without the reverse limit signal

11.4.12 Mode 12(6098h =12)

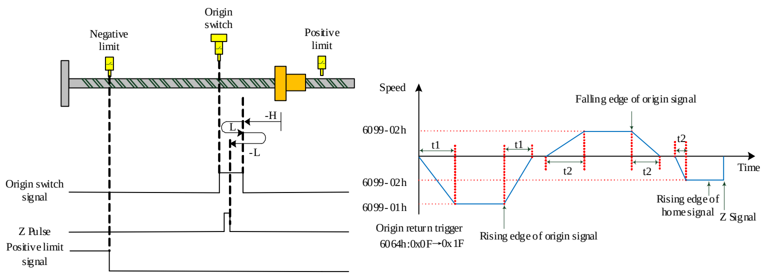
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW if there is no limit switch in-between, and then decelerates → reverses forwards at low speed until the HW falling edge, then reverses runs reversely at low spee and stops at the first Z signal after the rising edge of HW.

Figure 11- 56



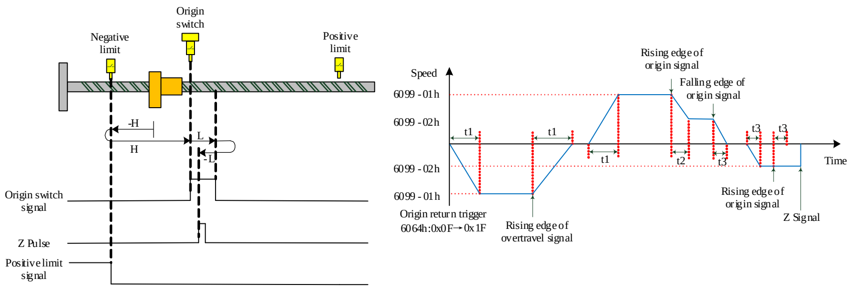
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(s), t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

a.6098h=12,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing, with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, reverses automatically and forwards at high speed until the rising edge of HW, and then decelerates and goes on forward running at low speed until the falling edge of HW, reverses again and goes on reverse running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 57



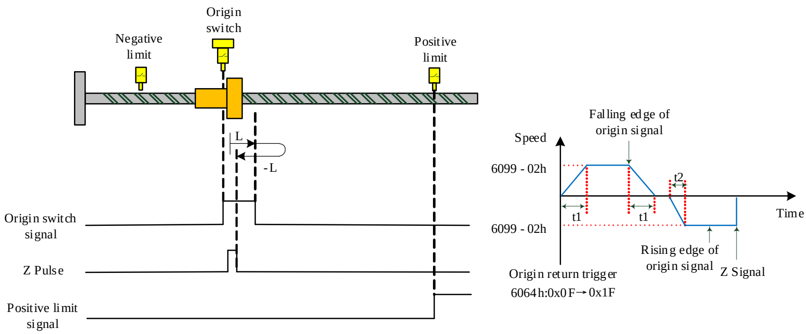
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=12,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts at forward low speed until the falling edge of HW, and then reverses → runs reversely at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 58



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=12,initial deceleration point signal=ON without the reverse limit signal

11.4.13 Mode 13(6098h =13)

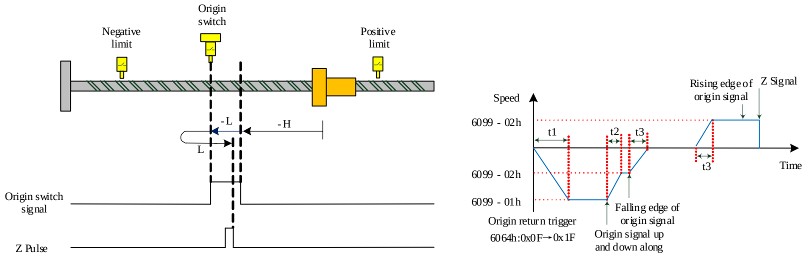
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the falling edge of HW if there is no limit switch in-between, and then reverses and goes on forward running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 59



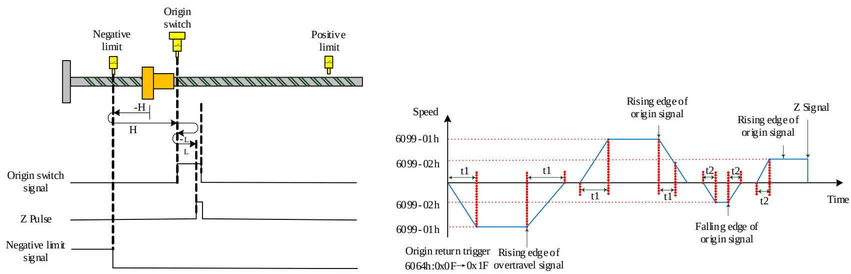
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(\mathrm{s}), t_2 = \frac{(6099\text{ - }01h) - (6099\text{ - }02h)}{609Ah}(\mathrm{s}), t_3 = \frac{6099\text{ - }02h}{609Ah}(\mathrm{s})$$

a.6098h=13,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, reverses automatically and forward at high speed until the rising edge of HW, and then decelerates →reverses →and goes on reserse running at low speed until the falling edge of HW, reverses again →goes on forward running at low speed until it stops at the first Z signal after the rising edge of HW.

Figure 11- 60



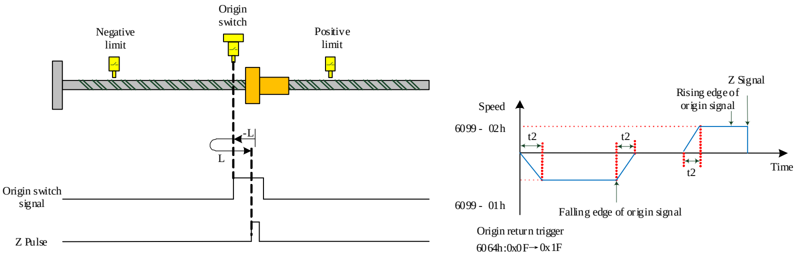
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(\mathrm{s}), t_2 = \frac{6099\text{ - }02h}{609Ah}(\mathrm{s})$$

b.6098h=13,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed until the HW falling edge, and then reverses and goes on forward running at low speed, and stops at the first Z signal after the rising edge of HW.

Figure 11- 61



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=13,initial deceleration point signal=ON without the reverse limit signal

11.4.14 Mode 14(6098h =14)

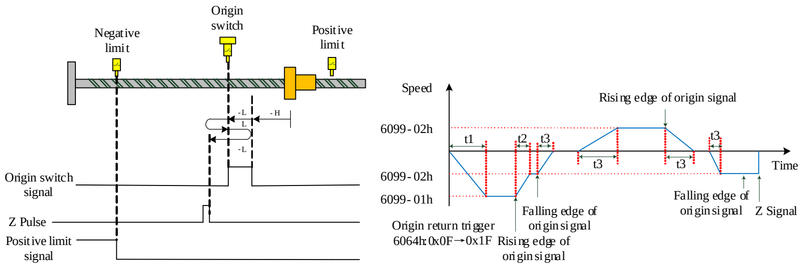
Home signal: Z signal

Deceleration point signal: HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, and then decelerates → runs reversely at low speed until the falling edge of HW, decelerates → reverses →forwards at low speed until the HW rising edge, decelerates → reverses→ runs reversely at low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 62



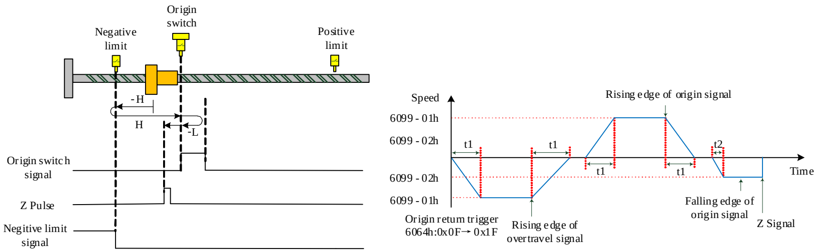
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=14,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, reverses automatically, forwards at high speed until the rising edge of HW, and then decelerates → reverses → and runs reversely at low speed until the falling edge of HW, and stops at the first Z signal.

Figure 11- 63



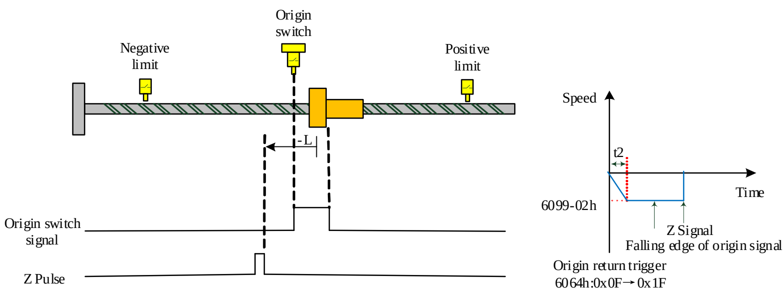
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=14,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed, and stops at the first Z signal after the falling edge of HW.

Figure 11- 64



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=14,initial deceleration point signal=ON without the reverse limit signal

11.4.15 Mode 15(6098h =15)、16(6098h =16)

These two modes of zero return are not defined in the standard 402 protocol.

11.4.16 Mode 17(6098h = 17)

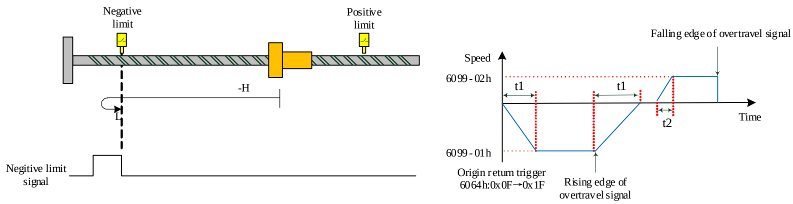
Home signal:N-OT signal (reverse overtravel) falling edge

Deceleration point signal:N-OT signal (reverse overtravel)

(1)The deceleration point signal is OFF during homing

Trajectory:N-OT=0 when homing starts at reverse high speed until N-OT rising edge, and then decelerates → reverses → forwards at low speed until it stops immediately at the N-OT falling edge.

Figure 11- 65



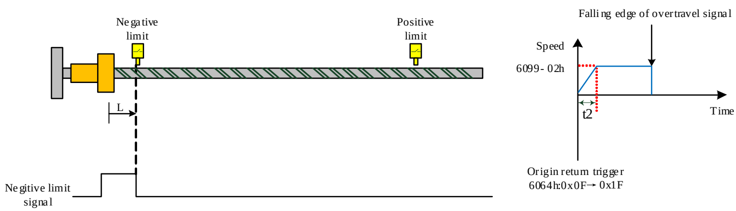
$$t_1 = \frac{6099-01h}{609Ah}(s), t_2 = \frac{6099-02h}{609Ah}(s)$$

a.6098h=17, initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory:N-OT=1 when homing starts directly at forward low speed until it stops immediately at the N-OT falling edge.

Figure 11- 66



$$t_2 = \frac{6099-02h}{609Ah}(s)$$

b.6098h=17, initial deceleration point signal=ON

11.4.17 Mode 18(6098h = 18)

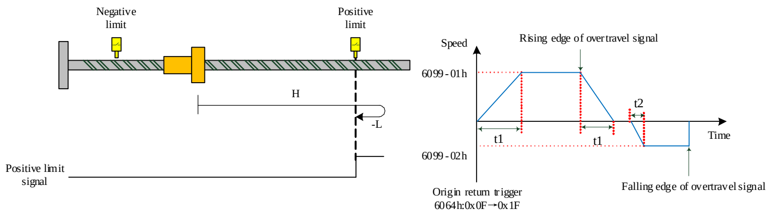
Home signal:P-OT signal (forward overtravel) falling edge

Deceleration point signal:P-OT signal (forward overtravel)

(1)The deceleration point signal is OFF during homing

Trajectory: P-OT=0 when homing starts at forward high speed until P-OT rising edge, and then decelerates → reverses → runs reversly at low speed until it stops immediately at the P-OT falling edge.

Figure 11- 67



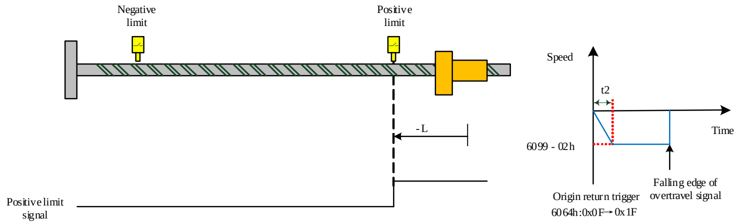
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=18,initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: P-OT=1 when homing starts directly at reverse low speed until it stops immediately at the P-OT falling edge.

Figure 11- 68



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=18, initial deceleration point signal=ON

11.4.18 Mode 19(6098h = 19)

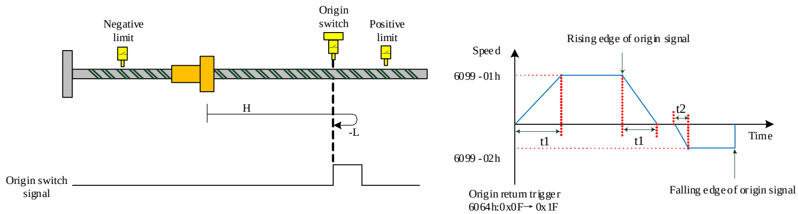
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW, decelerates → reverses →runs reversely at low speed, and stops at the falling edge of HW.

Figure 11- 69



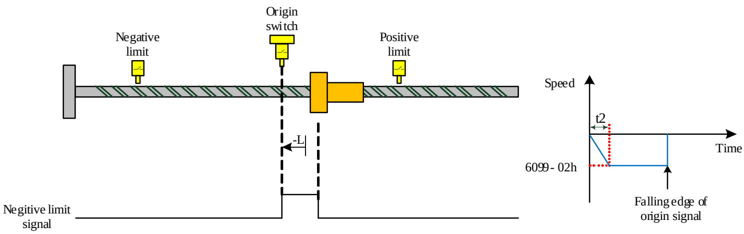
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(s), t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

a.6098h=19,initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts directly at reverse low speed until it stops immediately at the HW falling edge.

Figure 11- 70



$$t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

b.6098h=19, initial deceleration point signal=ON

11.4.19 Mode 20(6098h = 20)

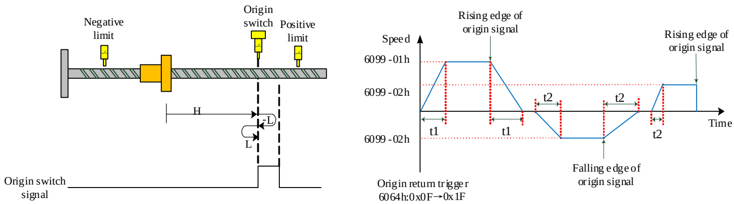
Home signal:HW (home switch) signal rising edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory:HW=0 when homing starts at forward high speed until the rising edge of HW, decelerates → reverses →runs reversely at low speed until the falling edge of HW, and decelerates → reverses→ resumes forward low speed running and it stops immediately at the HW rising edge.

Figure 11- 71



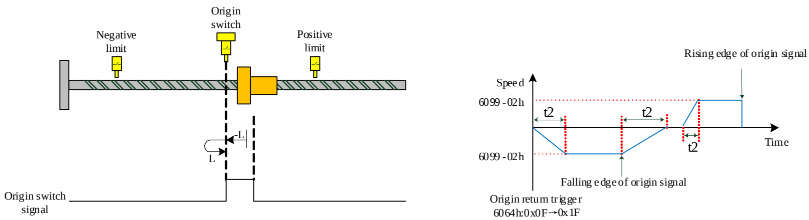
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=20,initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts directly at reverse low speed until the HW falling edge, and then decelerates → reverses→forwards at low speed and it stops immediately at the HW rising edge.

Figure 11- 72



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=20, initial deceleration point signal=ON

11.4.20 Mode 21(6098h = 21)

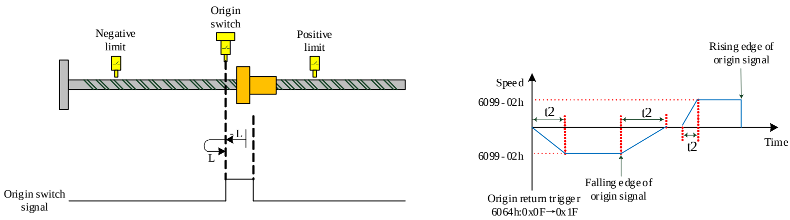
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, decelerates → reverses → forwards at low speed and it stops immediately at the HW falling edge.

Figure 11- 73



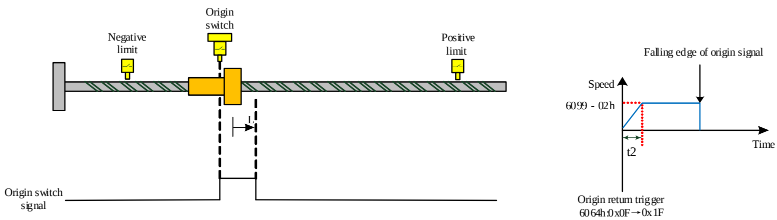
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=21, initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts directly at forward low speed, and it stops immediately at the HW falling edge.

Figure 11- 74



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=21,initial deceleration point signal=ON

11.4.21 Mode 22(6098h = 22)

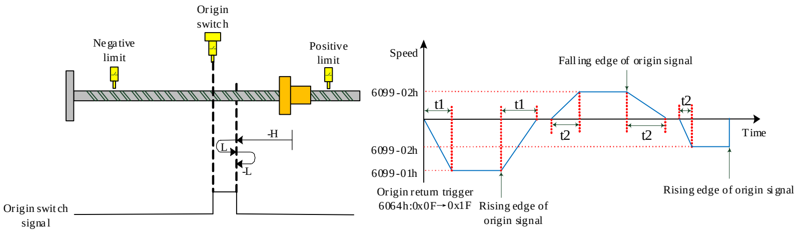
Home signal:HW (home switch) signal rising edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, decelerates → reverses → forwards at low speed until the HW falling edge, decelerates → reverses → resumes reverse rumming at low speed and it stops immediately at the HW rising edge.

Figure 11- 75



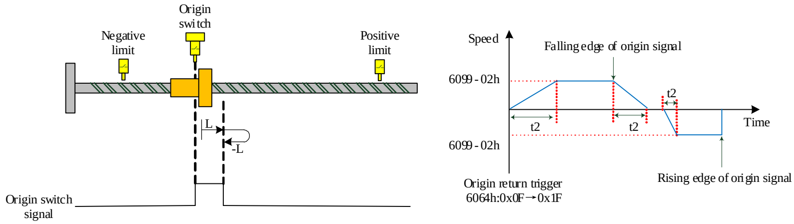
$$t_1 = \frac{6099\text{ - }01h}{609Ah}(s), t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

a.6098h=22,initial deceleration point signal=OFF

(2)The deceleration point signal is ON during homing

Trajectory: HW=1 when homing starts at forward high speed until the falling edge of HW, decelerates → reverses → runs reversely at low speed, and it stops immediately at the HW rising edge.

Figure 11- 76



$$t_2 = \frac{6099\text{ - }02h}{609Ah}(s)$$

b.6098h=22,initial deceleration point signal=ON

11.4.22 Mode 23(6098h = 23)

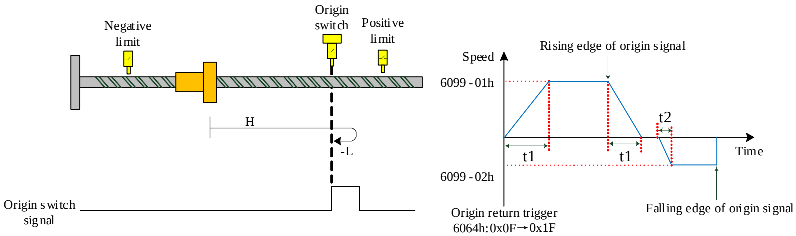
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch, and then decelerates → reverses → runs reversely at low speed and it stops immediately at the HW falling edge.

Figure 11- 77



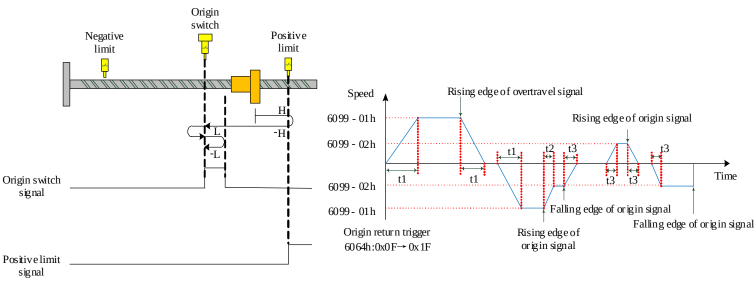
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=23,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed if there is a limit switch, decelerates → reverses → runs reversely at high speed until the rising edge of HW, decelerates and runs reversely at low speed until the falling edge of HW, decelerates → reverses → forwards at low speed until the rising edge of HW, decelerates and runs reversely at low speed, and then stops immediately at the HW falling edge.

Figure 11- 78



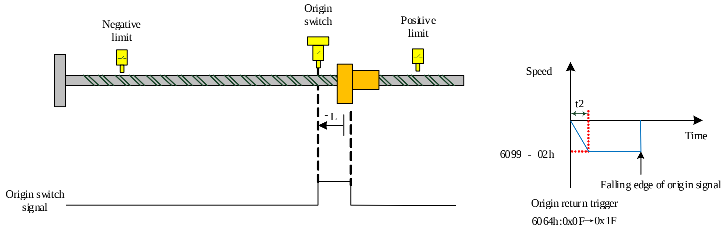
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=23,initial deceleration point signal=OFF without forward limit signal

(3)The deceleration point signal is ON during homingwithout forward limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed, and it stops immediately at the HW falling edge.

Figure 11- 79



$$t_2 = \frac{6099 - 02h}{609Ah}(\mathrm{s})$$

c.6098h=23, initial deceleration point signal=ON without forward limit signal

11.4.23 Mode 24(6098h = 24)

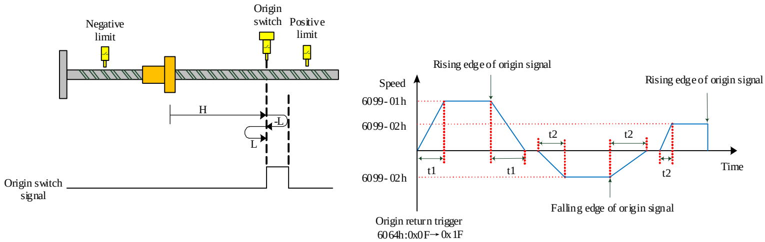
Home signal:HW (home switch) signal rising edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch, decelerates → reverses →runs reversely at low speed until the falling edge of HW, and reverses→forwards at low speed and it stops immediately at the HW rising edge.

Figure 11- 80



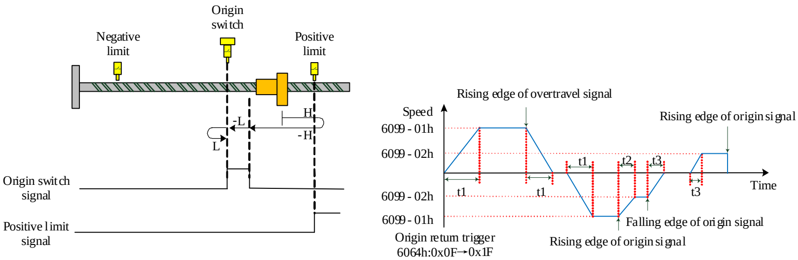
$$t_1 = \frac{6099 - 01h}{609Ah}(\mathrm{s}), t_2 = \frac{6099 - 02h}{609Ah}(\mathrm{s})$$

a.6098h=24,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed if there is a limit switch, decelerates → reverses → runs reversely at high speed until the rising edge of HW, decelerates and runs reversely at low speed until the falling edge of HW, reverses → forwards at low speed until the rising edge of HW and then stops immediately.

Figure 11- 81



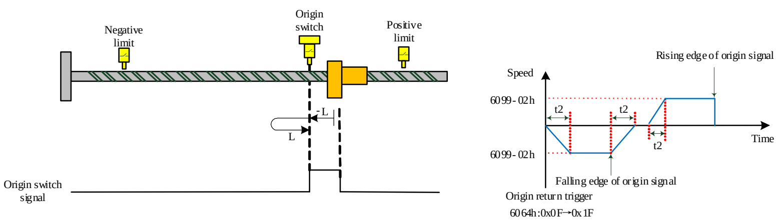
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=24, initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homingwithout forward limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed until the falling edge of HW, reverses and forwards at low speed, , and stops immediately at the rising edge of HW.

Figure 11- 82



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=24,initial deceleration point signal=ON without forward limit signal

11.4.24 Mode 25(6098h = 25)

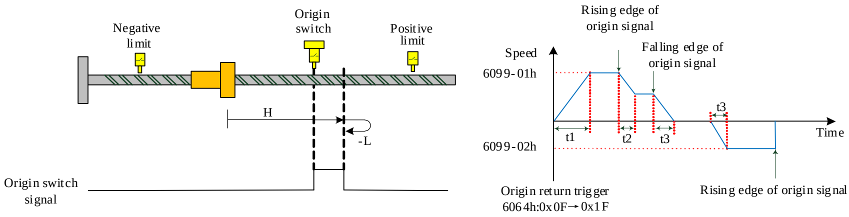
Home signal: HW (home switch) signal rising edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW if there is no limit switch, decelerates →forwards at low speed until the falling edge of HW, and reverses→runs reversely at low speed and it stops immediately at the HW rising edge.

Figure 11- 83



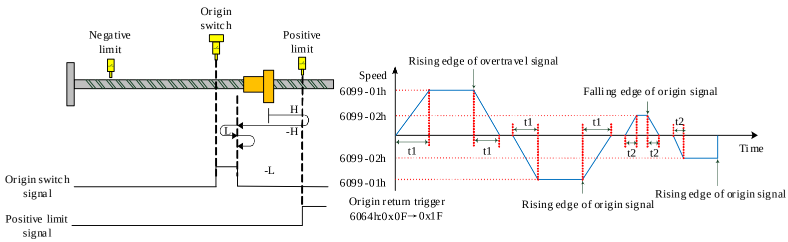
$$t_1=\frac{6099-01h}{609Ah}(s),t_2=\frac{(6099-01h)-(6099-02h)}{609Ah}(s),t_3=\frac{6099-02h}{609Ah}(s)$$

a.6098h=25, initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed, and if there is a limit switch, reverses automatically and goes on reverse running at high speed until the rising edge of HW, and then decelerates and reverses and resumes forward running at low speed until the falling edge of HW, reverses and goes on reverse running at low speed until it stops at the rising edge of HW.

Figure 11- 84



$$t_1=\frac{6099-01h}{609Ah}(s),t_2=\frac{6099-02h}{609Ah}(s)$$

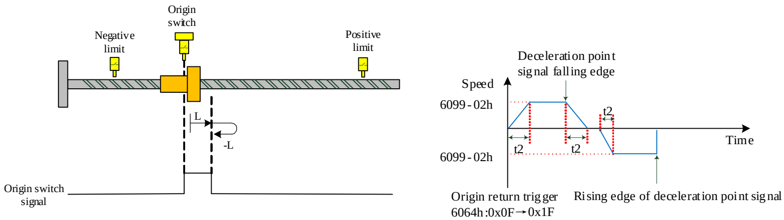
b.6098h=25,initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homingwithout forward limit signal

Trajectory: HW=1 when homing starts at forward high speed until the falling edge of HW, decelerates → reverses → runs reversely at low speed, and it stops immediately at the HW rising edge.

11 Communication

Figure 11- 85



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=25,initial deceleration point signal=ON without forward limit signal

11.4.25 Mode 26(6098h =26)

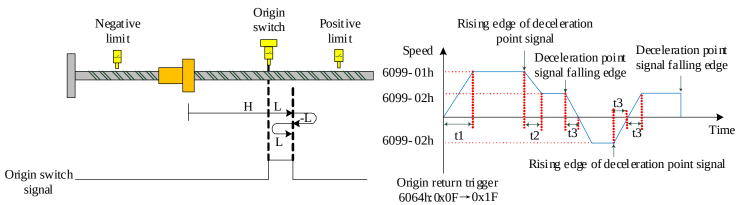
Home signal: Z signal

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homingwithout forward limit signal

Trajectory: HW=0 when homing starts at forward high speed until the rising edge of HW, decelerates →forwards at low speed until the falling edge of HW, and decelerates → reverses→ runs reversely at low speed until the rising edge of HW, decelerates → reverses→ reseumes forward low speed running until the HW falling edge, and it stops immediately.

Figure 11- 86



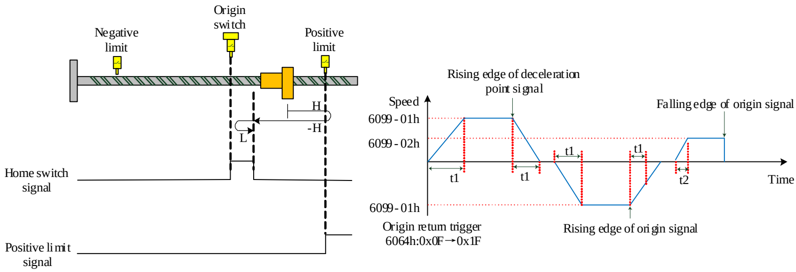
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=26,initial deceleration point signal=OFF without forward limit signal

(2)The deceleration point signal is OFF during homing with forward limit signal

Trajectory: HW=0 when homing starts at forward high speed, and if there is a limit switch, reverses automatically and goes on reverse running at high speed until the rising edge of HW, and then decelerates and reverses and resumes forward running at low speed until the falling edge of HW, and then it stops immdiately.

Figure 11- 87



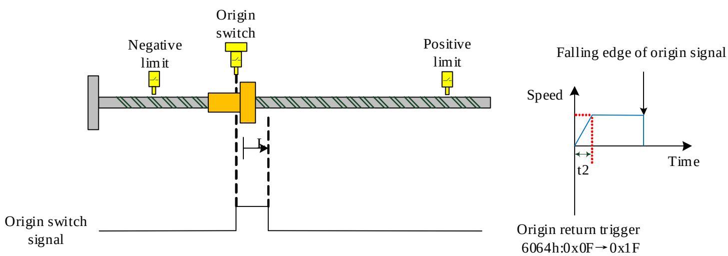
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=26,initial deceleration point signal=OFF with forward limit signal

(3)The deceleration point signal is ON during homingwithout forward limit signal

Trajectory: HW=1 when homing starts directly at forward low speed, and it stops immediately at the HW falling edge.

Figure 11- 88



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=26,initial deceleration point signal=ON without forward limit signal

11.4.26 Mode 27(6098h =27)

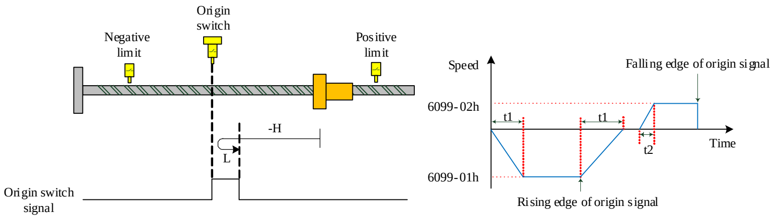
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW if there is no limit switch, decelerates→ reverses → forwards at low speed until the HW falling edge, and it stops immediately.

Figure 11- 89



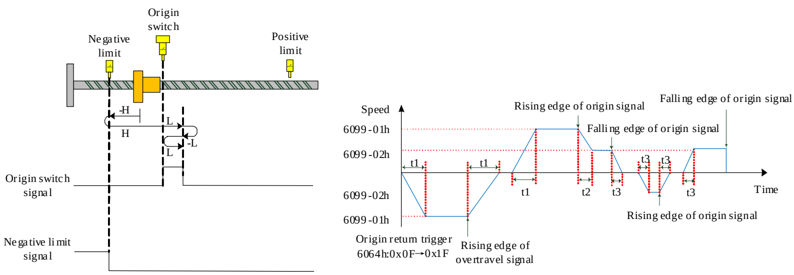
$$t_1 = \frac{6099-01h}{609Ah}(s), t_2 = \frac{6099-02h}{609Ah}(s)$$

a.6098h=27,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, if there is a limit switch, decelerates→ reverses → forwards at high speed until the HW rising edge, decelerates→ forwards at low speed until the HW falling edge, decelerates→ reverses → runs reversely at low speed until the HW rising edge, decelerates→ reverses and forwards at low speed until the HW falling edge, and it stops immediately.

Figure 11- 90



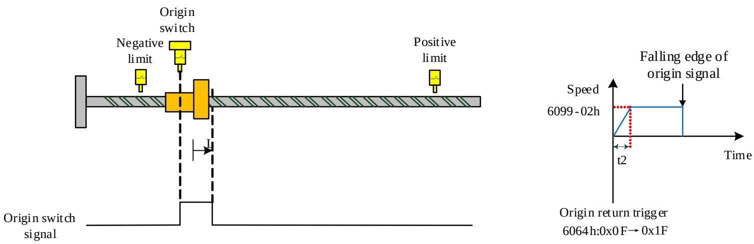
$$t_1 = \frac{6099-01h}{609Ah}(s), t_2 = \frac{(6099-01h)-(6099-02h)}{609Ah}(s), t_3 = \frac{6099-02h}{609Ah}(s)$$

b.6098h=27,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at forward low speed, and stops immediately at the falling edge of HW.

Figure 11- 91



$$t_2 = \frac{6099-02h}{609Ah}(s)$$

c.6098h=27,initial deceleration point signal=ON without the reverse limit signal

11.4.27 Mode 28(6098h =28)

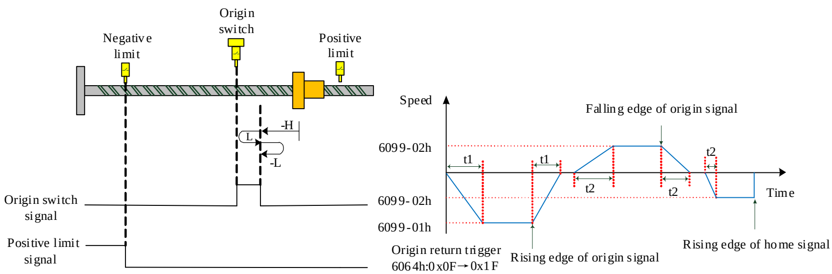
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW if there is no limit switch, decelerates→ reverses → forwards at low speed until the HW falling edge, reverses to run reversely at low speed until the the rising edge of HW, and it stops immediately.

Figure 11- 92



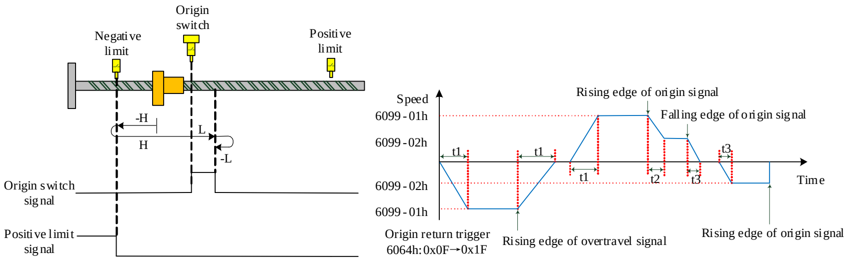
$$t_1 = \frac{6099-01h}{609Ah}(s), t_2 = \frac{6099-02h}{609Ah}(s)$$

a.6098h=28,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, reverses automatically and it turns to high speed forward running until the rising edge of HW, and then decelerates and goes on forward running at low speed until the falling edge of HW, reverses and runs reversely at low speed until the HW rising edge, and then it stops.

Figure 11- 93



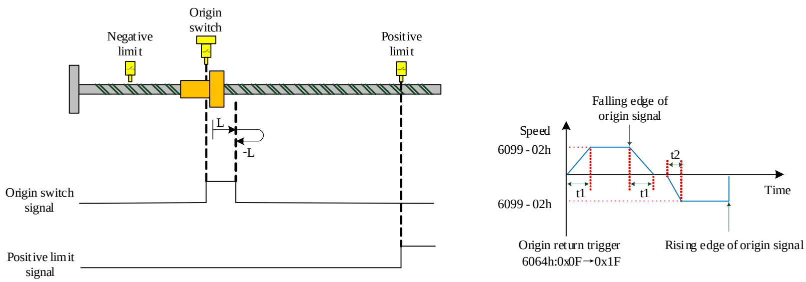
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=28,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at forward low speed until the HW falling edge, and then reverses →runs reversely at low speed, and then stops at the rising edge of HW.

Figure 11- 94



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=28,initial deceleration point signal=ON without the reverse limit signal

11.4.28 Mode 29(6098h =29)

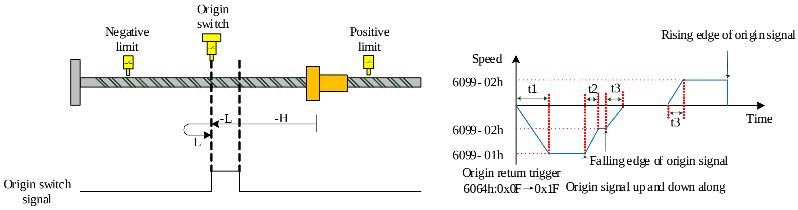
Home signal: HW (home switch) signal rising edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory:HW=0 when homing starts at reverse high speed until the falling edge of HW if there is no limit switch in-between, and then reverses → forwards at low speed until the rising edge of HW, and it stops.

Figure 11- 95



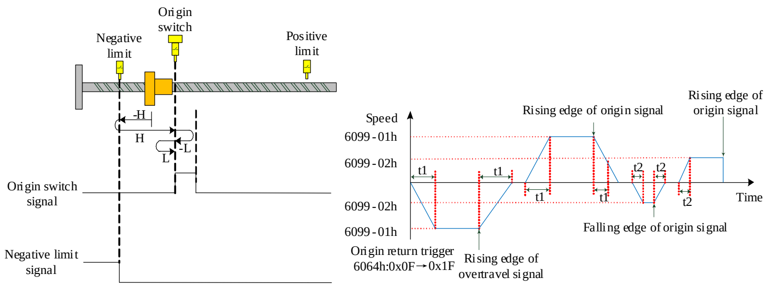
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=29,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, decelerates and reverses automatically and it turns to high speed forward running until the rising edge of HW, and then decelerates and reverses, so it turns into reverse running at low speed until the falling edge of HW, reverses again and forwards at low speed until the HW rising edge, and then it stops.

Figure 11- 96



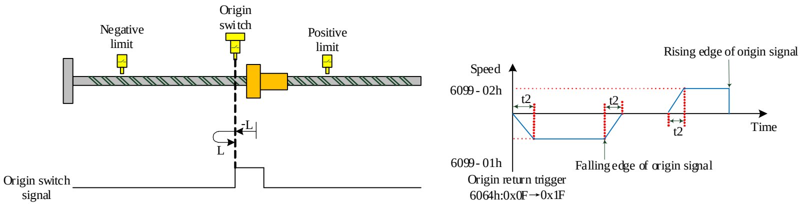
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{6099 - 02h}{609Ah}(s)$$

b.6098h=29,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed until the falling edge of HW, and reverses and forwards at low speed, and it stops immediately at the rising edge of HW.

Figure 11- 97



$$t_2 = \frac{6099 - 02h}{609Ah}(s)$$

c.6098h=29,initial deceleration point signal=ON without the reverse limit signal

11.4.29 Mode 30(6098h =30)

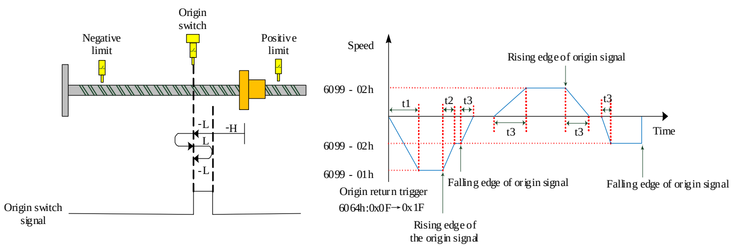
Home signal:HW (home switch) signal falling edge

Deceleration point signal:HW (home switch) signal

(1)The deceleration point signal is OFF during homing without the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed until the rising edge of HW, decelerates → runs reversely at low speed until the HW falling edge, decelerates →reverses→ runs forward at low speed until the HW rising edge, decelerates→reverses→resumes reverse running at low speed until the HW falling edge,and it stops.

Figure 11- 98



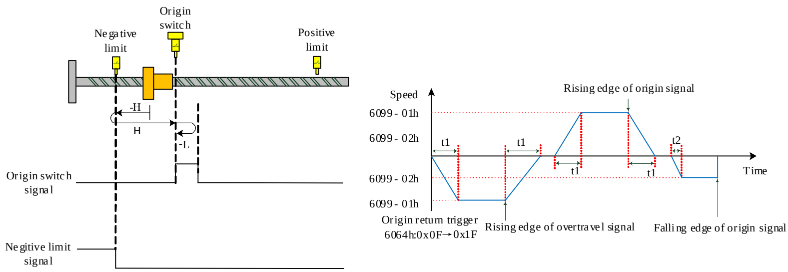
$$t_1 = \frac{6099 - 01h}{609Ah}(s), t_2 = \frac{(6099 - 01h) - (6099 - 02h)}{609Ah}(s), t_3 = \frac{6099 - 02h}{609Ah}(s)$$

a.6098h=30,initial deceleration point signal=OFF with the reverse limit signal

(2)The deceleration point signal is OFF during homing with the reverse limit signal

Trajectory: HW=0 when homing starts at reverse high speed, and if there is a limit switch, decelerates and reverses automatically and it turns to high speed forward running until the rising edge of HW, and then decelerates and reverses, so it turns into reverse running at low speed until the falling edge of HW, and then it stops.

Figure 11- 99



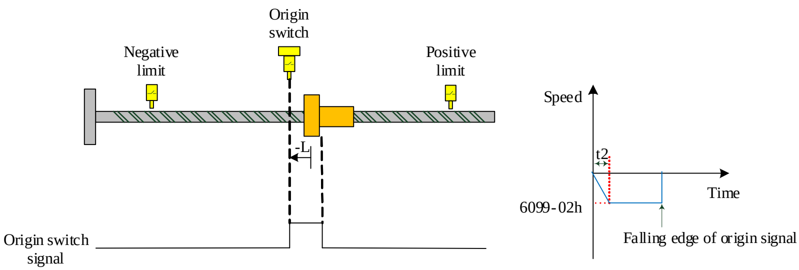
$$t_1 = \frac{6099-01h}{609Ah}(s), t_2 = \frac{6099-02h}{609Ah}(s)$$

b.6098h=30,initial deceleration point signal=OFF with the reverse limit signal

(3)The deceleration point signal is ON during homing without the reverse limit signal

Trajectory: HW=1 when homing starts directly at reverse low speed until the falling edge of HW, and it stops immediately.

Figure 11- 100



$$t_2 = \frac{6099-02h}{609Ah}(s)$$

c.6098h=30,initial deceleration point signal=ON without the reverse limit signal

11.4.30 Mode 31(6098h =31)、32(6098h =32)

These two modes are not defined in the standard 402 protocol.

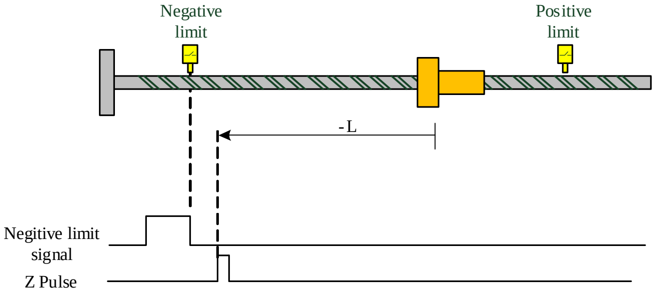
11.4.31 Mode 33(6098h =33)

Home signal: Z signal

Deceleration point signal: none

Trajectory: reverse low speed running until the first Z signal.

Figure 11- 101



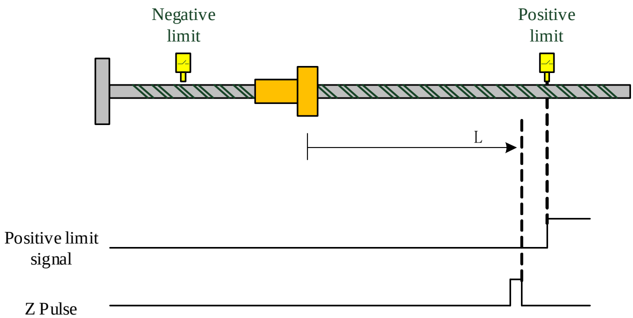
11.4.32 Mode 34(6098h =34)

Home signal: Z signal

Deceleration point signal: none

Trajectory:Forward low speed running until the first Z signal.

Figure 11- 102



11.4.33 Mode 35(6098h =35)

Take the current position as the mechanical home position, and after triggering homing mode, the user position (6064h) = home position offset (607Ch).