SYNTRON 森创

LS series Communication Software Mannual

北京和利时电机技术有限公司
BEIJING HOLLYSYS ELECTRIC TECHNOLOGY CO.,LTD.

Safety Precautions

The content of the manual is based on relevant laws and industrial norms. In case you have any question about the content in this manual during your use of our products, please consult the sales clerks from whom you've bought the products, or contact us via our customer service hotline or by emailing at our email address.

As our products are being upgraded repeatedly, HollySys Electric reserves the right to change any of the products and their specifications and parameters that are contained in this manual without prior notice. We hereby recommend use of the latest edition of this manual to the customers.

Rev. No.: 05/2020

Table of Contents

Chapter I Introduction to the Products

1.1 Introduction to the Products

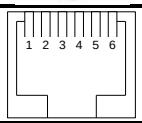
LS-series and DM-series low-voltage servo drivers (hereinafter LS and DM drivers) are new-generation high-performance and high-reliability products developed by HollySys Electric as demanded by the market. The low-voltage DC powered LS drivers are small and easy to form a network; they support CAN, Modbus, pulse and analog control modes. DM drivers are all-in-one type ones, and support CAN and Modbus bus control modes. LS and DM drivers are widely used in AGV, logistics, medical equipment, on-board equipment, defense products, carving machines, surface mounting machines, spray painting machines, textile machines and other fields with very strict requirements for voltage and size.

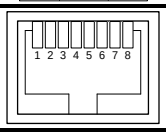
For the users in special industries such as those involving low temperature, communication or mounting application, HollySys Electric can provide specially developed low-voltage servo drivers in 4 series: 4-axis-integrated, 2-axis-integrated, single-axis and all-in-one series. To achieve the best operation performance, please perform installation as instructed by the *LS and DM Series Low-voltage Servo Driver Manual*, and perform programmed control also as instructed by the manual.

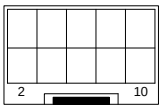
1.2 Communication Interface and Wire Connection

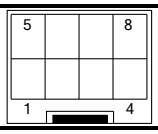
Depending on product models, a communication interface can be one of the following types defined in the table below where sockets are used as examples, and RS232 communication is a customized edition.

	**PS2-8**							
	S/N of pin	1	2	4	5	6	7	8
	Signal definition	CAN_L	232_TX	485_B	GND	CAN_H	485_A	232_RX

	**RJ11**					
	S/N of pin	2	3	4	5	6
	Signal definition	485_A	485_B	CAN_H	CAN_L	GND

	**RJ45**				
	S/N of pin	1	2	4	5
	Signal definition	CAN_H	CAN _L	485_A	485_B

	**Molex 10P**									
	S/N of pin	1	3	5	2	4	6	7	8	9	10
	Signal definition	DC-			DC+			CAN_L	CAN_H	485_B	485_A

	**Molex 8P**							
	S/N of pin	1	2	3	4	5	6	7	8
	Signal definition	DC-		RS485_B	CAN_L	DC+		RS485_A	CAN_H

Through the communication interface, the upper device's system can use Modbus bus or CAN bus to perform parameter reading/writing, and motion control over the LS driver.

Normally, CAN bus employs differential signal transmission with twisted-pair cables as the physical layer, in which two cores (CAN_H and CAN_L) are used for differential signals.

1.3 Communication Setting

Setting of applied work mode is performed by use of parameter Fn000, when it's set as 3, the user can control CANopen communication.

CAN bus' baud rate is to be set by parameter Fn 0F3.

CAN bus' node No. range is 0~255, and is to be set by parameter Fn 0F4.

Parameter No.	Index	Description	Setting Range	Setting Unit	Ex-works Setting
Fn000	2000-00$_h$	Applied work mode	1~3	—	3
Fn0F3	20F3-00$_h$	CAN bus baud rate	0~1000	kHz	500
Fn0F4	20F4-00$_h$	CAN bus node No.	0~255	—	1

Chapter II CANopen Bus Communication

2.1 Power-on Initial Data Packet

Node getting online message

When the driver's power is on, it will send an NMT (Network management) node getting online message to the master station to report its current state.

Standard data frame of a node getting online message for a driver whose CAN node No. is 1 and current state is BOOTUP (started state) is as shown below:

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	14:26:38.016			0x00000701	数据帧	标准帧	0x01	00

Frame ID, which is also COB-ID, is 701h, DLC data length is 1 byte and the data are 00h.

Format of an NMT message is as shown in the table below:

COB-ID	Data Field
700_h+Node-ID	Byte0
	Device state

Device state can be any in the following table, which also gives meanings corresponding to different state data values:

Status Data (decimal)	Meaning
0	BOOTUP (started sate)
4	STOPPED (stopped state)
5	OPERATIONAL (operational state)
127	PRE-OPERATIONAL (pre-operational state)

2.2 Node Guarding Function

Node Guarding

State of a servo node can also be monitored in a poll pattern, which is also called node guarding mode. From the master station node, a standard remote frame (without data) as shown below is sent:

COB-ID
700_h+Node-ID

A standard data frame is sent from the answering servo node, the data length is 1 Byte:

COB-ID	Data Field
700_h+Node-ID	Byte0
	Bit7: Invert Bit6~Bit0: State

The data field includes a triggering bit (bit7), whose value is alternately set as 0 or 1 in the node's answering operations. The part of bit0~bit6 indicates the node's state, detailed description of which is in the list of device state.

Below is an example of node guarding:

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	15:39:43.658	成功		0x00000701	远程帧	标准帧	0x00	
1	接收	15:39:43.664			0x00000701	数据帧	标准帧	0x01	7F
2	发送	15:39:43.668	成功		0x00000701	远程帧	标准帧	0x00	
3	接收	15:39:43.673			0x00000701	数据帧	标准帧	0x01	FF

Data of the 2 answering operations are 7Fh and FFh, whose binary values are respectively (0111 1111)b and (1111 1111)b, bit7's value is inverted, and the servo state corresponding to 7Fh is pre-operational state.

2.3 NMT Node State Switching Command

NMT Node State Switching Command

There are 6 kinds of state for a servo driver which is turned on:

Initializing: initialization of all parameters after the servo is turned on.

Application Reset: reset of the servo's application programs, such as the ones for initialization of all the switching values and analog quantities' output.

Communication Reset: It is reset of the servo's CANopen communication, after which the node can perform CANopen communication.

Pre-operational State: The servo's CANopen communication is ready for operation, and PDO communication cannot be used; the user can only use SDO to set parameters and operate NMT network management.

Operational State: After the servo receives the start up command from the master NMT node, CANopen communication is activated, the user can perform PDO communication control. SDO can also be used for data transmission and parameter configuration.

Stopped State: When the servo has received the stop command from the master NMT node, the servo's PDO communication is stopped, and the user can only use SDO to configure parameters and perform NMT network management operation.

Format of NMT management message:

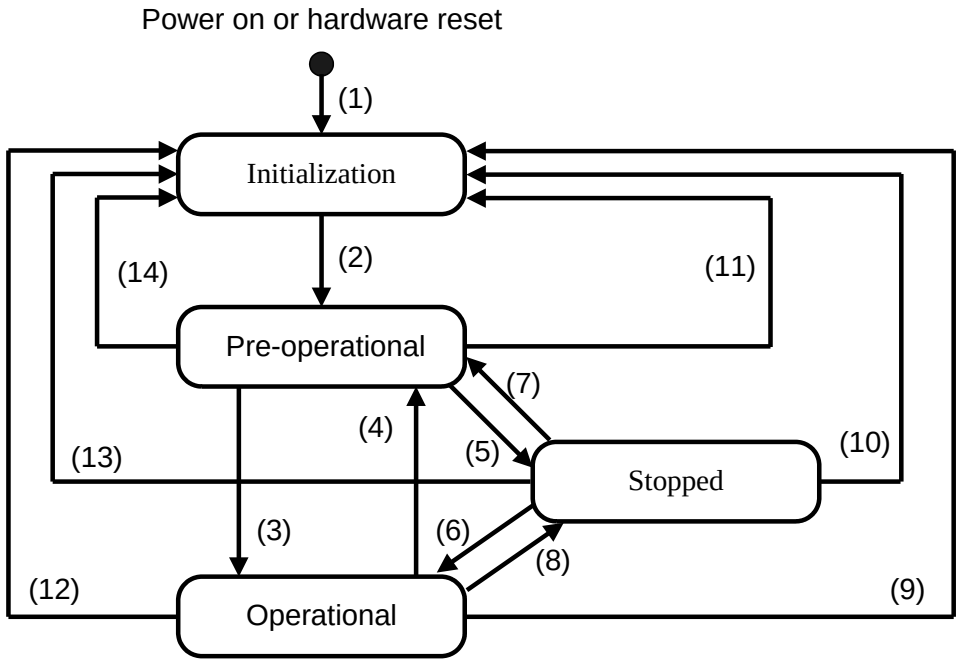
COB-ID	Data Field	
00_h	Byte0	Byte1
	CMD	Node-ID

Byte0 command word is one byte long, and its functions are as shown in the table below:

Command Word	NMT Service	Function
01_h	Start Remote Node	Servo is set to operational state
02_h	Stop Remote Node	Servo is set to stopped state
80_h	Enter Pre-operational State	Servo is set to preoperational state
81_h	Reset Node	Servo is set to application reset state
82_h	Reset Communication	Servo is set to communication reset state

Through NMT Node Guarding service, the master NMT node can check current state of each slave node. This service is especially meaningful when these nodes are not being used for data transmission.

NMT state switching is as shown below:



Transition	Description
(1)	At Power on, the initialization state is entered autonomously
(2)	Once initialization is finished, the Pre-Operational state is automatically entered
(3), (6)	Start Remote Node
(4), (7)	Enter Pre-Operational State
(5), (8)	Stop Remote Node
(9), (10), (11)	Reset Node
(12), (13), (14)	Reset Communication

2.4 Power Failure Emergency Data Packet

Emergency Protocol

When the driver's power fails, it will send to the master station a packet of emergency protocol. The driver's CAN node No. is 1, and standard data frame of the emergency protocol to be reported is as shown below:

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	15:26:25.118			0x00000081	数据帧	标准帧	0x08	02 31 00 00 98 08 00 00

Frame ID, which is also COB-ID, is 81h, DLC data length is 8 byte, and the data are (02 31 00 00 98 08 00 00)h.

When the driver has an internal fault or sends an error alarm, it will trigger the emergency protocol to actively submit the error code. Format of the message is as shown below:

COB-ID	Data Field				
80_h+Node_ID	Byte0~ Byte1	Byte2	Byte3	Byte4~Byte5	Byte6~Byte7
	Emergency error code	Error register	Reserved	Warning	Fault
			Manufacturer-specific error code		

Warning to be sent in case of driver power failure is 0898 h (decimal number 2200), it's an alarm to indicate the power supply's voltage is too low. Details of the manufacturer-specific error codes are in 5.2

2.5 Reading/Writing Parameter (SDO)

Service Data Object (SDO)

An SDO (Service Data Object) is mainly for accessing a node's object dictionary. It can be used to directly configure the driver's parameter reading and writing.

During debugging of each servo driver, the user needs to access the driver's parameters, i.e. to read and write their values, and needs to perform PDO configuration, too. These operations have to be performed through SDO.

2.5.1 Read Values in the Object Dictionary

Read values in the object dictionary:

COB-ID	Data Field							
600_h+Node-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	CMD	Index		Sub-Index	00_h	00_h	00_h	00_h
	40_h	LSB	MSB					

Read a value in the object dictionary in a normal answering operation (a 32-bit value)

COB-ID	Data Field							
580_h+Node-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	CMD	Index		Sub-Index	Data			
	43_h	LSB	MSB		LSB … … … MSB			

Read a value in the object dictionary in a normal answering operation (a 16-bit value)

COB-ID	Data Field							
580_h+Node-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	CMD	Index		Sub-Index	Data			
	$4B_h$	LSB	MSB		LSB	MSB	00_h	00_h

Read a value in the object dictionary in a normal answering operation (a 8-bit value)

COB-ID	Data Field							
580_h+Node-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	CMD	Index		Sub-Index	Data	00_h	00_h	00_h
	$4F_h$	LSB	MSB					

Read a value in the object dictionary in a wrong answering operation

COB-ID	Data Field							
580_h+Node-ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
	CMD	Index		Sub-Index	Error Code			
	80_h	LSB	MSB					

In an example of reading the current speed, the index is 606Ch, subindex is 00h, answer CMD is 43h, unit is puu/s (pulses per second); they are arranged in inverse order, the actual value is 0001870Ah. If the motor encoder is 2500-line, after the line number is multiplied by the frequency multiplying number 4, one turn of the motor needs 10,000 pulses, and the rotation speed is (100106/10000*60)=600.636RPM

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	17:34:20.066	成功		0x00000601	数据帧	标准帧	0x08	40 6C 60 00 00 00 00 00
1	接收	17:34:20.071			0x00000581	数据帧	标准帧	0x08	43 6C 60 00 0A 87 01 00

In the example of reading the current status word, the index is 6041h, subindex is 00h, answer CMD is 4Bh, 0031h.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	11:19:47.934	成功		0x00000601	数据帧	标准帧	0x08	40 41 60 00 00 00 00 00
1	接收	11:19:47.940			0x00000581	数据帧	标准帧	0x08	4B 41 60 00 31 00 00 00

In the example of reading the current work mode, the index is 6061h, subindex is 00h, answer CMD is 4Fh, 03h speed mode.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	17:11:00.176	成功		0x00000601	数据帧	标准帧	0x08	40 61 60 00 00 00 00 00
1	接收	17:11:00.183			0x00000581	数据帧	标准帧	0x08	4F 61 60 00 03 00 00 00

2.5.2 Values Written in the Object Dictionary

Write a 32-bit value in the object dictionary

COB-ID	Data Field							
	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600_h+Node-ID	CMD	Index		Sub-Index	Data			
	23_h	LSB	MSB		LSB … … … MSB			

Write a 16-bit value in the object dictionary

COB-ID	Data Field							
	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600_h+Node-ID	CMD	Index		Sub-Index	Data			
	$2B_h$	LSB	MSB		LSB	MSB	00_h	00_h

Write a 8-bit value in the object dictionary

COB-ID	Data Field							
	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
600_h+Node-ID	CMD	Index		Sub-Index	Data	00_h	00_h	00_h
	$2F_h$	LSB	MSB					

Write a value in the object dictionary in a normal answering operation

COB-ID	Data Field							
	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580_h+Node-ID	CMD	Index		Sub-Index	Reserved			
	60_h	LSB	MSB					

Write a value in the object dictionary in a wrong answering operation

COB-ID	Data Field							
	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
580_h+Node-ID	CMD	Index		Sub-Index	Error Code			
	80_h	LSB	MSB					

2.6 PDO (Process Data Object)

PDO is a kind of procedural data, and is used to send (TPDO) or receive (RPDO) real-time data, such as the motor's real-time speed, position and I/O control. The receiving node doesn't have to respond with a CAN message to confirm. PDO data can be sent from one sender to one or multiple recipients, one

PDO can maximally send 8 bytes of data in one transmission operation.

Type and content of the real-time data carried by a PDO are determined by its mapping structure prescribed in the device's object dictionary. The device in pre-operational state supports dynamic PDO configuration; PDO mapping configuration is performed through a service data object (SDO).

For RPDOs, communication parameters are in 1400h~15FFh, mapping parameters are in 1600h~17FFh, and data are stored in the manufacturer-defined area after 2000h; for TPDOs, communication parameters are in 1800h~19FFh, mapping parameters are in 1A00h~1BFFh, and data are stored in the manufacturer-defined area after 2000h.

Index	Sub-index	Description	Data Type
RPDO communication parameters 1400_h~$15FF_h$ TPDO communication parameters 1800_h~$19FF_h$	00_h	Number of entries, i.e. number of effective entries	Uint8
	01_h	COB-ID: ID of the frame sending/receiving the PDO	Uint32
	02_h	Transmission type 00_h: non-cycling synchronous 01_h: cycling synchronous FC_h: remote synchronous FD_h: remote asynchronous FE_h: asynchronous, manufacturer-specific event FF_h: asynchronous, device sub-protocol-specific event	Uint8
	03_h	In_hibit time, i.e. PDO production prohibition limit time (0.1ms)	Uint16
	05_h	Event timer, i.e. triggering time of the event timer (ms)	Uint16
	06_h	SYNC start value, i.e. synchronous start value	Uint8

Number of entries: it tells how many parameters there are under the index;

COB-ID: ID of the CAN frame corresponding to the PDO's sending or reception;

Transmission type: it tells the form of transmission of the PDO's sending or reception;

Inhibit time: it limits the smallest interval of PDO sending operations to avoid abrupt rise of load on the bus, for example, overly quick numeric value input will cause overly frequent TPDO sending due to change of state, and load on the bus will rise, so a limit time is needed for "filtering"; the unit of limit time is 0.1ms;

Event timer: triggering time of the event timer, it tells the fixed length of time

for regular sending of PDO; if its value is 0, it means the PDO is to be sent at change of event; unit of time is ms;

SYNC start value: PDO for synchronous transmission will be sent only after several synchronous packets are received. The SYNC start value is the number of synchronous packets. For example, when this value is set as 2, it means sending will be performed only when 2 synchronous packets are received.

2.6.1 Example of RPDO Mapping

In the example below, RPDO1 is mapped to asynchronous device sub-protocol-specific mode, only for sending of speed commands; CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	16:56:22.564	成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送	16:56:22.575	成功		0x00000601	数据帧	标准帧	0x08	2F 00 14 02 FF 00 00 00
2	接收	16:56:22.580			0x00000581	数据帧	标准帧	0x08	60 00 14 02 00 00 00 00
3	发送	16:56:22.585	成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 00 00 00 00
4	接收	16:56:22.590			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
5	发送	16:56:22.595	成功		0x00000601	数据帧	标准帧	0x08	23 00 16 01 20 00 FF 60
6	接收	16:56:22.599			0x00000581	数据帧	标准帧	0x08	60 00 16 01 00 00 00 00
7	发送	16:56:22.606	成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 01 00 00 00
8	接收	16:56:22.613			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00

Meanings of the commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1400_h,subindex 02_h, value FF_h, transmission type: asynchronous, device-sub-protocol-specific event;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index 1600_h, subindex 00_h, value 00_h, number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index 1601_h, subindex 01_h, value $60FF0020_h$, mapped to index $60FF_h$, subindex 00_h; the object is 32-bit, and is to specify number of pulses per second.

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index 1600_h, subindex 00_h, value 01_h; number of entries is set as 1.

S/N 8: Setting succeeds, and answering is correct;

2.6.2 Example of TPDO Mapping

In the example below, TPDO1 is mapped to speed feedback and position feedback, current actual speed value and position value are sent between each two synchronous objects, the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	16:56:22.564	成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送	16:56:22.766	成功		0x00000601	数据帧	标准帧	0x08	2F 00 18 02 02 00 00 00
2	接收	16:56:22.769			0x00000581	数据帧	标准帧	0x08	60 00 18 02 00 00 00 00
3	发送	16:56:22.776	成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 00 00 00 00
4	接收	16:56:22.782			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00
5	发送	16:56:22.786	成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 01 20 00 6C 60
6	接收	16:56:22.791			0x00000581	数据帧	标准帧	0x08	60 00 1A 01 00 00 00 00
7	发送	16:56:22.796	成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 02 20 00 63 60
8	接收	16:56:22.800			0x00000581	数据帧	标准帧	0x08	60 00 1A 02 00 00 00 00
9	发送	16:56:22.806	成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 02 00 00 00
10	接收	16:56:22.809			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00

Meanings of the commands sent are:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1800_h, subindex 02_h, value 02_h, frequency of sending: one transmission between each two synchronous objects;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 00_h; number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index $1A01_h$, subindex 01_h, value $606C0020_h$; mapped to index $606C_h$, subindex 00_hh; the object is 32-bit, actual speed;

S/N 6: Setting succeeds, and the answering is correct;

S/N 7: Mapping parameter setting is: index $1A01_h$, subindex 02_h, value 60630020_h; mapped to index 6063_h, subindex 00_h; the object is 32-bit, actual position;

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 02_h; the number of entries is set as 2.

S/N 10: Setting succeeds, and answering s correct;

2.6.3 Example of PDO Commands

As continuation of the preceding example where PDO's relevant setting is completed, this example shows the user can now send RPDO commands to make the motor run under speed mode, and then send synchronous frames to trigger TPDO, to monitor the servo driver's actual speed and actual position.

2.6.3.1 RPDO Commands

In the example, RPDO1 is already mapped to a speed command, and the CANOPEN node No. is 1, It's used for sending of speed command, whose COB-ID is 201h, value is 000186A0h, and unit is puu/s (number of pulses run every second).

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 03 00 00 00
1	接收	00:43:18.199.2			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
2	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
3	接收	00:43:20.722.2			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
4	发送		成功		0x00000201	数据帧	标准帧	0x08	A0 86 01 00 00 00 00 00
5	发送		成功		0x00000201	数据帧	标准帧	0x08	00 00 00 00 00 00 00 00

Send RPDO commands to make the motor run under speed mode; the COB-ID is 201_h.

S/N 0: Work mode setting is: index 6060_h, subindex 00_h, value 03_h; the driver is set to work under speed control mode;

S/N 1: Setting succeeds, and answering is correct;

S/N 2: Motor enabling setting is: index 6040_h, subindex 00_h, value $000F_h$;

S/N 3: Setting succeeds, and answering is correct;

S/N 4: RPDO1 sends a speed command to set speed at 600 RPM (with a 2500-line motor as example), the value is $000186A0_h$, and unit is puu/s (number of pulses run every second).

S/N 5: RPDO1 sends a speed command to set speed at 0, the value is 00000000_h, and unit is puu/s (number of pulses run every second).

2.6.3.2 TPDO Commands

In the example below, TPDO1 is already mapped to feedback of actual current speed value and position value; the condition for triggering feedback events is reception of two synchronous objects; the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	11:21:49.224	成功		0x00000080	数据帧	标准帧	0x00	
1	发送	11:21:49.234	成功		0x00000080	数据帧	标准帧	0x00	
2	接收	11:21:49.238			0x00000181	数据帧	标准帧	0x08	7B 86 01 00 80 49 1B 00

For commands to feed back actual current speed value and position value, COB-ID is 181_h.

S/N 0: Frame 1 of synchronous object;

S/N 1: Frame 2 of synchronous object;

S/N 2: Actual speed value is $0001867B_h$, and actual position value is $001B4980_h$.

2.7 Heartbeat Data Packet

Each CANopen slave station sends heartbeat messages as per the heartbeat production time (ms) written in 1017_h of its object dictionary, and CANopen master station (NMT master station) will perform check as per the heartbeat consumer time written in 1016_h. If no heartbeat message is received from a slave station after a number of periods of heartbeat consumer time, such slave station can be deemed to be already offline or damaged.

2.7.1 Heartbeat Data Packets Reported by Servo Driver

In the example, servo driver's Node_ID is 1, fixed interval between reports of heartbeat data packet is set as 1s, the user uses SDO to write in 1017_h.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 17 10 00 E8 03 00 00
1	接收	02:28:02.303.9			0x00000581	数据帧	标准帧	0x08	60 17 10 00 00 00 00 00
2	接收	02:28:03.303.2			0x00000701	数据帧	标准帧	0x01	7F
3	接收	02:28:04.302.6			0x00000701	数据帧	标准帧	0x01	7F
4	接收	02:28:05.302.0			0x00000701	数据帧	标准帧	0x01	7F
5	接收	02:28:06.301.4			0x00000701	数据帧	标准帧	0x01	7F
6	接收	02:28:07.300.8			0x00000701	数据帧	标准帧	0x01	7F

Configure the servo driver to make it submit heartbeat data at fixed intervals of 1s:

S/N 0: Set the time for the servo driver's regular sending of heartbeat to the master station: COB-ID is 601_h, made of up 600_h+Node_ID, data $2B_h$ in the data field is the function code for SDO's writing of 16-bit data, index is 1017_h, subindex is 00_h, the value written in is $03E8_h$ (1000 in decimal system), and the unit is ms. When writing has succeeded, the servo driver will submit heartbeat data packets at intervals of 1s.

S/N 1: Correct answering to SDO's writing function;

S/N 2~ S/N 6: they are the heartbeat data packets regularly submitted by the servo driver to the master station; the COB-ID is 701_h, made up of 700_h+Node_ID, the data in the data field are 1 byte long, representing the servo driver's current state; 04_h represents stopped state, 05_h represents operational state and $7F_h$ represents pre-operational state.

Below is an example of disabling the driver's regular submission of heartbeat data packet.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	01:13:32.321.4			0x00000701	数据帧	标准帧	0x01	05
1	接收	01:13:33.320.8			0x00000701	数据帧	标准帧	0x01	05
2	接收	01:13:34.320.2			0x00000701	数据帧	标准帧	0x01	05
3	接收	01:13:35.319.6			0x00000701	数据帧	标准帧	0x01	05
4	接收	01:13:36.319.0			0x00000701	数据帧	标准帧	0x01	05
5	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 17 10 00 00 00 00 00
6	接收	01:13:37.041.3			0x00000581	数据帧	标准帧	0x08	60 17 10 00 00 00 00 00

Disable the driver's regular submission of heartbeat data packet:

S/N 0~S/N4: They are the heartbeat data packets regularly submitted by the servo driver to the master station; the COB-ID is 701_h, made up of 700_h+Node_ID, data in the data field are 1 byte long, representing the servo driver's current state; 04_h is stopped state, 05_h is operational state, and $7F_h$ is pre-operational state.

S/N 5: It's for setting the time for the servo driver's regular submission of heartbeat data packet; the COB-ID is 601_h, made up of 600_h+Node_ID; data in the data field is $2B_h$, they are the

function code for SDO's writing of 16-bit data (index: 1017_h, subindex: 00_h, and value written: 0000_h, namely 0 in decimal system, and the unit: ms). When writing has succeeded, the servo driver's submission of heartbeat data packet is disabled.

S/N 6: It's SDO writing function's correct answering; the servo driver doesn't submit heartbeat data packets any more;

2.7.2 Master Station Sends Heartbeat Data Packets

When the servo driver is being enabled or running under speed mode, by setting 1016 $_h$, the user can achieve the function of triggering communication-time-out alarm when the servo driver cannot receive the heartbeat data packets from the master station.

Note: After setting 1016 $_h$, the master station's sending of heartbeat data packet need be performed once to trigger the alarm function.

Below is an example where the master station sends its heartbeat data packets to the servo driver at intervals of 1s, and the servo driver's Node_ID is1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 03 00 00 00
2	接收	02:23:53.686.5			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
4	接收	02:23:55.635.4			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 16 10 01 E8 03 01 00
6	接收	02:23:57.850.1			0x00000581	数据帧	标准帧	0x08	60 16 10 01 00 00 00 00
7	发送		成功		0x00000701	数据帧	标准帧	0x01	01
8	发送		成功		0x00000701	数据帧	标准帧	0x01	01
9	发送		成功		0x00000701	数据帧	标准帧	0x01	01
10	发送		成功		0x00000701	数据帧	标准帧	0x01	01
11	发送		成功		0x00000701	数据帧	标准帧	0x01	01
12	发送		成功		0x00000701	数据帧	标准帧	0x01	01
13	发送		成功		0x00000701	数据帧	标准帧	0x01	01
14	发送		成功		0x00000701	数据帧	标准帧	0x01	01
15	发送		成功		0x00000701	数据帧	标准帧	0x01	01
16	发送		成功		0x00000701	数据帧	标准帧	0x01	01
17	接收	02:24:06.292.1			0x00000081	数据帧	标准帧	0x08	30 81 00 00 15 05 00 00

Configure the master station to make it send master station heartbeat data packets to the servo driver at intervals of 1s:

S/N 0: It's an NMT network management command to start the servo driver, its COB-ID is 000_h, data in the data field are 2 bytes long; the first byte 01_h is a command to start, and the second byte is Node_ID.

S/N 1: It's for configuring the servo driver to make it work under speed mode; its COB-ID is

601_h, made up of 600_h+Node_ID, the data $2F_h$ in data field are the function code for SDO's writing of 8-bit data (index: 6060_h, subindex: 00_h, and value written: 03_h).

S/N 2: It's SDO writing function's correct answering;

S/N 3: It's for setting the enabling of servo driver; its COB-ID is 601_h, made up of 600_h+Node_ID, data $2B_h$ in the data field are the function code for SDO's writing of 16-bit data (index: 6040_h, subindex: 00_h, and the value written: $0F_h$);

S/N 4: It's SDO writing function's correct answering;

S/N 5: It's for setting the time for the master station's regular sending of master station heartbeat data packets to the servo driver at fixed intervals; its COB-ID is 601_h, made up of 600_h+Node_ID, the data 23_h in data field is the function code for SDO's writing of 32-bit data (index: 1016_h, subindex: 01_h, and value written: $000103E8_h$); the table below shows data definition:

MSB			LSB
Corresponding bit	31-24	23-16	15-0
Data value	Reserved	Node-ID	Heartbeat time
Data Type	---------	UNSIGNED8	UNSIGNED16

As shown in the table above, 01_h is Node_ID, $03E8_h$ (1000 in decimal system, unit: ms) is the time for regular triggering of the master station's sending of heartbeat data packet.

S/N 6: It's SDO writing function's correct answering;

S/N 7~16: They are master station heartbeat data packets sent at fixed intervals by the master station to the servo driver; their COB-ID is 701_h, made up of 700_h+Node_ID, and data in the data field are 1 byte long.

S/N 17: It's the emergency message sent by the servo driver. When the servo driver doesn't receive the master station heartbeat data packets from the mainframe after the set time interval, the servo driver will send an error report message; its COB-ID is 81_h, made up of 80_h+Node_ID, and the data in the data field is 0000051500008130_h; the table below gives definition of the data:

MEF			ER	EEC
Manufacturer-defined Error Code			Error Register	Emergency Error Code
2Byte	2Byte	1Byte	1Byte	2Byte
Fault code	Warning code	Reserved		

Based on the table above, 0000_h is the manufacturer-defined fault code, 0515_h is manufacturer-defined warning code (communication time out【1】), 00_h is a reserved value, 00_h is error register, and 8130_h is emergency error code (heartbeat error).

Chapter III Motion Mode

3.1 Servo State Control

3.1.1 Servo State Machine

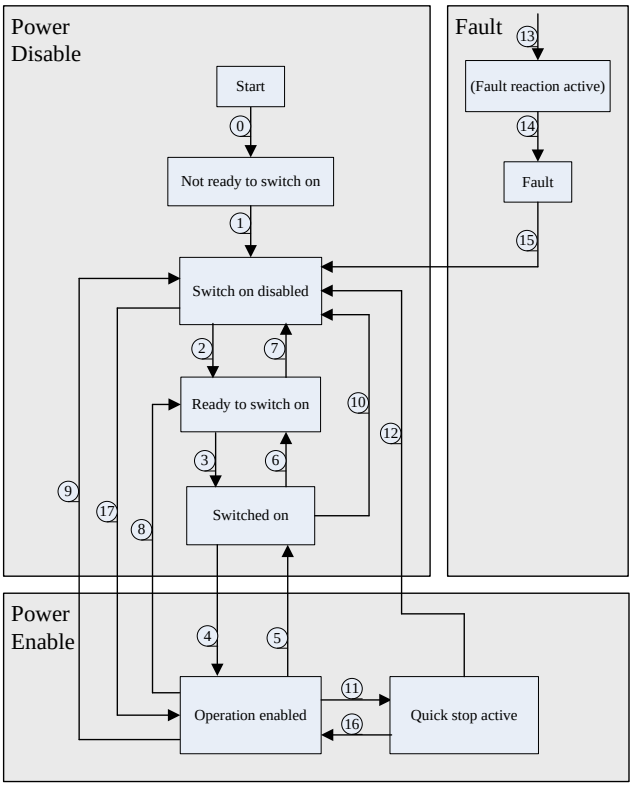


Figure 3-1 CiA 402 State Machine Switching Diagram

Below is the table to describe all the states:

State	Description
Not ready to switch on	Driver's initialization and internal self-test are already completed. Driver parameters can't be set and the driver cannot perform drive functions.
Switch on disabled	Servo driver has no fault or the error is solved. Driver parameters can be set.
Ready to switch on	Servo driver is ready. Driver parameters can be set.

Switched on	Servo driver is waiting to be enabled. Driver parameters can be set.
Operation enabled	Driver is running normally, and is already enabled to operate under some operation mode, when the motor's power is turned on and command value isn't 0, the motor rotates. In the driver parameters, the ones that are "effective immediately" can be set, and others can't be.
Quick stop active	The function of quick stop is activated, and the driver is performing this function. In the driver parameters, the ones that are "effective immediately" can be set, and others can't be.
Fault reaction active	Driver has a fault and is performing fault reaction (namely shutdown). In the driver parameters, the ones that are "effective immediately" can be set, and others can't be.
Fault	Fault reaction is completed, all the drive functions are prohibited, and the user is allowed to change the driver parameters to remove the fault. In case of a resettable fault, after changing the parameters, the user may set Control word 6040h=0x80 to reset the fault.

Control Commands and State Switching:

CiA402 State Switching		Control Word 6040_h	Bit0-bit9 of Status Word 6041_h
0	Start →Not ready to switch on	Natural transition for which no control command is needed	0x0031
1	Not ready to switch on → Switch on disabled	Natural transition for which no control command is needed In case of error during initialization, go to 13 directly	0x0031
2	Switch on disabled →Ready to switch on	0x06	0x0031
3	Ready to switch on →Switched on	0x07	0x0033
4	Switched on → Operation enabled	0x0F	0x0037
5	Operation enabled →Switched on	0x07	0x0033
6	Switched on →Ready to switch on	0x06	0x0031
7	Ready to switch on →Switch on disabled	0x00	0x0031
8	Operation enabled → Ready to switch on	0x06	0x0031
9	Operation enabled → Switch on disabled	0x00	0x0031
10	Switched on →Switch on disabled	0x00	0x0031
11	Operation enabled	0x02	0x0031

	→Quick stop active		
12	Quick stop active →Switch on disabled	Quick stop mode 605A can be set as 0~3. After completion of shutdown is a natural transition, for which no control command is needed	0x0031
13	→Fault reaction active	In any state other than “Fault”, once the servo driver has a fault, it will automatically switch to fault reaction active state, for which no control command is needed.	0x00B8
14	Fault reaction active → Fault	After completion of fault reaction active, there is a natural transition, for which no control command is needed	0x00B8
15	Fault → Switch on disabled	0x80 bit7 is rising edge effective; Bit7 is maintained at 1, and no other control command can be effective.	0x0031
16	Quick stop active →Operation enabled	Quick stop mode 605A can be set at 5~7. After completion of shutdown, 0x0F is sent.	0x0037
17	Switch on disabled → Operation enabled	It’s unnecessary to send 0x06 and 0x07; 0x0F can be directly sent to enter the enabled state	0x0037

3.1.2 Control Word 6040h

Servo driver state switching is achieved through 6040_h; the user can use it to switch the driver between enabled, disabled, fault removal and other states.

Index 6040_h	Name	Control Word					Data Structure	VAR	Data Type	Uint16
	Accessability	RW	Mappability	YES	Related Modes	ALL	Data Range	$0~2^{16}-1$	Ex-works Setting	----

Set control commands:

Bit	Name	Description
0	Switch on	0- ineffective 1- effective
1	(Enable voltage)	0- ineffective 1- effective
2	Quick stop	1- ineffective 0- effective
3	Enable operation	0- ineffective 1- effective
4~6	Operation mode specific	It depends on servo operation modes, see the table below for details.
7	Fault reset	For resettable faults and warning, fault reset function is performed. It's effective on bit7's rising edge; When bit7 is maintained at 1, no other control command can be effective.
8	Halt	0- ineffective 1- effective
9~10	Reserved	Reserved
11~15	Manufacturer specific	Reserved and undefined

Note:

Value assignment to a single bit of the control word is meaningless, a control command can be made only by combination of multiple bits.

Meanings of bit0~bit3 and bit7 remain unchanged for all servo modes. The commands may be sent in order, and the user can switch the process of servo driver to lead it into the expected state by following the CiA 402 state machine (each command corresponds to a definite state), or the user can directly send 0x0F to enter "Operation Enabled" state.

Command	Bits Corresponding to 6040_h				
	7	3	2	1	0
	Fault reset	Operation enabled	Quick stop active	Enable voltage	Ready to switch on
Servo off	0	x	1	1	0
Ready to switch on	0	0	1	1	1
Servo enabling	0	1	1	1	1
Main power supply off	0	x	x	0	x

Quick stop	0	x	0	1	x
Stop operation	0	0	1	1	1
Operation enabled	0	1	1	1	1
Fault reset	Rising edge	x	x	x	x

Meanings of bit4~bit6 depend on servo modes.

6040_h's Corresponding Bit	Control Mode		
	Speed mode Torque mode	Position mode	Homing mode
4	Reserved	New set-point (Positive edge triggering)	(Home operation start) (Positive edge triggering)
5	Reserved	Change set immediately	Reserved
6	Reserved	1 - Relative position command 0 - Absolute position command	Reserved

3.1.3 Status Word 6041h

Servo driver's current state is to be observed through status word 6041_h.

Index 6041_h	Name	Status Word					Data Structure	VAR	Data Type	Uint16
	Accessability	RO	Mappability	TPDO	Related Modes	ALL	Data Range	$0~2^{16}-1$	Ex-works Setting	-

It is used to reflect the servo driver's state:

位bit	Name	Description
0	Ready to switch on	0 - Servo driver is not ready to switch on 1 –Servo driver is ready to switch on
1	Switched on	0 - Servo driver is not waiting to be enabled 1 –Servo driver is waiting to be enabled
2	Operation enabled	0 - Servo driver operation is not enabled 1 - Servo driver operation is already enabled
3	Fault	0 - No fault exists 1 - A fault has happened
4	(Voltage enabled)	0 - Main circuit's power is not on 1 - Main circuit's power is on
5	Quick stop active	0 - Quick stop is already performed 1 - Quick stop is not performed

	6	(Switch on disabled)	0 - Servo driver is not switched off 1 - Servo driver is already switched off	
	7	Warning	0 - No warning is given 1 - Warning is given	
	8	(Manufacturer)	Reserved, always as 0	
	9	Remote	Always as 0	
	10	Target reached	0 - The target is not yet reached 1 - The target is already reached	
	11	internal limit active	0 - No positional overtravel limit exists 1 - Positional overtravel is occuring	
	12~13	Operation mode specific	It's related to each servo mode; for details, please refer to the table below	
	14~15	(Manufacturer specific)	Reserved	

Note:

Reading a single bit of a status word is meaningless, the servo driver's current state can only be fed back through combination of multiple bits.

Meanings of bit0~bit11 remain unchanged for all servo modes. When the control word 6040_h has sent commands in the required order, the servo driver feeds back the confirmed state.

Values of bits 0-3, 5 and 6 are in the table below:

Values of Bits 0-3,5,6 (binary)	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	（Switch on)
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Meanings of bit12~bit13 depend on the servo drivers' modes.

6041_h's Corresponding Bit	Control Mode			
	Speed mode	Torque mode	Position mode	Homing mode
12	Zero speed state	Reserved	Set-point Acknowledge	Homing attained
13	Reserved	Reserved	Following error	Homing error

3.2 Servo Operation Modes

3.2.1 Operation Modes Supported by the Servo Driver (6502_h)

LS low-voltage servo drivers' CANopen protocol supports 4 servo modes. Object dictionary index 6502_h is used to display servo modes supported by the servo driver.

Operation modes supported by the servo driver (6502_h)

Index 6502_h	Name	Status Word					Data Structure	VAR	Data Type	Uint32
	Accessability	RO	Mappability	TPDO	Related Modes	-	Data Range	$0\sim2^{32}-1$	Ex-works Setting	CF_h

Description of servo operation modes supported by the driver:

Bit	Description	Support or not: 0- not supported; 1-supported
0	pp (position mode)	1
1	Reserved	1
2	pv (speed mode)	1
3	tq (torque mode)	1
4	Reserved	0
5	hm (homing mode)	1
6	ip (interpolation mode)	——
7	csp (cycling synchronous position mode)	0
8	csv (cycling synchronous velocity mode)	0
9	Cst (cycling synchronous torque mode)	0
10~31	Reserved	0

If the CANopen equipment supports object dictionary index 6502_h, the user can use it to find out what servo modes the driver supports.

3.2.2 Index 6060h for Control Modes

Servo driver's operation modes can be set through object dictionary index 6060_h.

Servo control mode selection (6060_h)

Index 6060_h	Name	Modes of Operation					Data Structure	VAR	Data Type	int8
	Accessablity	RW	Mappability	YES	Related	ALL	Data	$-2^7\sim2^7-1$	Ex-works	---

					Modes		Range		Setting	

Selection of servo operation mode:

Value	Description
1	Position mode
3	Speed mode
4	Torque mode
6	Homing mode

The modes supported are can still be updated.

Tips about mode switching:

When the servo driver switches from position mode to any other mode, no matter what its state has been, the position commands unexecuted under the position mode will be aborted.

When the servo driver switches from speed mode to any other mode, no matter what its state has been, slope stop (stop deceleration 6084_h) will be performed first. Only when stop has completed, can the driver switch to any other mode.

3.2.3 Control Mode Display 6061h

Servo driver's current operation mode can be observed through object dictionary index 6061_h.

Mode display (6061_h)

Index 6061_h	Name	Modes of Operation Display					Data Structure	VAR	Data Type	int8
	Accessablity	RO	Mappability	TPDO	Related Modes	ALL	Data Range	-2^7~2^7-1	Ex-works Setting	---

Servo driver's actual operation modes to be displayed:

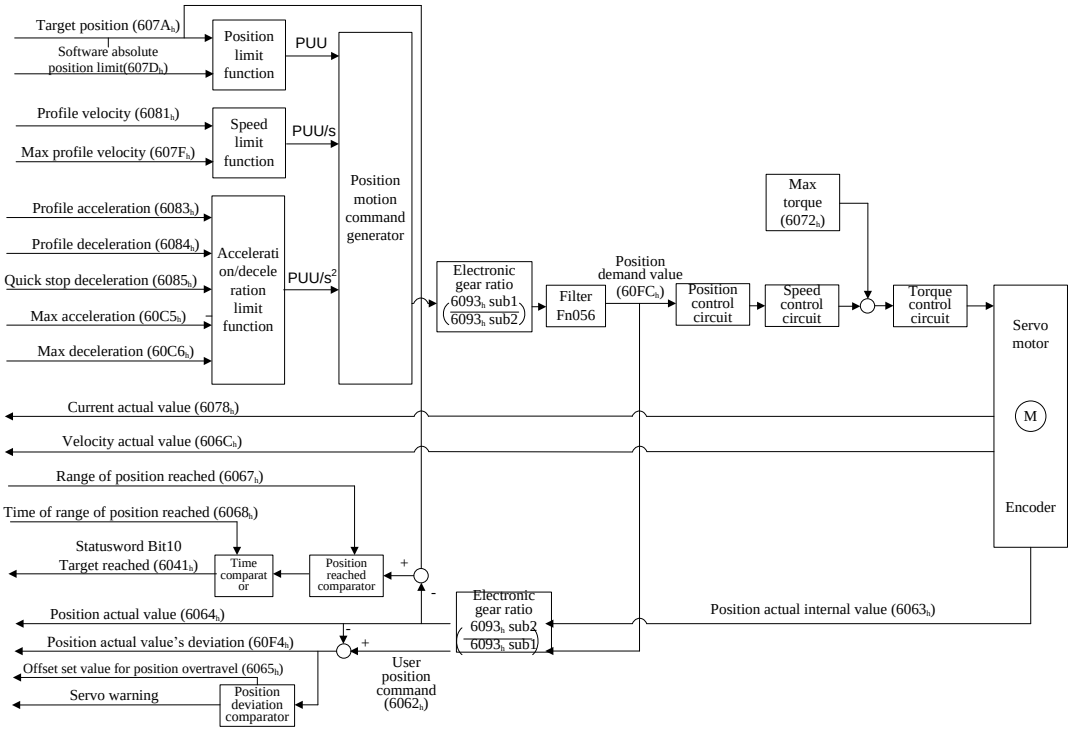
Value	Description
1	Position mode
3	Speed mode
4	Torque mode
6	Homing mode

The modes supported can still be updated.

3.3 Position Mode

When the servo driver has received a position command from the upper device, the servo driver controls the servo motor to make it reach the target position. Under position control mode, the upper device only gives the driver at the initial time the target position, speed command, acceleration, deceleration and other relevant setting items; the planning for the movement from command triggering to reaching the target position is to be performed by the driver internally.

3.3.1 Control Block Diagram of Position Mode



3.3.2 Setting of Relevant Objects Under Position Mode

Index	Subindex	Name	Accessibility	Data Type	Unit	Mappability
6040$_h$	00$_h$	Control Word	RW	Uint16	-	RPDO
6041$_h$	00$_h$	Status Word	RO	Uint16	-	TPDO
6060$_h$	00$_h$	Mode Of Operation	RW	int8	-	YES
6061$_h$	00$_h$	Modes of Operation Display	RO	int8	-	TPDO
6062$_h$	00$_h$	User position command	RO	int32	puu	TPDO
6063$_h$	00$_h$	Position Actual Internal Value	RO	int32	puu	TPDO
6064$_h$	00$_h$	Position Actual Value	RO	int32	puu	TPDO
6065$_h$	00$_h$	Offset Set Value for Position Overtravel	RW	Uint32	puu	NO
6067$_h$	00$_h$	Range of Position Reached	RW	Uint32	puu	NO
6068$_h$	00$_h$	Time of Range of Position Reached	RW	Uint16	ms	NO
607A$_h$	00$_h$	Target Position	RW	int32	puu	YES

6081_h	00_h	Profile Velocity	RW	Uint32	puu/s	YES
6083_h	00_h	Profile Acceleration	RW	Uint32	puu/s^2	YES
6084_h	00_h	Profile Deceleration	RW	Uint32	puu/s^2	YES
$60F4_h$	00_h	Position Actual Value's Deviation	RO	int32	puu	TPDO
$60FC_h$	00_h	Position Demand Value	RO	int32	puu	TPDO

3.3.3 Example of Operation under Position Mode

3.3.3.1 Steps of Operation for Control by Use of SDO under Position Mode

1. Set the mode: 6060_h-00_h = 01_h, to select position control mode.
2. Set the target position: $607A_h$-00_h, unit: puu (number of pulses)
3. Set speed command: 6081_h-00_h, unit: puu/s (number of pulses/second); If speed command is changed during operation, the triggering edge need be supplied again, or the change won't become effective.
4. Set acceleration command 6083_h-00_h, unit is puu/s^2 (pulses per square second)
5. Set deceleration command 6084_h-00_h, unit is puu/s^2 (pulses per square second).
6. Set control command 6040_h-00_h = $0F_h$.
7. Set control command for absolute position immediate triggering: 6040h-00_h =$3F_h$.

In the example below, the CANOPEN node No. is 1, the user uses SDO to change corresponding index's value to make the motor work under position mode, and to perform the command for absolute position immediate triggering.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 01 00 00 00
2	接收	03:04:55.053.8			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	23 7A 60 00 10 27 00 00
4	接收	03:04:55.063.9			0x00000581	数据帧	标准帧	0x08	60 7A 60 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 81 60 00 D0 07 00 00
6	接收	03:04:55.073.8			0x00000581	数据帧	标准帧	0x08	60 81 60 00 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	23 83 60 00 10 27 00 00
8	接收	03:04:55.083.8			0x00000581	数据帧	标准帧	0x08	60 83 60 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	23 84 60 00 20 4E 00 00
10	接收	03:04:55.093.8			0x00000581	数据帧	标准帧	0x08	60 84 60 00 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
12	接收	03:04:55.103.8			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
13	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 3F 00 00 00
14	接收	03:04:55.113.8			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00

Meanings of the commands sent:

S/N 0: Start up;

S/N 1: Operation mode setting is: index 6060_h, subindex 00_h,value 01_h; the driver's control mode is set as position control mode;

S/N 2: Setting succeeds, and answering is correct;

S/N3: Target position setting is: index $607A_h$, subindex 00_h, value 00002710_h; unit is puu (number of pulses);

S/N4: Setting succeeds, and answering is correct;

S/N 5: Speed command setting is: index is 6081_h, subindex 00_h, value $000007D0_h$; unit is puu/s (number of pulses run every second);

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Setting of the slope over acceleration time is: index 6083_h, subindex 00_h, value 00002710_h; unit is puu/s (number of pulses run every second);

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Setting of the slope over deceleration time is: index 6084_h, subindex 00_h, value $00004E20_h$; unit is puu/s (number of pulses run every second);

S/N 10: Setting succeeds, and answering is correct;

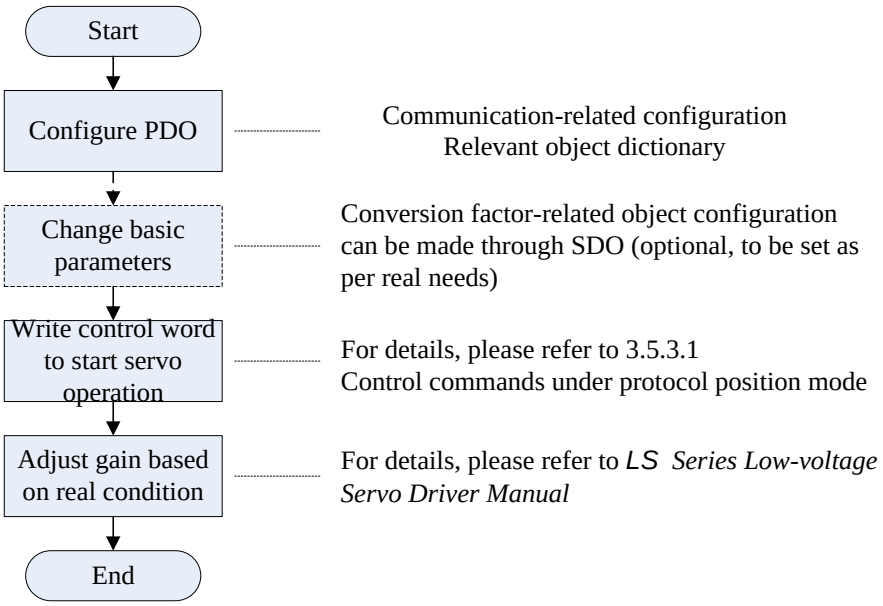
S/N 11: It's a control command to control enabling of the driver: index 6040_h, subindex 00_h, value $000F_h$; the motor is enabled by it;

S/N 12: Setting succeeds, and answering is correct;

S/N 13: The command is triggered on positive edge, and is then being performed: index 6040_h, subindex 00_h, value $003F_h$; the motor starts to execute the command.

S/N 14: Setting succeeds, and answering is correct.

3.3.3.2 Steps of Operation for Control by Use of PDO under Position Mode



RPDO Mapping

In the example below, RPDO1 is mapped to asynchronous, device-sub-protocol-specific mode, and is used to only send position commands; the CANOPEN node No. is 1.

RPDO2 is mapped to asynchronous, device-sub-protocol-specific mode, and is used to only send Control words, the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 14 02 FF 00 00 00
2	接收	02:39:54.194.6			0x00000581	数据帧	标准帧	0x08	60 00 14 02 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 00 00 00 00
4	接收	02:39:55.385.5			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 00 16 01 20 00 7A 60
6	接收	02:39:56.432.2			0x00000581	数据帧	标准帧	0x08	60 00 16 01 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 01 00 00 00
8	接收	02:39:57.551.6			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 14 02 FF 00 00 00
10	接收	02:39:59.231.2			0x00000581	数据帧	标准帧	0x08	60 01 14 02 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 16 00 00 00 00 00
12	接收	02:40:00.266.7			0x00000581	数据帧	标准帧	0x08	60 01 16 00 00 00 00 00
13	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 16 01 10 00 40 60
14	接收	02:40:01.354.1			0x00000581	数据帧	标准帧	0x08	60 01 16 01 00 00 00 00
15	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 16 00 01 00 00 00
16	接收	02:40:02.505.1			0x00000581	数据帧	标准帧	0x08	60 01 16 00 00 00 00 00

Meanings of the commands sent:

S/N 0: Start up;

S/N 1: Communication parameters setting is: index 1400h, subindex 02h, value FFh; the transmission type is asynchronous, device-sub-protocol-specific event;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index 1600_h, subindex 00_h, value 00_h; the number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index 1600_h, subindex 01_h, value $607A0020_h$, mapped to index $607A_h$, subindex 00_h; the object is a 32-bit position command.

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index 1600_h, subindex 00_h, value 01_h; the number of entries is set at 1.

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Communication parameter setting is: index 1401_h, subindex 02_h, value FF_h, and transmission type is asynchronous, device-sub-protocol-specific event;

S/N 10: Setting succeeds, and answering is correct;

S/N 11: Mapping parameter setting is: index 1601_h, subindex 00_h, value 00_h; the number of entries is reset;

S/N 12: Setting succeeds, and answering is correct;

S/N 13: Mapping parameter setting is: index 1601_h, subindex 01_h, value 60400010_h, mapped to index 6040_h, subindex 00_h; the object is a 16-bit control word

S/N 14: Setting succeeds, and answering is correct;

S/N 15: Mapping parameter setting is: index 1601_h, subindex 00_h, value 01_h; the number of entries is set at 1.

S/N 16: Setting succeeds, and answering is correct.

TPDO Mapping

In the example below, TPDO 1 is mapped to status word, position feedback; the servo driver will send current status word and position value every time two synchronous objects are received; the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 18 02 02 00 00 00
2	接收	06:51:21.595.2			0x00000581	数据帧	标准帧	0x08	60 00 18 02 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 00 00 00 00
4	接收	06:51:21.605.3			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 01 10 00 41 60
6	接收	06:51:21.615.3			0x00000581	数据帧	标准帧	0x08	60 00 1A 01 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 02 20 00 64 60
8	接收	06:51:21.625.3			0x00000581	数据帧	标准帧	0x08	60 00 1A 02 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 02 00 00 00
10	接收	06:51:21.635.2			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00

Meanings of the commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1800_h, subindex 02_h, value 02_h; transmission type is one transmission each time two synchronous objects are received;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 00_h; the number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index $1A00_h$, subindex 01_h, value 60410010_h, mapped to index 6041_h, subindex 00_h; the object is a 16-bit status word;

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index $1A00_h$, subindex 02_h, value 60640020_h, mapped to index 6064_h, subindex 00_h; the object is 32-bit actual position;

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 02_h; the number of entries is set at 2.

S/N 10: Setting succeeds, and answering is correct.

RPDO Command

In the example below, RPDO1 is mapped to a target position command, and RPDO2 is mapped to a Control word; CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 01 00 00 00
1	接收	03:15:12.867.7			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
2	发送		成功		0x00000601	数据帧	标准帧	0x08	23 81 60 00 D0 07 00 00
3	接收	03:15:12.878.1			0x00000581	数据帧	标准帧	0x08	60 81 60 00 00 00 00 00
4	发送		成功		0x00000601	数据帧	标准帧	0x08	23 83 60 00 10 27 00 00
5	接收	03:15:12.887.7			0x00000581	数据帧	标准帧	0x08	60 83 60 00 00 00 00 00
6	发送		成功		0x00000601	数据帧	标准帧	0x08	23 84 60 00 20 4E 00 00
7	接收	03:15:12.897.7			0x00000581	数据帧	标准帧	0x08	60 84 60 00 00 00 00 00
8	发送		成功		0x00000201	数据帧	标准帧	0x08	10 27 00 00 00 00 00 00
9	发送		成功		0x00000301	数据帧	标准帧	0x08	0F 00 00 00 00 00 00 00
10	发送		成功		0x00000301	数据帧	标准帧	0x08	3F 00 00 00 00 00 00 00

Meanings of commands sent:

S/N 0: Operation mode setting is: index 6060_h, subindex 00_h, value 01_h; the driver is set to work under position control mode;

S/N 1: Setting succeeds, and answering is correct;

S/N 2: Speed command setting is: index 6081_h, subindex 00_h, value $000007D0_h$; the unit is puu/s (number of pulses run every second);

S/N 3: Setting succeeds, and answering is correct;

S/N 4: Setting of slope over the acceleration time is: index 6083_h, subindex 00_h, value 00002710_h; the unit is puu/s (number of pulses run every second);

S/N 5: Setting succeeds, and answering is correct;

S/N 6: Setting of slope over the deceleration time is: index 6084h, subindex 00h, value 00004E20h; the unit is puu/s (number of pulses run every second);

S/N 7: Setting succeeds, and answering is correct;

S/N 8: RPDO1 is for setting target position: value: 00002710_h, unit: puu (number of pulses);

S/N 9: RPDO2 is for setting the control word to enable the driver: value: $000F_h$;

S/N 10: RPDO2 is for setting the control word to trigger a command, and let the driver then execute the command: value: $003F_h$, absolute position mode, to be executed immediately.

TPDO Command

In the example below, TPDO1 is mapped to feedback of current status word and position; the condition for occurrence of feedback event is reception of two synchronous objects; the CANOPEN node No. is 1.

To feed back the actual current status word and position, the COB-ID is 181_h.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000080	数据帧	标准帧	0x00	
1	发送		成功		0x00000080	数据帧	标准帧	0x00	
2	接收	04:09:02.345.4			0x00000181	数据帧	标准帧	0x06	37 14 10 27 00 00

Meanings of the commands sent:

S/N 0: Synchronous object frame 1;

S/N 1: Synchronous object frame 2;

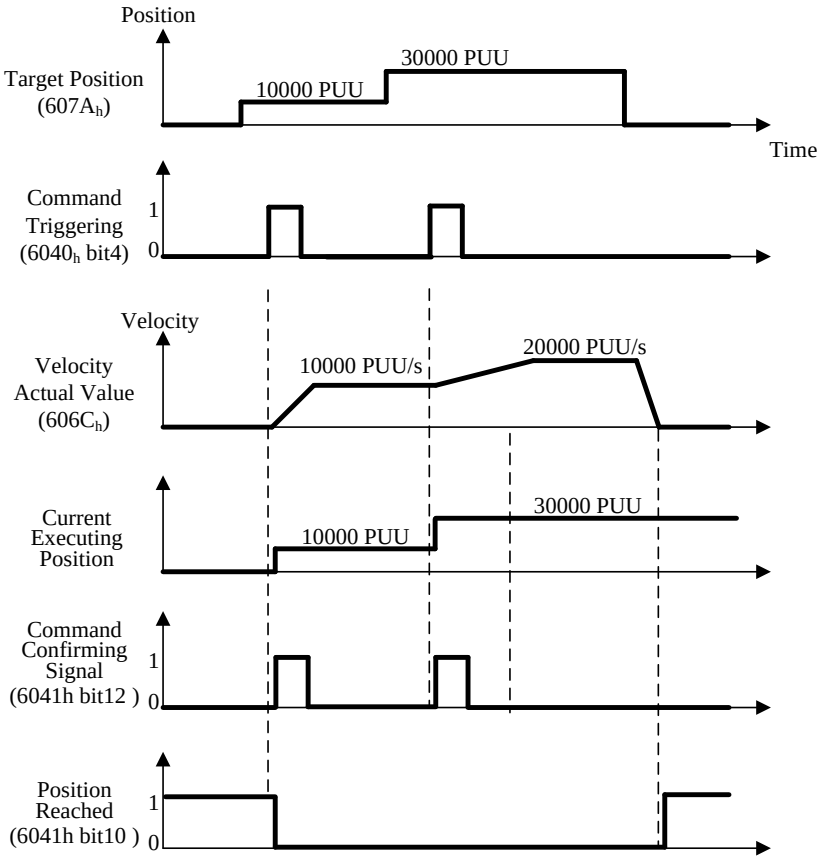
S/N 2: Return value is 000027101437_h, in which 00002710_h is the motor's actual position, and 1437_h is the status word.

3.3.4 Immediately Effective Mode and Non-immediately Effective Mode under Position Mode

Under position control mode, there are 2 modes of position command becoming effective: immediately effective and non-immediately effective. The user uses control word 6040_h's bit 5 to select between them.

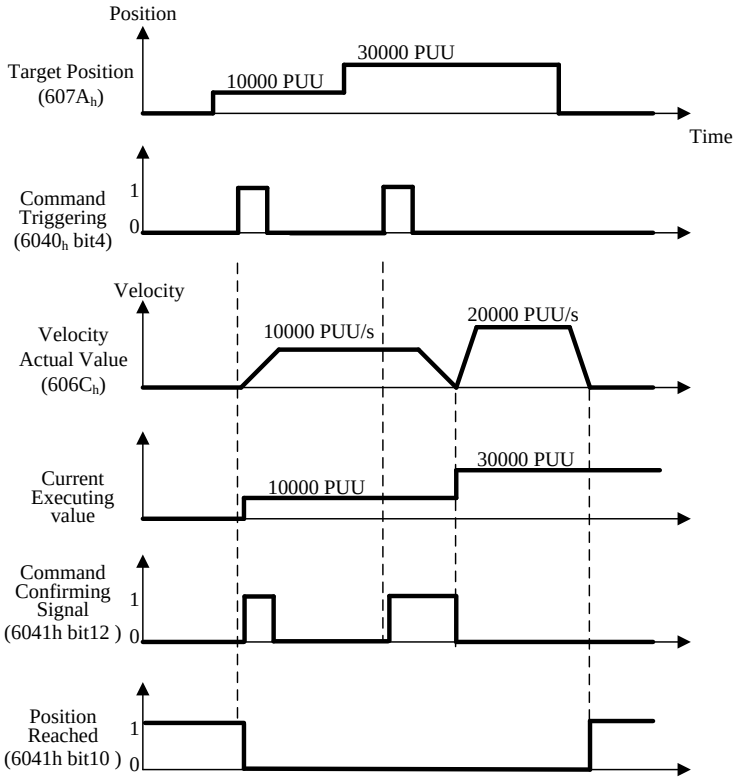
3.3.4.1 Immediately Effective Mode under Position Mode

If 6040_h's bit5 is set at 1, the command will become effective immediately. If, when the immediately effective mode is enabled, the current motion command is still being executed (i.e. its execution is not completed yet), and the servo driver receives a new command's triggering, the servo driver will immediately stop execution of the current command and start executing the new command.



3.3.4.2 Non-immediately Effective Mode under Position Mode

If 6040$_h$'s bit5 is set at 0, the suspension of command will immediately become effective. If, when the immediately effective mode is not enabled, the current motion command is still being executed (i.e. its execution is not completed yet), even if the servo driver has received a new command's triggering, the servo driver will still continue execution of the current motion command, and start executing the new command only after execution of the current command is completed.



3.3.5 Customization Function under Position Mode

By setting the driver's Fn05F (Index 205F-00$_h$), the user can make the driver work under a customized position-triggering mode. This parameter is operational only under CANOPEN position mode.

Fn05F's default value is 0, corresponding to standard triggering mode, under which execution of position commands will be triggered only when Control word 6040$_h$'s bit4 (NewSetPoint) provides rising-edge triggering.

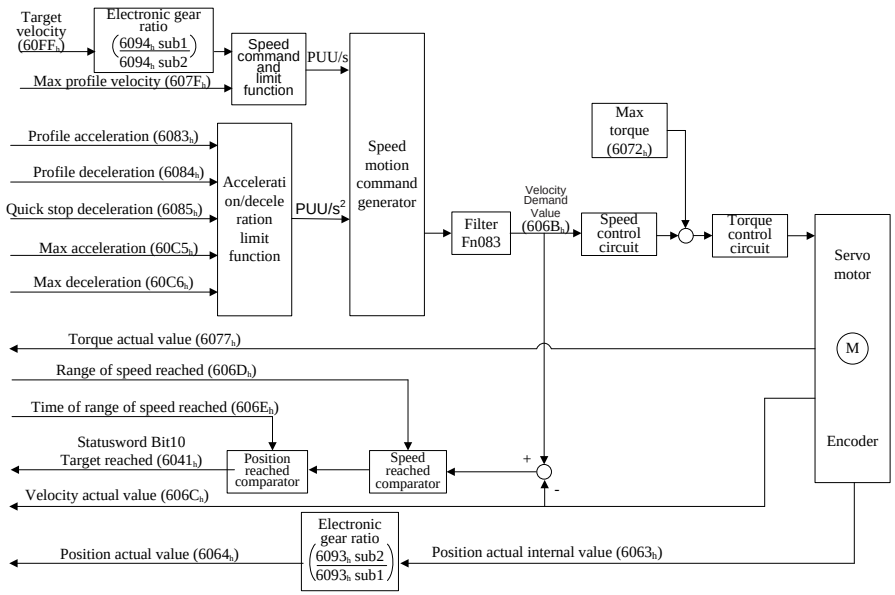
Fn05F's Value	Function
-1	Only if the servo driver is enabled, internal tracking of absolution position will be performed at fixed intervals of 10ms; such function is effective only under absolute position mode.
0	It's the value for standard mode, under which triggering of execution needs control word 6040$_h$'s bit4 (NewSetPoint) to be on rising edge
N	Only if the servo driver is enabled, internal tracking of absolute position will be performed at fixed interval of N ms; unit: ms

In the customization modes above, the first one (Fn05F=-1) and third (Fn05F=N) can be used for fixed-interval updating of helm's real-time position.

3.4 Speed Mode

Under speed mode, when the user has set speed, acceleration and deceleration, the servo driver can plan the motor's speed diagram as per their setting, and achieve smooth switching between different speed commands.

3.4.1 Control Block Diagram of Speed Mode



3.4.2 Setting of Speed Mode-related Objects

Index	Subindex	Name	Accessability	Data Type	Unit	Mappability
6040$_h$	00$_h$	Control Word	RW	Uint16	-	RPDO
6041$_h$	00$_h$	Status Word	RO	Uint16	-	TPDO
6060$_h$	00$_h$	Mode Of Operation	RW	int8	-	YES
6061$_h$	00$_h$	Modes of Operation Display	RO	int8	-	TPDO
6063$_h$	00$_h$	Position Actual Internal Value	RO	int32	puu	TPDO
6064$_h$	00$_h$	Position Actual Value	RO	int32	puu	TPDO
606B$_h$	00$_h$	Velocity Demand Value	RO	int32	puu/s	TPDO
606C$_h$	00$_h$	Velocity Actual Value	RO	int32	puu/s	TPDO

$606D_h$	00_h	Range of Speed Reached	RW	Uint16	puu/s	NO
$606E_h$	00_h	Time of Range of Speed Reached	RW	Uint16	ms	NO
$606F_h$	00_h	Zero Speed Threshold	RW	Uint16	puu/s	NO
6081_h	00_h	Speed Command	RW	Uint32	puu/s	YES
6083_h	00_h	Profiled Acceleration	RW	Uint32	puu/s^2	YES
6084_h	00_h	Profile Deceleration	RW	Uint32	puu/s^2	YES
6094_h	01_h	Velocity Encoder Factor's Numerator	RW	Uint32	-	YES
	02_h	Velocity Encoder Factor's Divisor	RW	Uint32	-	YES
$60C5_h$	00_h	Max acceleration	RW	Uint32	puu/s^2	YES
$60C6_h$	00_h	Max deceleration	RW	Uint32	puu/s^2	YES
$60FF_h$	00_h	Target Velocity	RW	int32	puu/s	RPDO

3.4.3 Example of Speed Mode Operation

3.4.3.1 Steps of Operation for Control by Use of SDO under Speed Mode

1. Set the mode: 6060_h-00_h = 03_h, speed control mode.
2. Set the slope over acceleration time (6083_h-00_h); the unit is puu/s^2 (pulses per square second).
3. Set the slope over deceleration time (6084_h-00_h); the unit is puu/s^2 (pulses per square second).
4. Set the control command: 6040_h-00_h = $0F_h$.
5. Set the target speed (60FFh-00_h), the unit is puu/s^2 (pulses per second).

In the example below, CANOPEN node No. is 1, the user uses SDO to change corresponding index's value, to make the motor work under speed mode.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 03 00 00 00
2	接收	03:53:33.348.4			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	23 83 60 00 20 4E 00 00
4	接收	03:53:33.358.5			0x00000581	数据帧	标准帧	0x08	60 83 60 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 84 60 00 40 9C 00 00
6	接收	03:53:33.368.4			0x00000581	数据帧	标准帧	0x08	60 84 60 00 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
8	接收	03:53:33.378.1			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	23 FF 60 00 10 27 00 00
10	接收	03:53:33.388.9			0x00000581	数据帧	标准帧	0x08	60 FF 60 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Operation mode setting is: index 6060_h, subindex 00_h, value 03_h; the driver's operation

mode is set as speed control mode;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Profile acceleration's setting is: index 6083_h, subindex 00_h, value $00004E20_h$; the unit is puu/s (number of pulses run per second);

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Profile deceleration setting is: index 6084_h, subindex 00_h, value $00009C40_h$; the unit is puu/s (number of pulses run per second);

S/N 6: Setting succeeds, and answering is correct;

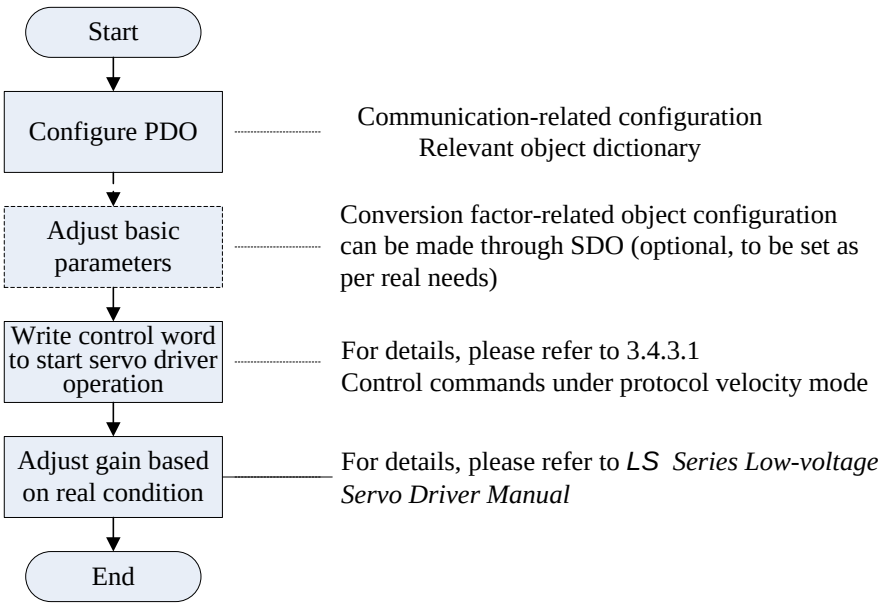
S/N 7: Setting of the motor's enabling is: index 6040_h, subindex 00_h, value $000F_h$;

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Speed command setting is: index $60FF_h$, subindex 00_h, value 00002710_h; the unit is puu/s (number of pulses run per second).

S/N 10: Setting succeeds, and answering is correct;

3.4.3.2 Steps of Operation for Control by Use of PDO under Speed Mode



RPDO Mapping

In the example below, RPDO1 is mapped to asynchronous device-sub-protocol-specific mode, only for sending of speed commands; CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	16:56:22.564	成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送	16:56:22.575	成功		0x00000601	数据帧	标准帧	0x08	2F 00 14 02 FF 00 00 00
2	接收	16:56:22.580			0x00000581	数据帧	标准帧	0x08	60 00 14 02 00 00 00 00
3	发送	16:56:22.585	成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 00 00 00 00
4	接收	16:56:22.590			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
5	发送	16:56:22.595	成功		0x00000601	数据帧	标准帧	0x08	23 00 16 01 20 00 FF 60
6	接收	16:56:22.599			0x00000581	数据帧	标准帧	0x08	60 00 16 01 00 00 00 00
7	发送	16:56:22.606	成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 01 00 00 00
8	接收	16:56:22.613			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1400_h, subindex 02_h, value FF_h; transmission type is asynchronous, device sub-protocol-specific event;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index 1600_h, subindex 00_h, value is 00_h; number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index 1600_h, subindex 01_h, value $60FF0020_h$, mapped to index $60FF_h$, subindex 00_h; the object is a 32-bit command speed.

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index 1600_h, subindex 00_h, value 01_h; number of entries is set at 1.

S/N 8: Setting succeeds, and answering is correct.

TPDO Mapping

In the example below, TPDO1 is mapped to speed feedback and position feedback, the servo driver transmits actual current speed value and position

value once every time two synchronous objects are received; the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送	16:56:22.564	成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送	16:56:22.766	成功		0x00000601	数据帧	标准帧	0x08	2F 00 18 02 02 00 00 00
2	接收	16:56:22.769			0x00000581	数据帧	标准帧	0x08	60 00 18 02 00 00 00 00
3	发送	16:56:22.776	成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 00 00 00 00
4	接收	16:56:22.782			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00
5	发送	16:56:22.786	成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 01 20 00 6C 60
6	接收	16:56:22.791			0x00000581	数据帧	标准帧	0x08	60 00 1A 01 00 00 00 00
7	发送	16:56:22.796	成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 02 20 00 63 60
8	接收	16:56:22.800			0x00000581	数据帧	标准帧	0x08	60 00 1A 02 00 00 00 00
9	发送	16:56:22.806	成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 02 00 00 00
10	接收	16:56:22.809			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1800_h, subindex 02_h; transmission type is one transmission every time two synchronous objects are received;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 00_h; number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index $1A00_h$, subindex 01_h, value $606C0020_h$, mapped to index $606C_h$, subindex 00_h; the object is 32-bit actual speed value;

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index $1A00_h$, subindex 02_h, value 60630020_h, mapped to index 6063_h, subindex 00_h; the object is 32-bit actual position value;

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 02_h; number of entries is set at 2.

S/N 10: Setting succeeds, and answering is correct.

RPDO Command

In the example below, RPDO1 is already mapped to speed command, and

the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 03 00 00 00
1	接收	00:08:42.154.0			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
2	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
3	接收	00:08:44.190.1			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
4	发送		成功		0x00000201	数据帧	标准帧	0x08	A0 86 01 00 00 00 00 00

Meanings of commands sent:

S/N 0: Operation mode setting is: index 6060_h, subindex 00_h, value 03_h; the driver's operation mode is set as speed control mode;

S/N 1: Setting succeeds, and answering is correct;

S/N 2: Setting of the motor's enabling is: index 6040_h, subindex 00_h, value $000F_h$;

S/N 3: Setting succeeds, and answering is correct;

S/N 4: Speed command setting is: index $60FF_h$, subindex 00_h, value $000186A0_h$; the unit is puu/s (number of pulses run per second).

TPDO Command

In the example below, TPDO1 is already mapped to feedback of actual current speed value and position value; the condition of occurrence of feedback event is reception of two synchronous objects; the CANOPEN node No. is 1.

For commands to feed back actual current speed value and position value, COB-ID is 181_h.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000080	数据帧	标准帧	0x00	
1	发送		成功		0x00000080	数据帧	标准帧	0x00	
2	接收	02:08:01.869.1			0x00000181	数据帧	标准帧	0x08	18 87 01 00 18 A9 0D 00

Meanings of commands sent:

S/N 0: Synchronous object frame 1;

S/N 1: Synchronous object frame 2;

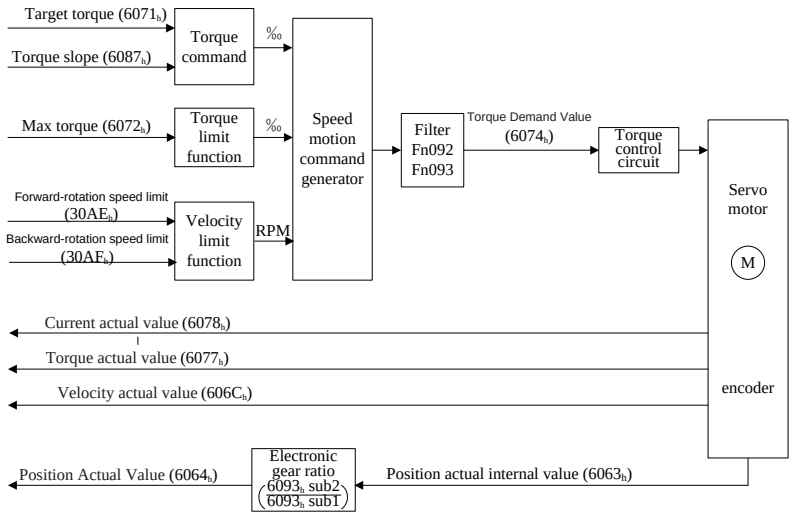
S/N 2: The actual speed value is 00018718_h, and actual position value is $000DA918_h$.

3.5 Torque Mode

Under torque mode, the upper device specifies torque command and

filtering conditions first, then the driver's motion command generator generates the torque slope based on these conditions.

3.5.1 Control Block Diagram of Torque Mode



3.5.2 Setting of Objects Related to Torque Mode

Index	Subindex	Name	Accessibility	Data Type	Unit	Mappability
6040_h	00_h	Control Word	RW	Uint16	-	RPDO
6041_h	00_h	Status Word	RO	Uint16	-	TPDO
6060_h	00_h	Mode of Operation	RW	int8	-	YES
6061_h	00_h	Modes of Operation Display	RO	int8	-	TPDO
6071_h	00_h	Target Torque	RW	int16	‰Tn	YES
6074_h	00_h	Torque Demand Value	RO	int16	‰Tn	TPDO
6075_h	00_h	Motor Rated Current	RW	Uint32	mA	NO
6077_h	00_h	Torque Actual Value	RO	int16	‰Tn	TPDO
6078_h	00_h	Current Actual Value	RO	int16	mA	TPDO
6087_h	00_h	Torque Slope	RW	Uint32	‰Tn	YES
$30AE_h$	00_h	Basic Forward-rotation Speed Limit under Toque Mode	RW	Uint16	RPM	YES
$30AF_h$	00_h	Basic Backward-rotation Speed Limit under Toque Mode	RW	Uint16	RPM	YES

3.5.3 Example of Torque-mode Operation

3.5.3.1 Steps of Operation for Control by Use of SDO under Torque Mode

1. Set the mode: 6060_h-00_h = 04_h (torque control mode).
2. Set the basic forward-rotation speed limit under torque mode: $30AE_h$-00_h, unit: RPM (number of turns per minute).
3. Set basic backward-rotation speed limit under torque mode: $30AF_h$-00_h, unit: RPM (number of turns per minute).
4. Set torque slope: 6087_h-00_h, unit: ‰Tn (thousandth of rated torque).
5. Set target torque: 6071_h-00_h, unit: ‰Tn (thousandth of rated torque).
6. Set the control command: 6040_h-00_h = $0F_h$.

In the example below, CANOPEN node No. is 1, the user uses SDO to change corresponding index's value, so that he can make the motor work under torque mode.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 04 00 00 00
2	接收	01:11:38.772.8			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2B AE 30 00 64 00 00 00
4	接收	01:11:39.668.0			0x00000581	数据帧	标准帧	0x08	60 AE 30 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	2B AF 30 00 64 00 00 00
6	接收	01:11:40.546.6			0x00000581	数据帧	标准帧	0x08	60 AF 30 00 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	23 87 60 00 C8 00 00 00
8	接收	01:11:41.448.8			0x00000581	数据帧	标准帧	0x08	60 87 60 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 71 60 00 F4 01 00 00
10	接收	01:11:42.291.6			0x00000581	数据帧	标准帧	0x08	60 71 60 00 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
12	接收	01:11:43.039.8			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Operation mode setting is: index 6060_h, subindex 00_h, value 04_h; the driver is set to work under torque control mode;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Setting of basic forward-rotation speed limit under torque mode is: index $30AE_h$, subindex 00_h, value 0064_h; the unit is RPM (number of turns per minute);

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Setting of basic backward-rotation speed limit under torque mode is: index $30AF_h$, subindex 00_h, value 0064_h; the unit is RPM (number of turns per second);

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Torque slope's setting is: index 6087_h, subindex 00_h, value $00C8_h$; the unit is ‰Tn (thousandth of rated torque);

S/N 8: Setting succeeds, and answering is correct;

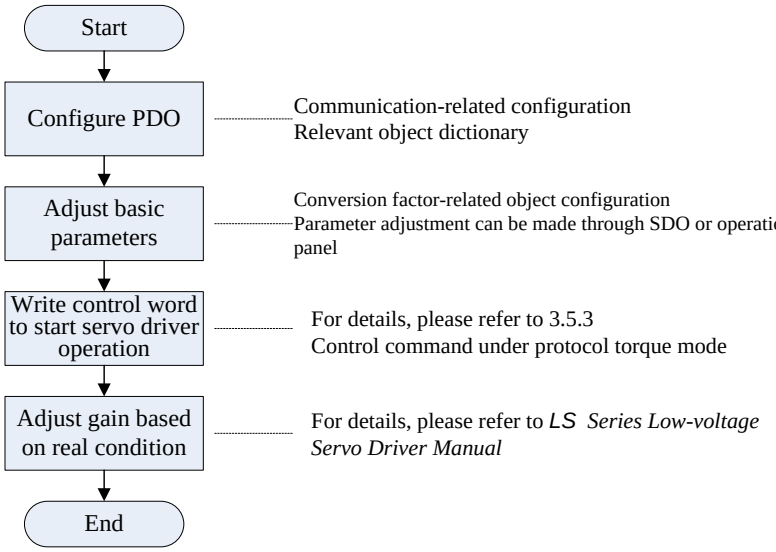
S/N 9: Target torque setting is: index 6071_h, subindex 00_h, value $01F4_h$; the unit is ‰Tn (thousandth of rated torque);

S/N 10: Setting succeeds, and answering is correct;

S/N 11: Setting of enabling of the motor is: index 6040_h, subindex 00_h, value $000F_h$;

S/N 12: Setting succeeds, and answering is correct;

3.5.3.2 Steps of Operation for Control by Use of PDO under Torque Mode



RPDO Mapping

In the example below, RPDO1 is mapped to asynchronous-device sub-protocol-specific mode, only for sending target torque commands; CANOPEN node No. is 1.

RPDO2 is mapped to asynchronous device-sub-protocol-specific mode, only for sending commands for basic forward-rotation speed limit under torque mode and basic backward-rotation speed limit under torque mode; CANOPEN node No.

is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 14 02 FF 00 00 00
2	接收	01:52:39.334.6			0x00000581	数据帧	标准帧	0x08	60 00 14 02 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 00 00 00 00
4	接收	01:52:40.165.9			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 00 16 01 10 00 71 60
6	接收	01:52:40.987.6			0x00000581	数据帧	标准帧	0x08	60 00 16 01 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 16 00 01 00 00 00
8	接收	01:52:41.762.9			0x00000581	数据帧	标准帧	0x08	60 00 16 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 14 02 FF 00 00 00
10	接收	01:52:42.602.5			0x00000581	数据帧	标准帧	0x08	60 01 14 02 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 16 00 00 00 00 00
12	接收	01:52:43.378.6			0x00000581	数据帧	标准帧	0x08	60 01 16 00 00 00 00 00
13	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 16 01 10 00 AE 30
14	接收	01:52:44.146.8			0x00000581	数据帧	标准帧	0x08	60 01 16 01 00 00 00 00
15	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 16 02 10 00 AF 30
16	接收	01:52:45.103.9			0x00000581	数据帧	标准帧	0x08	60 01 16 02 00 00 00 00
17	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 16 00 02 00 00 00
18	接收	01:52:45.981.5			0x00000581	数据帧	标准帧	0x08	60 01 16 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1400_h, subindex 02_h, value FF_h; transmission type is asynchronous, device-sub-protocol-specific event;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index 1600_h, subindex 00_h, value 00_h; the number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index 1600_h, subindex 01_h, value 60710010_h, mapped to index 6071_h, subindex 00_h; the object is a 16-bit target torque command.

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index 1600_h, subindex 00_h, value 01_h; the number of entries is set at 1.

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Communication parameter setting is: index 1401_h, subindex 02_h, value FF_h; the transmission type is asynchronous, device-sub-protocol-specific event;

S/N 10: Setting succeeds, and answering is correct;

S/N 11: Mapping parameter setting is: index 1601_h, subindex 00_h, value 00_h; the number of entries is reset;

S/N 12: Setting succeeds, and answering is correct;

S/N 13: Mapping parameter setting is: index 1601_h, subindex 01_h, value $30AE0010_h$, mapped to index $30AE_h$, subindex 00_h; the object is 16-bit, basic forward-rotation speed limit under torque mode.

S/N 14: Setting succeeds, and answering is correct;

S/N 15: Mapping parameter setting is: index 1601_h, subindex 01_h, value $30AF0010_h$, mapped to index $30AF_h$, subindex 00_h; the object is 16-bit command for basic backward-rotation speed limit under torque mode;

S/N 16: Setting succeeds, and answering is correct;

S/N 17: Mapping parameter setting is: index 1601_h, subindex 00_h, value 01_h; the number of entries is set at 2;

S/N 18: Setting succeeds, and answering is correct.

TPDO Mapping

In the example below, TPDO1 is mapped to speed feedback; the servo driver will send current actual speed value each time two synchronous objects are received; the CANOPEN node No. is 1.

TPDO2 is mapped to current feedback, status word feedback and control word feedback; the servo driver will send current current value, status word and control word each time two synchronous objects are received; the CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 18 02 02 00 00 00
2	接收	01:58:26.591.9			0x00000581	数据帧	标准帧	0x08	60 00 18 02 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 00 00 00 00
4	接收	01:58:27.396.8			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	23 00 1A 01 20 00 6C 60
6	接收	01:58:28.236.8			0x00000581	数据帧	标准帧	0x08	60 00 1A 01 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 00 1A 00 01 00 00 00
8	接收	01:58:29.010.9			0x00000581	数据帧	标准帧	0x08	60 00 1A 00 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 18 02 02 00 00 00
10	接收	01:58:29.809.8			0x00000581	数据帧	标准帧	0x08	60 01 18 02 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 1A 00 00 00 00 00
12	接收	01:58:31.591.8			0x00000581	数据帧	标准帧	0x08	60 01 1A 00 00 00 00 00
13	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 1A 01 10 00 78 60
14	接收	01:58:32.367.1			0x00000581	数据帧	标准帧	0x08	60 01 1A 01 00 00 00 00
15	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 1A 02 10 00 41 60
16	接收	01:58:33.150.8			0x00000581	数据帧	标准帧	0x08	60 01 1A 02 00 00 00 00
17	发送		成功		0x00000601	数据帧	标准帧	0x08	23 01 1A 03 10 00 40 60
18	接收	01:58:33.966.9			0x00000581	数据帧	标准帧	0x08	60 01 1A 03 00 00 00 00
19	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 01 1A 00 03 00 00 00
20	接收	01:58:34.724.2			0x00000581	数据帧	标准帧	0x08	60 01 1A 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Communication parameter setting is: index 1800_h, subindex 02_h, value 02_h; transmission type is one transmission each time two synchronous objects are received;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 00_h; the number of entries is reset;

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Mapping parameter setting is: index $1A00_h$, subindex 01_h, value $606C0020_h$, mapped to index $606C_h$, subindex 00_h; the object is 32-bit actual speed;

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Mapping parameter setting is: index $1A00_h$, subindex 00_h, value 01_h, the number of entries is set at 1.

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Communication parameter setting is: index 1801_h, subindex 02_h, value 02_h; transmission type is one transmission each time two synchronous objects are received;

S/N 10: Setting succeeds, and answering is correct;

S/N 11: Mapping parameter setting is: index $1A01_h$, subindex 00_h, value 00_h; the number of entries is reset;

S/N 12: Setting succeeds, and answering is correct;

S/N 13: Mapping parameter setting is: index $1A01_h$, subindex 01_h, value 60780010_h, mapped to index 6078_h, subindex 00_h; the object is 16-bit actual current value;

S/N 14: Setting succeeds, and answering is correct;

S/N 15: Mapping parameter setting is: index $1A01_h$, subindex 02_h, value 60410010_h, mapped to index 6041_h, subindex 00_h; the object is a 16-bit status word;

S/N 14: Setting succeeds, and answering is correct;

S/N 13: Mapping parameter setting is: index $1A01_h$, subindex 03_h, value 60400010_h, mapped to index 6040_h, subindex 00_h; the object is a 16-bit control word;

S/N 14: Setting succeeds, and answering is correct;

S/N 13: Mapping parameter setting is: index $1A01_h$, subindex 00_h, value 03_h; the number of entries is set at 3.

S/N 14: Setting succeeds, and answering is correct;

RPDO Commands

In the example below, RPDO1 is mapped to a target torque command, and RPDO2 is mapped to a command for basic forward-rotation speed limit under torque mode and a basic backward-rotation speed limit under torque mode; CANOPEN node No. is 1.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 04 00 00 00
1	接收	02:00:26.617.0			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
2	发送		成功		0x00000301	数据帧	标准帧	0x08	64 00 64 00 00 00 00 00
3	发送		成功		0x00000201	数据帧	标准帧	0x08	F4 01 00 00 00 00 00 00
4	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
5	接收	02:00:31.539.5			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00

Meanings of commands sent:

S/N 0: Operation mode setting is: index 6060_h, subindex 00_h, value 04_h; the driver is set to work under torque control mode;

S/N 1: Setting succeeds, and answering is correct;

S/N 2: RPDO2 is used for setting basic forward-rotation speed limit under torque mode and basic backward-rotation speed limit under torque mode, value of the former is 0064_h, value of the latter is 0064_h, and the unit is RPM (number of turns per minute);

S/N 3: RPDO1 is used for setting target torque, whose value is $01F4_h$, and unit is thousandth of rated torque;

S/N 4: Setting of the motor's enabling is: index 6040_h, subindex 00_h, value $000F_h$;

S/N 5: Setting succeeds, and answering is correct;

TPDO Commands

In the example below, TPDO1 is mapped to speed feedback, the condition for occurrence of feedback event is reception of two synchronous objects; the CANOPEN node No. is 1.

TPDO2 is mapped to feedback of current, status word and Control code; the condition for occurrence of feedback event is reception of two synchronous objects; the CANOPEN node No. is 1.

To feed back the current speed, COB-ID is 181_h; to feed back current current, status word and control word, COB-ID is 281_h.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000080	数据帧	标准帧	0x00	
1	发送		成功		0x00000080	数据帧	标准帧	0x00	
2	接收	02:05:09.432.9			0x00000181	数据帧	标准帧	0x04	26 41 00 00
3	接收	02:05:09.433.1			0x00000281	数据帧	标准帧	0x06	08 00 37 00 0F 00

Meanings of commands sent:

S/N 0: It's synchronous object frame 1;

S/N 1: It's synchronous object frame 2;

S/N 2: TPDO1's return value is 00004126_h, which is the motor's actual speed (unit: puu/s (pulses per second)).

序号 2： **S/N 2:** TPDO2's return value is $000F00370008_h$, in which $000F_h$ is the driver's control word, 0037_h is the driver's status word and 0008_h is the driver's actual current value.

3.6 Homing Mode

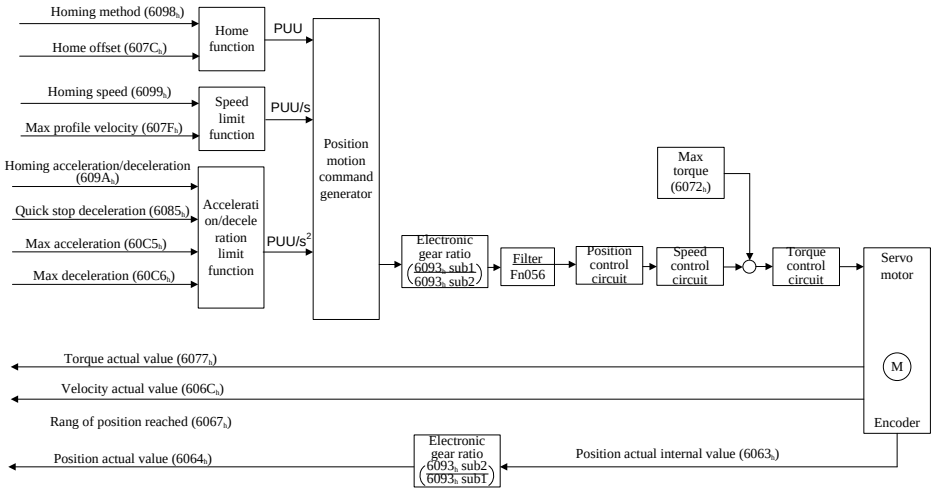
Upon completion of homing, the driver's coordinate system is established, and the driver can execute the position commands sent by the master station. LS, SM and DM series drivers support 37 kinds of homing mode, including home switch, forward/backward position limit, Z pulse, lock rotor, time-out, etc.

Homing needs setting of the following Fn parameters:

Fn01E ($201E_h$-00_h)	**Parameter Name**	**Forward-rotation Prohibited (CCWL) Setting**		
	Setting Range	Setting Unit	Ex-works Setting	Effective Time
	-4~1	——	0	Immediately
	1:internally effective; 0: internally ineffective; -1~ -4: to be determined by digital input signal ports IN 1~4.			
Fn01F ($201F_h$-00_h)	**Parameter Name**	**Backward-rotation Prohibited (CWL) Setting**		
	Setting Range	Setting Unit	Ex-works Setting	Effective Time
	-4~1	——	0	Immediately
	1:internally effective; 0: internally ineffective; -1~ -4: to be determined by digital input signal ports IN 1~4.			
Fn021 (2021_h-00_h)	**Parameter Name**	**Home Switch Setting**		
	Setting Range	Setting Unit	Ex-works Setting	Effective Time
	-4~1	——	0	Immediately
	1:internally effective; 0: internally ineffective; -1~ -4: to be determined by digital input signal ports IN 1~4.			
Fn02F ($202F_h$-00_h)	**Parameter Name**	**Function of Forward-rotation and Backward-rotation Position Limit Check**		
	Setting Range	Setting Unit	Ex-works Setting	Effective Time
	0~1	——	0	Immediately
	0: over-travel (forward-rotation prohibited/backward-rotation prohibited) check is not performed; 1: over-travel (forward-rotation prohibited/backward-rotation prohibited) check is performed			

When the user needs check of forward-rotation and backward-rotation position limit signals under homing mode, he needs to set Fn02F=1 to perform check of overtravel, otherwise, neither forward-rotation signals nor backward-rotation signals would be effective.

3.6.1 Control Block Diagram of Homing Mode



3.6.2 Setting of Objects Related to Homing Mode

Index	Subindex	Name	Accessibility	Mappability	Data Type	Unit
6040_h	00_h	Control Word	RW	YES	Uint16	-
6041_h	00_h	Status Word	RO	TPDO	Uint16	-
6060_h	00_h	Modes of Operation	RW	YES	int8	-
6061_h	00_h	Modes of Operation Display	RO	TPDO	int8	-
$607C_h$	00_h	Home Offset	RW	YES	int32	puu
6098_h	00_h	Homing Method	RW	YES	int8	-
6099_h	01_h	Speed During Search for Switch	RW	YES	Uint32	puu/s
	02_h	Speed During Search for Zero	RW	YES	Uint32	puu/s
$609A_h$	00_h	Homing Acceleration	RW	YES	Uint32	puu/s^2

3.6.3 Example of Operation under Homing Mode

1. Set the mode: 6060_h-00_h = 06_h, homing mode.
2. Set the Home Offset position: $607C_h$-00_h, unit: puu (number of pulses).
3. Set the homing method: 6098_h-00_h = 3; the setting steps are: locate the home switch first, then locate Z pulse, and then move to home offset position.
4. Set the speed for locating the home switch: 6099_h-01_h, unit: puu/s (pulses

per second).

5. Set the speed for locating Z pulse: 6099_h-02_h, unit: puu/s (pulses per second).
6. Set homing acceleration/deceleration: 609Ah-00_h, unit: puu/s^2 (pulses per square second).
7. Set control command: 6040_h-00_h = $0F_h$.
8. Set the command for immediate triggering at an absolute position: 6040h-00_h =$1F_h$.

In the example below, CANOPEN node No. is 1, the control mode makes the motor perform homing movement.

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	发送		成功		0x00000000	数据帧	标准帧	0x02	01 01
1	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 60 60 00 06 00 00 00
2	接收	00:01:52.469.0			0x00000581	数据帧	标准帧	0x08	60 60 60 00 00 00 00 00
3	发送		成功		0x00000601	数据帧	标准帧	0x08	23 7C 60 00 88 13 00 00
4	接收	00:01:53.412.2			0x00000581	数据帧	标准帧	0x08	60 7C 60 00 00 00 00 00
5	发送		成功		0x00000601	数据帧	标准帧	0x08	2F 98 60 00 03 00 00 00
6	接收	00:01:54.698.6			0x00000581	数据帧	标准帧	0x08	60 98 60 00 00 00 00 00
7	发送		成功		0x00000601	数据帧	标准帧	0x08	23 99 60 01 E8 03 00 00
8	接收	00:01:55.714.1			0x00000581	数据帧	标准帧	0x08	60 99 60 01 00 00 00 00
9	发送		成功		0x00000601	数据帧	标准帧	0x08	23 99 60 02 F4 01 00 00
10	接收	00:01:56.528.9			0x00000581	数据帧	标准帧	0x08	60 99 60 02 00 00 00 00
11	发送		成功		0x00000601	数据帧	标准帧	0x08	23 9A 60 00 64 00 00 00
12	接收	00:01:57.382.2			0x00000581	数据帧	标准帧	0x08	60 9A 60 00 00 00 00 00
13	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 0F 00 00 00
14	接收	00:01:58.198.3			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00
15	发送		成功		0x00000601	数据帧	标准帧	0x08	2B 40 60 00 1F 00 00 00
16	接收	00:01:59.124.3			0x00000581	数据帧	标准帧	0x08	60 40 60 00 00 00 00 00

Meanings of commands sent:

S/N 0: Start up;

S/N 1: Operation mode setting is: index 6060_h, subindex 00_h, value 06_h; the driver is set to work under homing mode;

S/N 2: Setting succeeds, and answering is correct;

S/N 3: Home offset value setting is: index $607C_h$, subindex 00_h, value 00001388_h; the unit is puu (number of pulses);

S/N 4: Setting succeeds, and answering is correct;

S/N 5: Homing method setting is: index 6098_h, subindex 00_h, value 0003_h;

S/N 6: Setting succeeds, and answering is correct;

S/N 7: Setting of the speed of locating home switch (i.e. Speed During Search for Switch) is: index 6099_h, subindex 01_h, value $000003E8_h$; the unit is puu/s (pulses per second);

S/N 8: Setting succeeds, and answering is correct;

S/N 9: Setting of the speed for finding Z pulse (i.e. Speed During Search for Zero) is: index 6099_h, subindex 02_h, value $000001F4_h$, the unit is puu/s (pulses per second);

S/N 10: Setting succeeds, and answering is correct;

S/N 11: Homing acceleration and homing deceleration setting is: index $609A_h$, subindex 00_h, value 00000064_h, the unit is puu/s^2 (pulses per square second);

S/N 12: Setting succeeds, and answering is correct;

S/N 13: Setting of control command to control enabling of the driver is: index 6040_h, subindex 00h, value 000Fh; the motor is enabled;

S/N 14: Setting succeeds, and answering is correct;

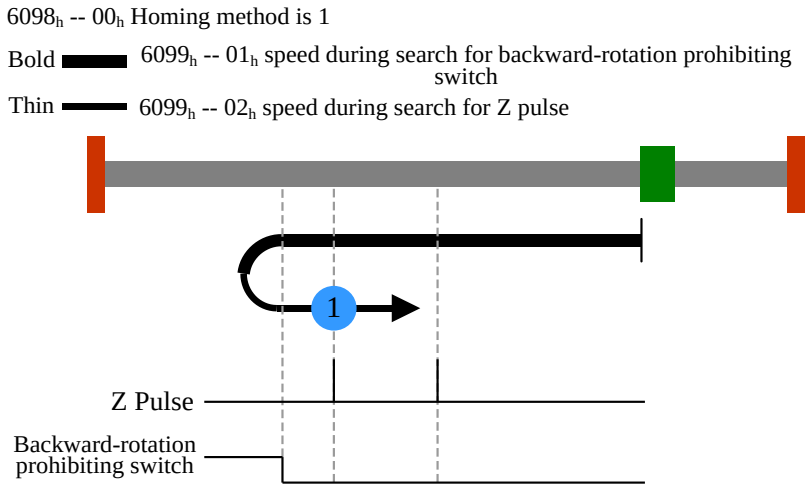
S/N 15: Setting of command to be triggered on the positive edge and to make the motor perform homing operation is: index 6040_h, subindex 00_h, value $001F_h$;

S/N 16: Setting succeeds, and answering is correct.

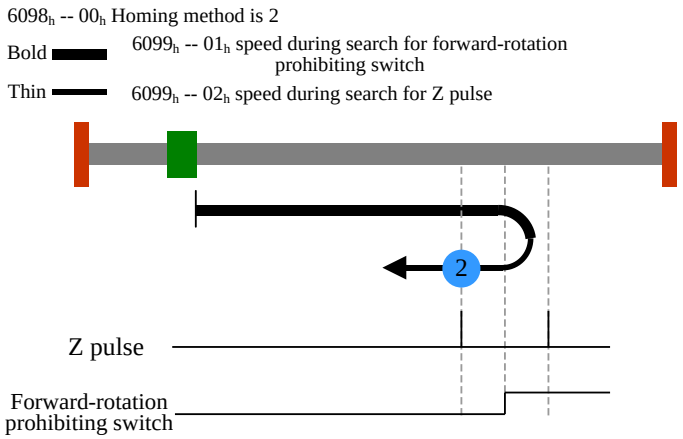
3.6.4 Detailed Explanation of Homing Mode

LS, SM and DM series drivers provide 30 kinds of homing mode, including home switch, forward and backward-rotation position limit, Z pulse, etc., the user can select one as per his needs to complete the operation of homing (i.e. finding the home).

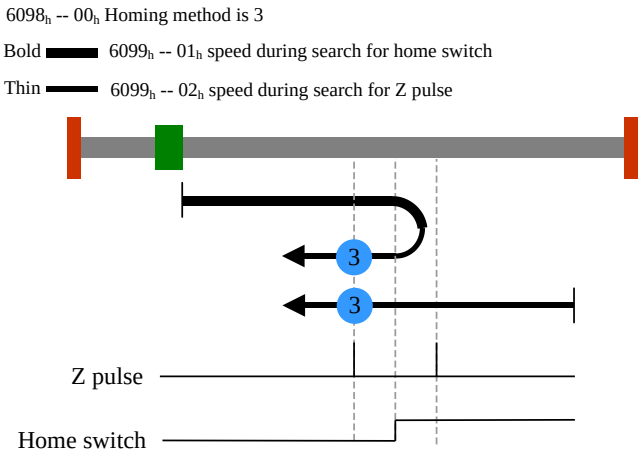
3.6.4.1 Homing upon Touching Backward-rotation Prohibiting Switch and Z Pulse



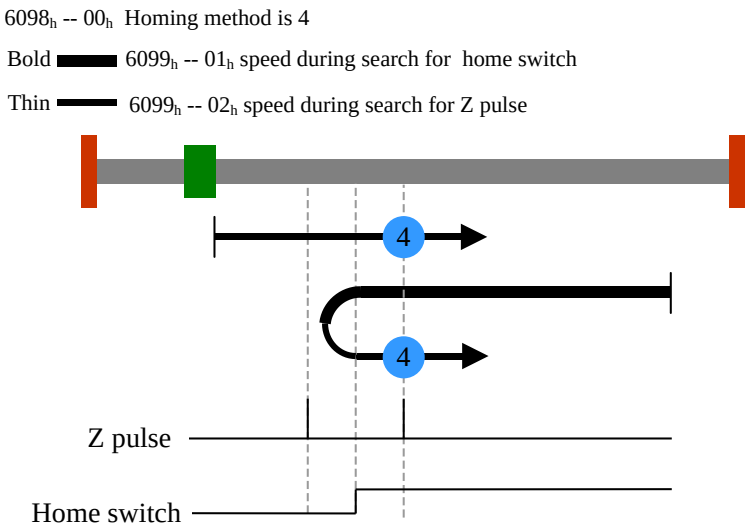
3.6.4.2 Homing upon Touching Forward-rotation Prohibiting Switch and Z Pulse



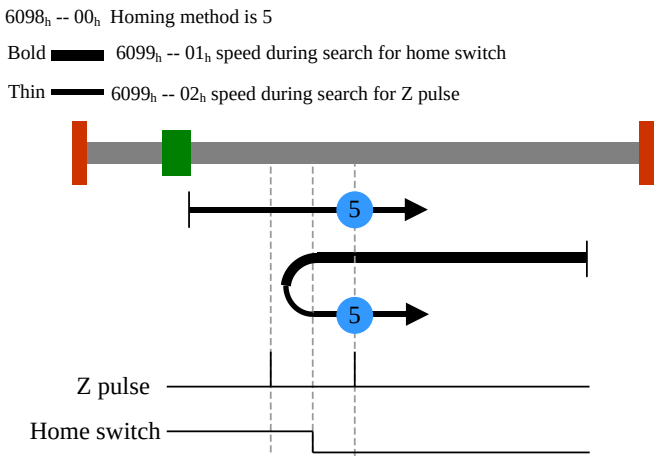
3.6.4.3 Backward-rotating Homing upon Touching Home Switch's Upper Edge and Z Pulse



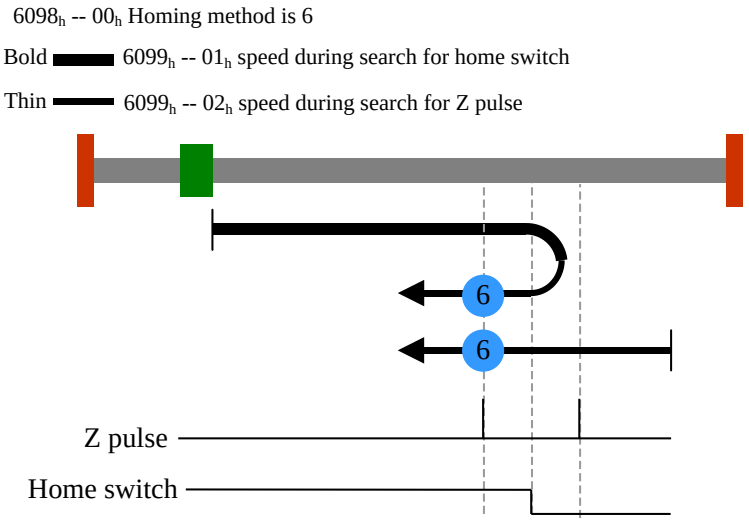
3.6.4.4 Forward-rotating Homing upon Touching Home Switch's Upper Edge and Z Pulse



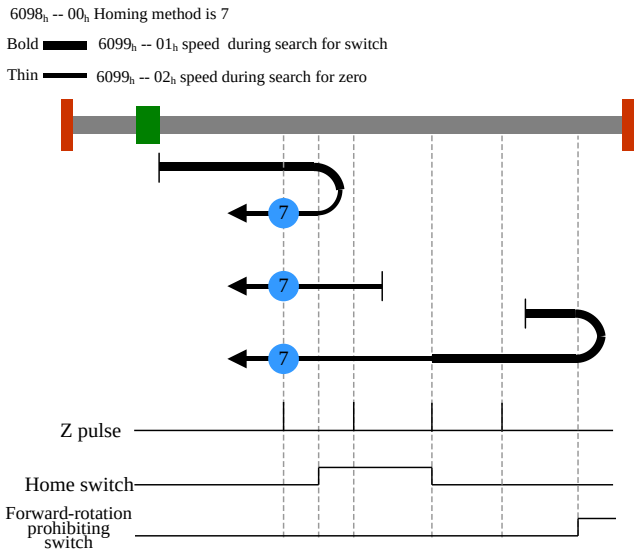
3.6.4.5 Forward-rotating Homing upon Touching Home Switch's Lower Edge and Z Pulse



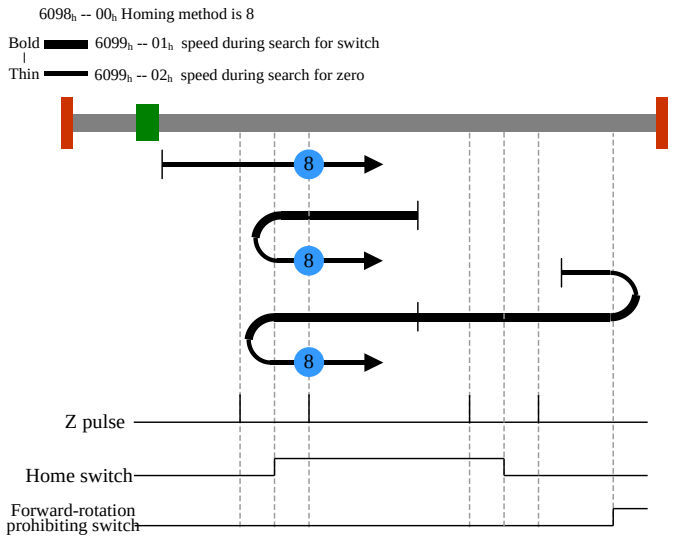
3.6.4.6 Backward-rotating Homing upon Touching Home Switch's Lower Edge and Z Pulse



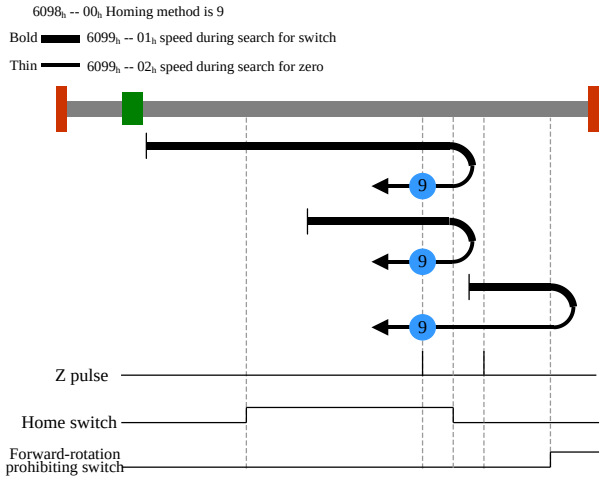
3.6.4.7 Backward-rotating Homing upon Touching Home Switch's Upper Edge, Forward-rotation Prohibiting Switch and Z Pulse



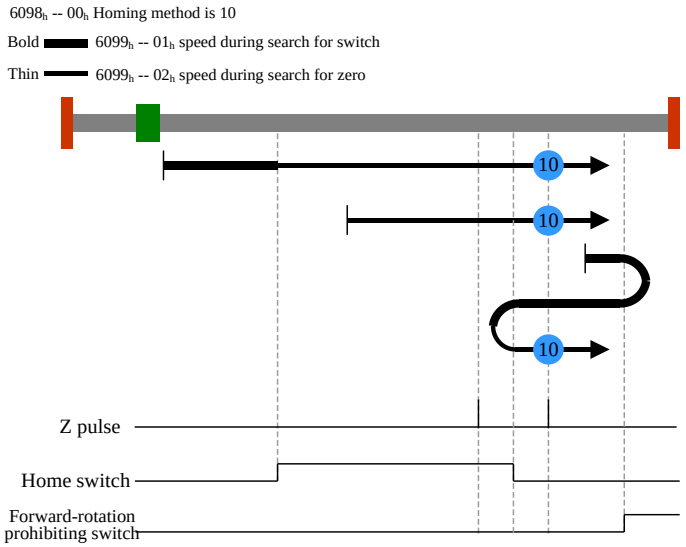
3.6.4.8 Forward-rotating Homing Upon Touching Home Switch's Upper Edge, Forward-rotation Prohibiting Switch and Z Pulse



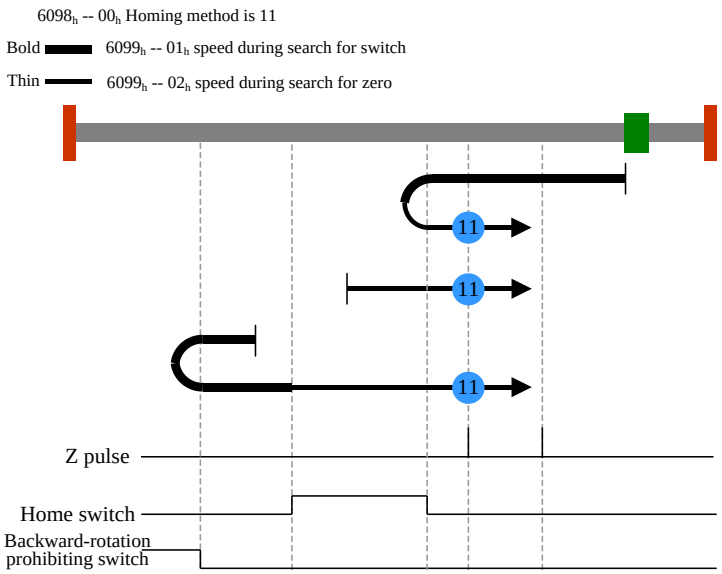
3.6.4.9 Backward-rotating Homing upon Touching Home Switch's Lower Edge, Forward-rotation Prohibiting Switch and Z Pulse



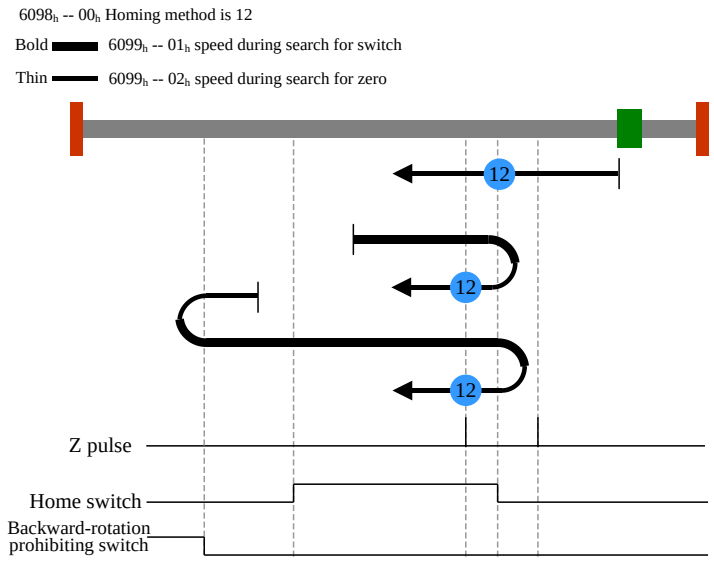
3.6.4.10 Forward-rotating Homing upon Touching Home Switch's Lower Edge, Forward-rotation Prohibiting Switch and Z Pulse



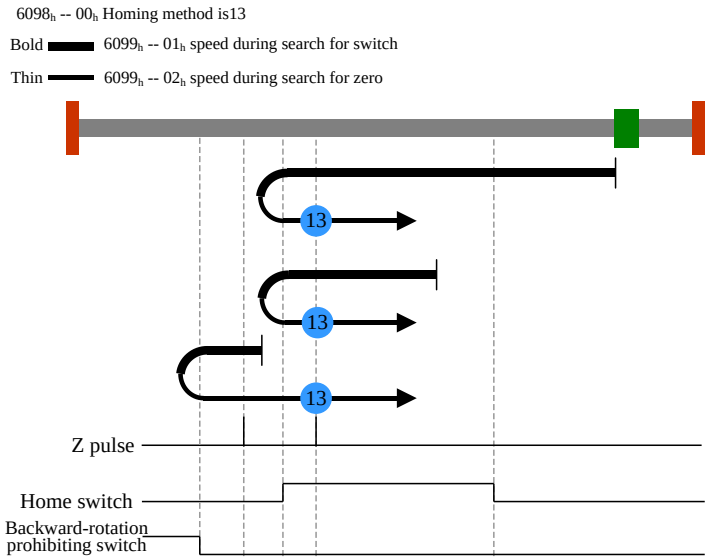
3.6.4.11 Forward-rotating Homing upon Touching Home Switch's Upper Edge, Backward-rotation Prohibiting Switch and Z Pulse



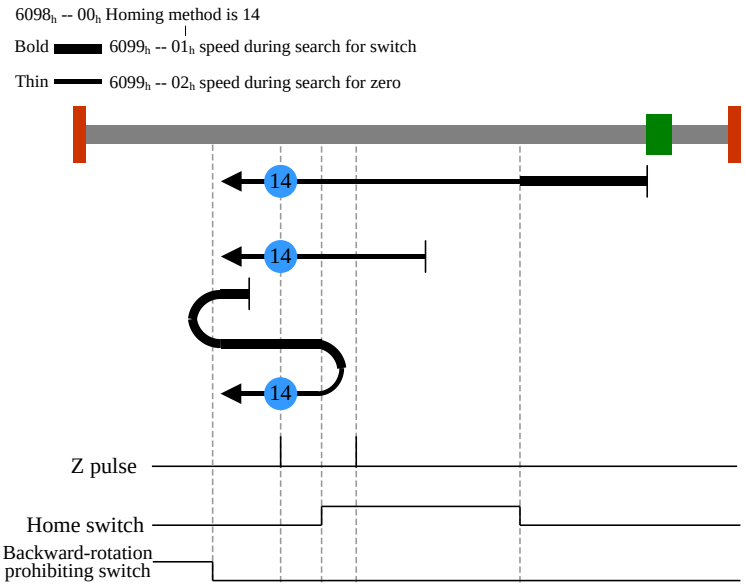
3.6.4.12 Backward-rotating Homing Upon Touching Home Switch's Lower Edge, Backward-rotation Prohibiting Switch and Z Pulse



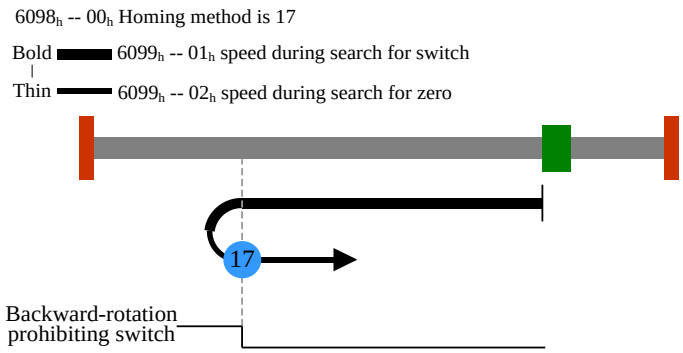
3.6.4.13 Forward-rotating Homing upon Touching Home Switch's Upper Edge, Backward-rotation Prohibiting Switch and Z Pulse



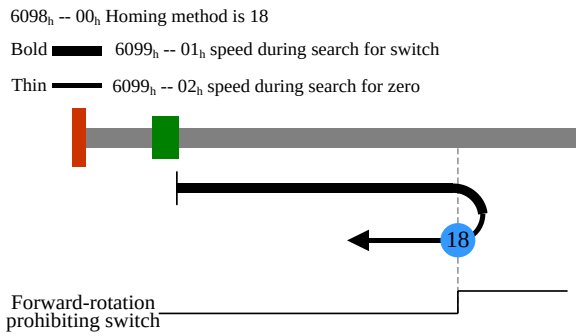
3.6.4.14 Backward-rotating Homing Upon Touching Home Switch's Upper Edge, Backward-rotation Prohibiting Switch and Z Pulse



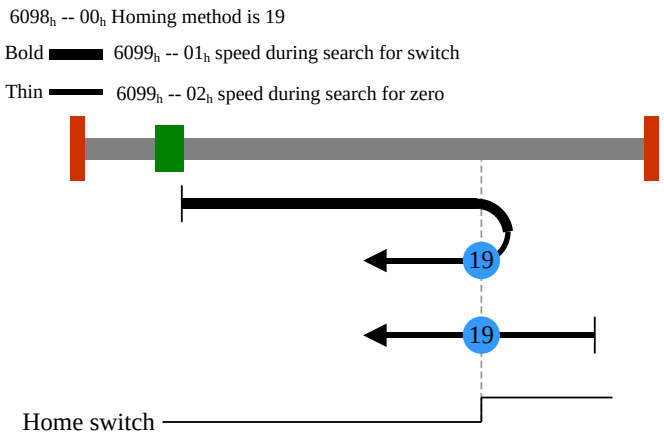
3.6.4.15 Homing Upon Touching Backward-rotation Prohibiting Switch



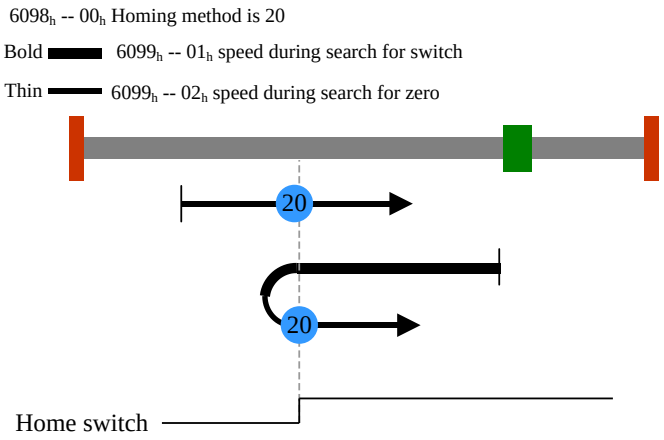
3.6.4.16 Homing upon Touching Forward-rotation Prohibiting Switch



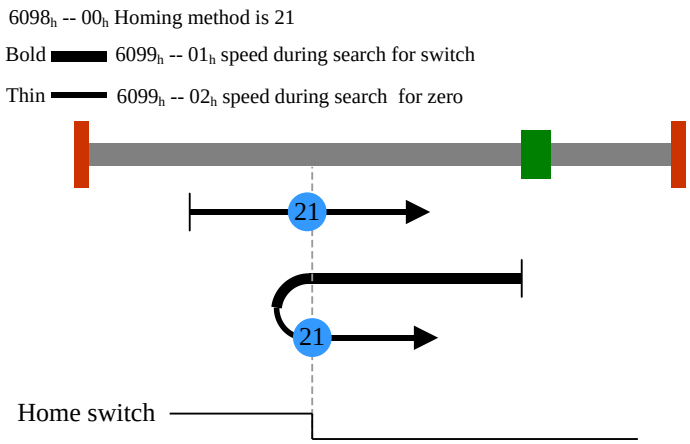
3.6.4.17 Backward-rotating Homing upon Touching Home Switch's Upper Edge



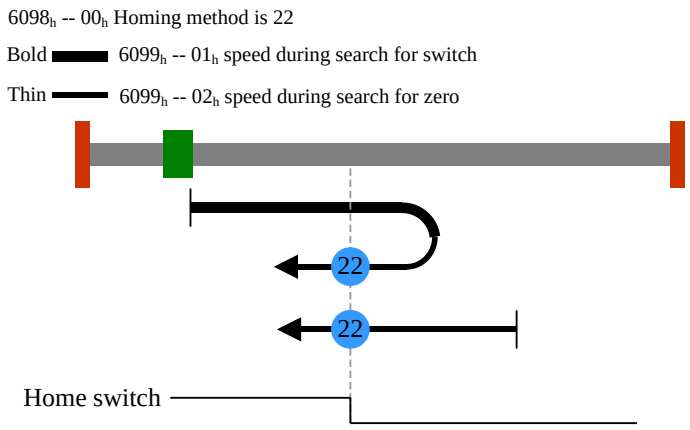
3.6.4.18 Forward-rotating Homing upon Touching Home Switch's Upper Edge



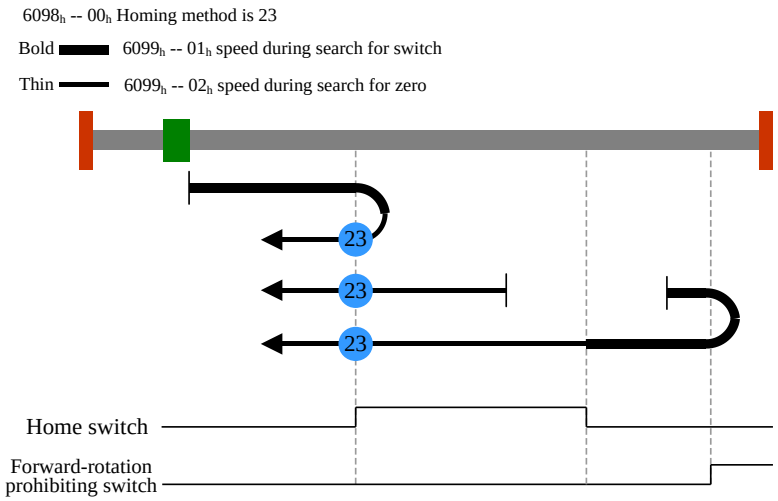
3.6.4.19 Forward-rotating Homing upon Touching Home Switch's Lower Edge



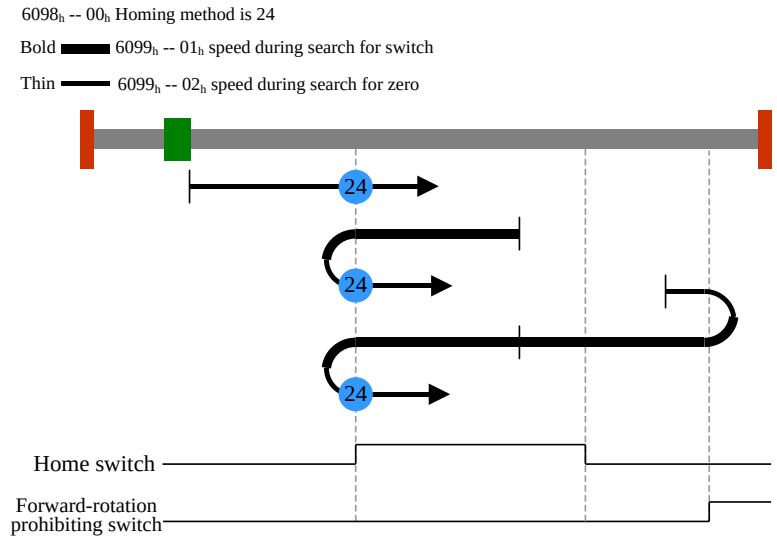
3.6.4.20 Backward-rotating Homing upon Touching Home Switch's Lower Edge



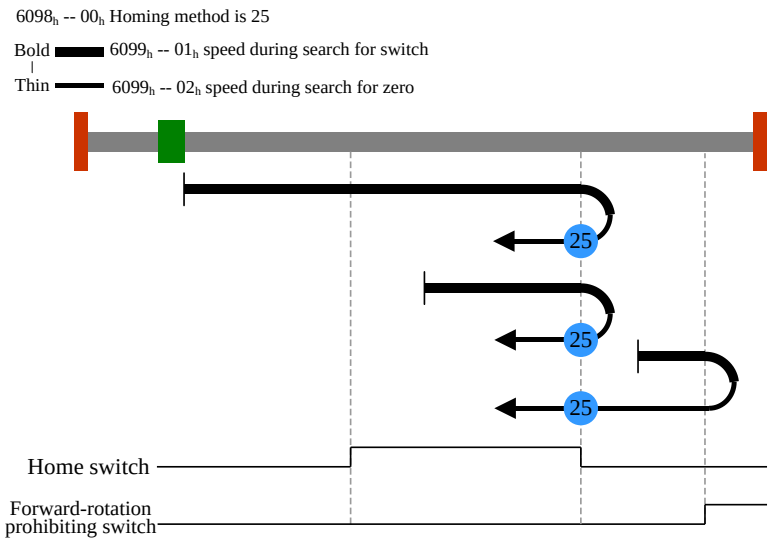
3.6.4.21 **Backward-rotating Homing upon Touching Home Switch's Upper Edge and Forward-rotation Prohibiting Switch**



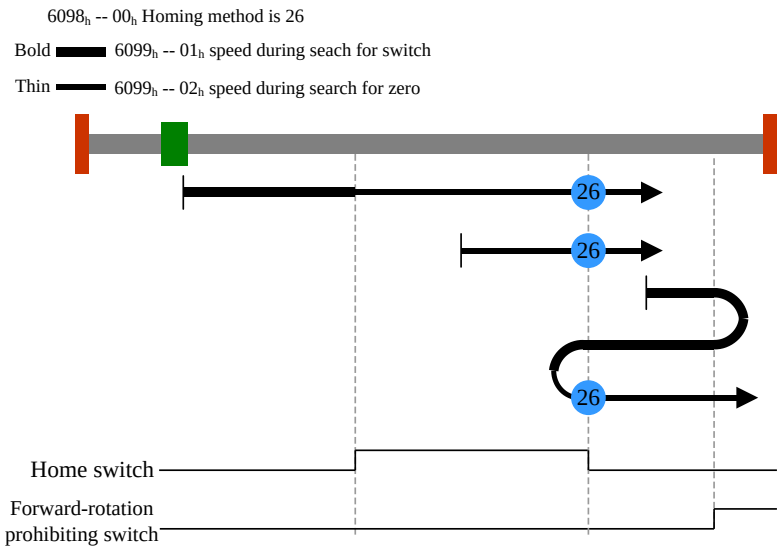
3.6.4.22 **Forward-rotating Homing Upon Touching Home Switch's Upper Edge and Forward-rotation Prohibiting Switch**



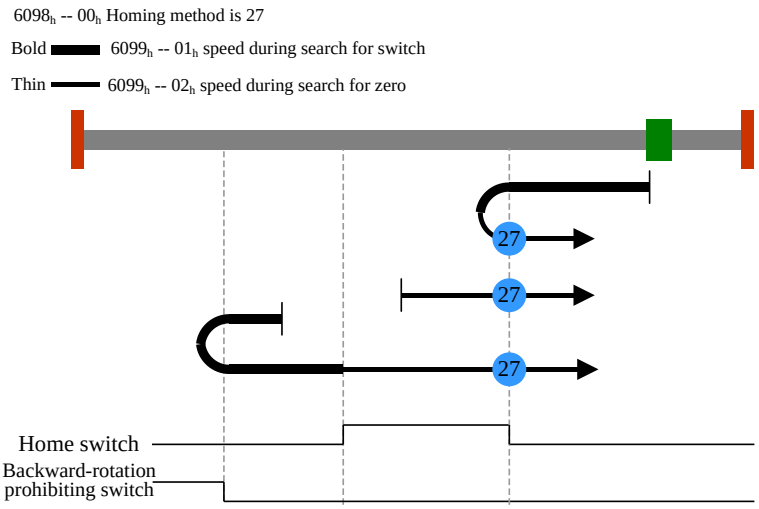
3.6.4.23 Backward-rotating Homing upon Touching Home Switch's Lower Edge and Forward-rotation Prohibiting Switch



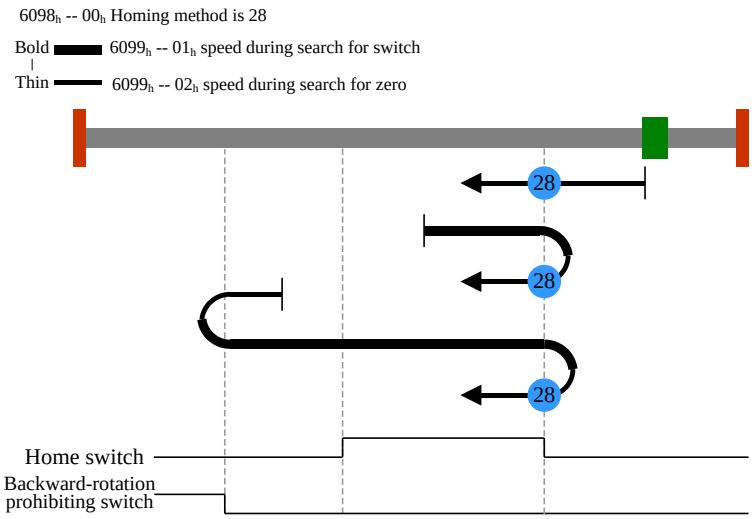
3.6.4.24 Forward-rotating Homing upon Touching Home Switch's Lower Edge and Forward-rotation Prohibiting Switch



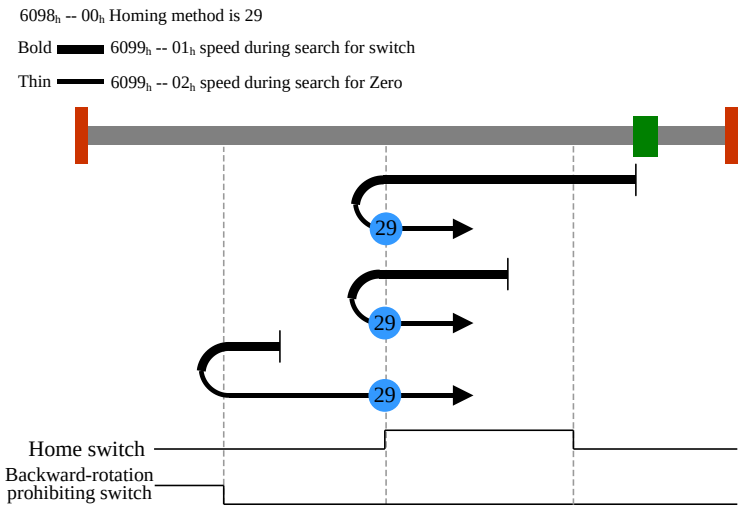
3.6.4.25 Forward-rotating Homing upon Touching Home Switch's Lower Edge and Backward-rotation Prohibiting Switch



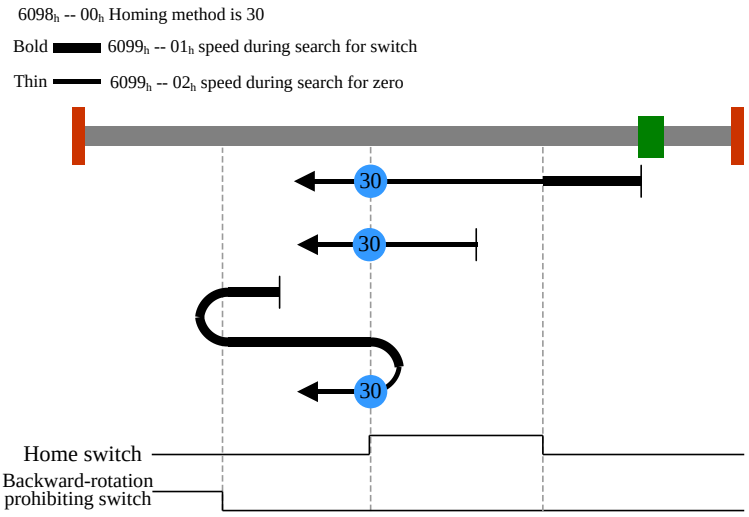
3.6.4.26 Backward-rotating Homing upon Touching Home Switch's Lower Edge and Backward-rotation Prohibiting Switch



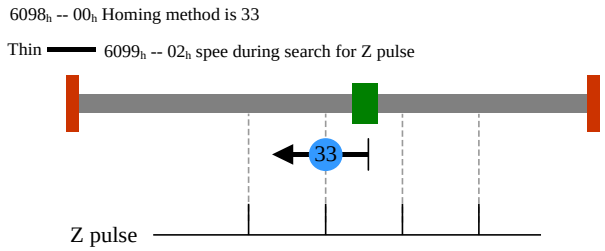
3.6.4.27 Forward-rotating Homing upon Touching Home Switch's Upper Edge and Backward-rotation Prohibiting Switch



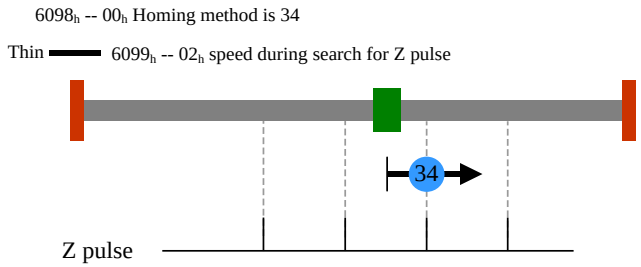
3.6.4.28 Backward-rotating Homing upon Touching Home Switch's Upper Edge and Backward-rotation Prohibiting Switch



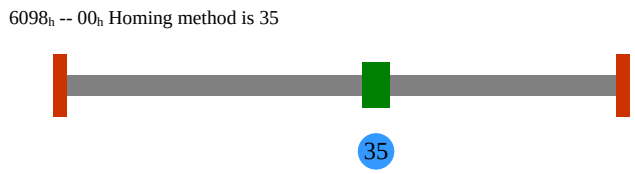
3.6.4.29 Backward-rotating Homing upon Touching Z Pulse



3.6.4.30 Forward-rotating Homing upon Touching Z Pulse



3.6.4.31 Current Position is Home



3.6.5 Customization Function under Homing Mode

In addition to these home locating methods, HollySys also designs home locating methods to be used during rotor lock and time-out. When the forward rotation prohibiting and backward rotation prohibiting switches are unable to function, but the user needs forward rotation prohibition and backward rotation prohibition as the condition for home locating, he can use rotor lock or time-out as virtual forward rotation prohibiting and backward rotation prohibiting switches

by setting the parameters below, so that home can be located.

Servo driver parameter Fn1A8 is used to activate the function of rotor lock serving as virtual position limit switch, Fn1A7 is used to set the rotor lock time, and Fn1AE is used to set the length of time-out to help time-out play its role as a virtual position limit switch. The 3 parameters are operational only under CANOPEN mode.

Parameter No.	Index	Parameter Description	Ex-works Setting
Fn1A7	$21A7_h$-00_h	It sets the rotor lock time for rotor lock's function as a virtual position limit switch. Unit: 2ms	0
Fn1A8	$21A7_h$-00_h	It is used to define whether to use rotor lock function as a virtual position limit switch. 999: Rotor lock is allowed to be used as a virtual position limit switch; Other values: Rotor lock is not allowed to be used as a virtual position limit switch;	0
Fn1AA	$21AA_h$-00_h	It sets duration of the signal indicating completion of homing. Unit: ms	-
Fn1AC	$21AC_h$-00_h	Torque limit of homing. Unit: 0.001 X rated torque value	-
Fn1AD	$21AD_h$-00_h	Error of homing. Unit: pulse	-
Fn1AE	$21A7_h$-00_h	It is used to define whether to use time-out function as a virtual position limit switch. >0: Time-out function is allowed to be used as a virtual position limit switch, and the parameter's value is length of time out (unit: ms); =0: Time-out function is not allowed to be used as a virtual position limit switch;	0

Chapter IV Object Dictionary

4.1 Classification of Objects

Each object dictionary can be operated through SDO. Writable and readable objects can be written in and read through SDO, and read-only objects can only be read through SDO. All objects support PDO mapping, and can be configured as RPDO or TPDO as per their accessibility.

Index (hexadecimal system)	Object
1000_h – $1FFF_h$	Communication Profile Area
2000_h – $2FFF_h$	Manufacturer specific Profile Area
6000_h – $6FFF_h$	Standardized Device Profile Area
$A000_h$ –$FFFF_h$	Reserved

Index: An index is shown in hexadecimal system, and defines corresponding object's position in the object dictionary.

The table below shows Data Type:

Data Type	Value Range	Data Length	DS301's Value
int8	-128~+127	1 Byte	0002
int16	-32768~+32767	2 Byte	0003
int32	-2147483648~+2147483647	4 Byte	0004
Uint8	0~256	1 Byte	0005
Uint16	0~65535	2 Byte	0006
Uint32	0~4294967295	4 Byte	0007
string	ASCII	——	0009

The table below shows value/readability/writability classification:

Value/Readability/Writability Classification	Description
RW	Readable and writable
WO	Write-only
RO	Read-only
CONST	Constant value, read-only

The table below shows types of objects.

Object Type	Description	DS301's Value
VAR	Single simple value, including data types such as int8, Uint16 and String	7
ARR	Data block of the same type	8
REC	Data block of the same type	9

4.2 Assignment List of Object Group 1000h

Object group 1000_h is made up of the parameters needed for CANopen communication. Communication parameters don't support PDO mapping.

Index	Subindex	Name	Accessibility	Data Type	Unit	Mappability	Object Type	Ex-works Setting
1000_h	-	Device Type	RO	Uint32	-	NO	VAR	20192_h
1001_h	-	Error Register	RO	Uint8	-	NO	VAR	-
1003_h	-	Pre-defined Error Field	RW	Uint8	-	NO	ARR	-
	$1\sim4_h$	Error field	RO	Uint32	-	NO		-
1004_h	-	Auto Operation: After 21930 is written in, the driver that is turned on can automatically enter operational state	RW	Uint16	-	NO	VAR	$55AA_h$
1005_h	-	COB-ID SYNC Message	RO	Uint32	-	NO	VAR	80_h
1006_h	-	Synchronous Cycle Period	RW	Uint32	us	NO	VAR	-
1008_h	-	Vendor-ID	RO	Uint32	-	NO	VAR	$4C532020_h$
1009_h	-	Manufacturer Hardware Version	RO	Uint32	-	NO	VAR	20081201_h
$100A_h$	-	Manufacturer Software Version	RO	Uint32	-	NO	VAR	20081201_h
1010_h	-	Save the Parameter	RO	Uint8	-	NO	ARR	4
	1_h	Save all Parameters	RW	Uint32	-	NO		-
	2_h	Save Communication Parameters	RW	Uint32	-	NO		-
	3_h	Save Application Parameters	RW	Uint32	-	NO		-
	4_h	Save Manufacturer-defined Parameters	RW	Uint32	-	NO		-
1011_h	-	Restore Default Parameters	RO	Uint8	-	NO	ARR	4
	1_h	Restore All Objects' Default Parameters	RW	Uint32	-	NO		-
	2_h	Restore Communication Objects' Default Parameters	RW	Uint32	-	NO		-
	3_h	Restore Application Objects' Default Parameters	RW	Uint32	-	NO		-
	4_h	Restore Manufacturer's Default Parameters	RW	Uint32	-	NO		-
1014_h	-	Emergency Message's	RO	Uint32	-	NO	ARR	80_h+CAN-ID

		COB-ID						
1015_h	-	Inhibit Time EMCY	RW	Uint16	100us	NO	ARR	$0A_h$
1016_h	-	Consumer Heartbeat Time	RO	Uint8	-	NO	ARR	1
	1_h	Consumer Heartbeat Time	RW	Uint32	ms	NO		-
1017_h	-	Producer Heartbeat Time	RW	Uint16	ms	NO	VAR	-
1018_h	-	Identity Object	RO	Uint8	-	NO	REC	4
	1_h	Vendor-ID	RO	Uint32	-	NO		$00000AB6_h$
	2_h	Product Code	RO	Uint32	-	NO		$4C532020_h$
	3_h	Revision Number	RO	Uint32	-	NO		00000004_h
	4_h	Error Behavior	RO	Uint32	-	NO		00000001_h
1200_h	-	SDO Server Parameter	RO	Uint8	-	NO	REC	2
	1_h	Client to service COB-ID	RO	Uint32	-	NO		600_h+CAN-ID
	2_h	Server to customer COB-ID	RO	Uint32	-	NO		580_h+CAN-ID
1400_h	-	RPDO1 Communication Parameter	RO	Uint8	-	NO	REC	2
	1_h	RPDO1's COB-ID	RO	Uint32	-	NO		200_h+CAN-ID
	2_h	RPDO1's Transmission Type	RW	Uint8	-	NO		FF_h
1401_h	-	RPDO2 Communication Parameter	RO	Uint8	-	NO	REC	2
	1	RPDO2's COB-ID	RO	Uint32	-	NO		300_h+CAN-ID
	2	RPDO2's Transmission Type	RW	Uint8	-	NO		FF_h
1402_h	-	RPDO3 Communication Parameter	RO	Uint8	-	NO	REC	2
	1_h	RPDO3's COB-ID	RO	Uint32	-	NO		400_h+CAN-ID
	2_h	RPDO3's Transmission Type	RW	Uint8	-	NO		FF_h
1403_h	-	RPDO4 Communication Parameter	RO	Uint8	-	NO	REC	2
	1_h	RPDO4's COB-ID	RO	Uint32	-	NO		500_h+CAN-ID
	2_h	RPDO4's Transmission Type	RW	Uint8	-	NO		FF_h
1600_h	-	RPDO1's Mapping Parameter	RW	Uint8	-	NO	REC	-
	$1~4_h$	RPDO1's Application Object	RW	Uint32	-	NO		-

1601_h	-	RPDO2's Mapping Parameter	RW	Uint8	-	NO	REC	-
	$1\sim4_h$	RPDO2's Application Object	RW	Uint32	-	NO		-
1602_h	-	RPDO3's Mapping Parameter	RW	Uint8	-	NO	REC	-
	$1\sim4_h$	RPDO3's Application Object	RW	Uint32	-	NO		-
1603_h	-	RPDO4's Mapping Parameter	RW	Uint8	-	NO	REC	-
	$1\sim4_h$	RPDO4's Application Object	RW	Uint32	-	NO		-
1800_h	-	TPDO1 Communication Parameter	RO	Uint8	-	NO	REC	5_h
	1_h	TPDO1's COB-ID	RW	Uint32	-	NO		180_h+CAN-ID
	2_h	TPDO1's Transmission Type	RW	Uint8	-	NO		FF_h
	3_h	Inhibit Time	RW	Uint16	0.1ms	NO		32_h
	5_h	Event Timer	RW	Uint16	ms	NO		32_h
1801_h	-	TPDO2 Communication Parameter	RO	Uint8	-	NO	REC	5_h
	1_h	TPDO2's COB-ID	RW	Uint32	-	NO		280_h+CAN-ID
	2_h	TPDO2's Transmission Type	RW	Uint8	-	NO		FF_h
	3_h	Inhibit Time	RW	Uint16	0.1ms	NO		32_h
	5_h	Event Timer	RW	Uint16	ms	NO		32_h
1802_h	-	TPDO3 Communication Parameter	RO	Uint8	-	NO	REC	5_h
	1_h	TPDO3's COB-ID	RW	Uint32	-	NO		380_h+CAN-ID
	2_h	TPDO3's Transmission Type	RW	Uint8	-	NO		FF_h
	3_h	Inhibit Time	RW	Uint16	0.1ms	NO		32_h
	5_h	Event Timer	RW	Uint16	ms	NO		32_h
1803_h	-	TPDO4 Communication Parameter	RO	Uint8	-	NO	REC	5_h
	1_h	TPDO4's COB-ID	RW	Uint32	-	NO		480_h+CAN-ID
	2_h	TPDO4's Transmission Type	RW	Uint8	-	NO		FF_h
	3_h	Inhibit Time	RW	Uint16	0.1ms	NO		32_h
	5_h	Event Timer	RW	Uint16	ms	NO		32_h
$1A00_h$	-	TPDO1's Mapping Parameter	RW	Uint8	-	NO	REC	-
	$1\sim4_h$	TPDO1's Application Object	RW	Uint32	-	NO		-
$1A01_h$	-	TPDO2's Mapping	RW	Uint8	-	NO	REC	-

		Parameter						
	1~4$_h$	TPDO2's Application Object	RW	Uint32	-	NO		-
1A02$_h$	-	TPDO3's Mapping Parameter	RW	Uint8	-	NO	REC	-
	1$_h$	TPDO3's Application Object	RW	Uint32	-	NO		-
1A03$_h$	-	TPDO4's Mapping Parameter	RW	Uint8	-	NO	REC	-
	1~4$_h$	TPDO4's Application Object	RW	Uint32	-	NO		-

4.3 Assignment List of Object Group 2000h

Object group 2000$_h$ is an object list defined by Beijing HollySys Electric Technology Co., Ltd. The objects in it correspond to applicable machines' function codes. All the objects in it support PDO mapping.

Function parameter Fns' offset addresses are 2000$_h$ and 3000$_h$, the former is used for writing to be saved, the latter is for writing not to be saved.

Status parameter Dns' offset address is 4000$_h$.

Interval	Starting address	Max. length	Remarks
Parameter Region	2000$_h$	512	It's for servo operation parameter register's Fn parameters. Written values are saved after reenergization.
	3000$_h$	512	It's for servo operation parameter register's Fn parameters. Written values are not saved after reenergization.
Status Region	4000$_h$	256	It's for servo operation status register's Dn parameters.

For detailed description of the parameters, please refer to *LS- and DM-Series Low-voltage Servo Driver Manual*.

4.4 Assignment List of Object Group 6000h

Object group 6000$_h$ is made up of objects related to protocol DSP402 that are supported.

Index	Subindex	Name	Accessibility	Data Type	Unit	Mappability	Object Type	Ex-works Setting
6007$_h$	-	Abort Connection Option Code	RW	int16	-	NO	VAR	-

$603F_h$	-	Error Code	RO	Uint16	-	TPDO	VAR	-
6040_h	-	Control Word	RW	Uint16	-	RPDO	VAR	-
6041_h	-	Status Word	RO	Uint16	-	TPDO	VAR	-
$605A_h$	-	Quick Stop Method Selection	RW	int16	-	NO	VAR	-
$605B_h$	-	Shutdown Option	RW	int16	-	NO	VAR	-
$605C_h$	-	Disable Operation Option	RW	int16	-	NO	VAR	-
$605D_h$	-	Suspension Option	RW	int16	-	NO	VAR	-
$605E_h$	-	Fault Response Option	RW	int16	-	NO	VAR	-
6060_h	-	Mode of Operation	RW	int8	-	YES	VAR	-
6061_h	-	Mode of Operation Display	RO	int8	-	TPDO	VAR	-
6062_h	-	User Position Command	RO	int32	puu	TPDO	VAR	-
6063_h	-	Position Actual Internal Value	RO	int32	puu	TPDO	VAR	-
6064_h	-	Position Actual Value	RO	int32	puu	TPDO	VAR	-
6065_h	-	Offset Set Value for Position Overtravel	RW	Uint32	puu	NO	VAR	-
6066_h	-	Position Overtravel Check Time	RW	Uint16	ms	NO	VAR	-
6067_h	-	Range of Position Reached	RW	Uint32	puu	NO	VAR	-
6068_h	-	Time of Range of Position Reached	RW	Uint16	ms	NO	VAR	-
6069_h	-	Velocity Sensor Actual Value	RO	int32	puu	TPDO	VAR	-
$606A_h$	-	Sensor Selection Code	RW	int16	-	NO	VAR	-
$606B_h$	-	Velocity Demand Value	RO	int32	puu/s	TPDO	VAR	-
$606C_h$	-	Velocity Actual Value	RO	int32	puu/s	TPDO	VAR	-
$606D_h$	-	Range of Speed Reached	RW	Uint16	puu/s	NO	VAR	-
$606E_h$	-	Time of Range of Speed Reached	RW	Uint16	ms	NO	VAR	-
$606F_h$	-	Zero Speed Threshold	RW	Uint16	puu/s	NO	VAR	-
6070_h	-	Zero Speed Time Window	RW	Uint16	ms	NO	VAR	-
6071_h	-	Target Torque	RW	int16	‰Tn	YES	VAR	-
6072_h	-	Max Torque	RW	Uint16	‰Tn	YES	VAR	-
6073_h	-	Max Current	RW	Uint16	‰RC	YES	VAR	-
6074_h	-	Torque Demand Value	RO	int16	‰Tn	TPDO	VAR	-
6075_h	-	Motor Rated Current	RW	Uint32	mA	NO	VAR	-
6076_h	-	Motor Rated Torque	RW	Uint32	0.001	NO	VAR	-

					N.m			
6077_h	-	Torque Actual Value	RO	int16	‰Tn	TPDO	VAR	-
6078_h	-	Current Actual Value	RO	int16	mA	TPDO	VAR	-
6079_h	-	DC Link Circuit Voltage	RO	Uint32	mV	TPDO	VAR	-
$607A_h$	-	Target Position	RW	int32		YES	VAR	-
$607B_h$	-	Position Range Limit	RW	Uint8	-	NO	ARR	-
	1_h	Min Position Range Limit	RW	int32		YES		-
	2_h	Max Position Range Limit	RW	int32		YES		-
$607C_h$	-	Home Offset	RW	int32		NO	VAR	-
$607D_h$	-	Software Absolute Position Limit	RO	Uint8	-	NO	ARR	-
	1_h	Min Software Absolute Position Limit	RW	int32		YES		-
	2_h	Max Software Absolute Position Limit	RW	int32		YES		-
$607E_h$	-	Polarity	RW	Uint8	-	YES	VAR	-
$607F_h$	-	Max Profile Velocity	RW	Uint32	puu/s	NO	VAR	-
6080_h	-	Max Motor Velocity	RW	Uint32	RPM	NO	VAR	-
6081_h	-	Profile Velocity	RW	Uint32	puu/s	YES	VAR	-
6082_h	-	Velocity at Reaching Target Position	RW	Uint32	puu/s	YES	VAR	-
6083_h	-	Profile Acceleration	RW	Uint32	puu/s^2	YES	VAR	-
6084_h	-	Profile Deceleration	RW	Uint32	puu/s^2	YES	VAR	-
6085_h	-	Quick Stop Deceleration	RW	Uint32	puu/s^2	YES	VAR	-
6086_h	-	Motion Profile Type	RW	int16	-	NO	VAR	0
6087_h	-	Torque Slope	RW	Uint32	‰Tn/S	YES	VAR	-
6088_h	-	Torque Type	RW	int16	-	YES	VAR	-
$608F_h$	-	Position Encoder Resolution	RO	Uint8	-	NO	ARR	2
	1_h	Encoder Increments	RW	Uint32	-	NO		-
	2_h	Motor Velocity	RW	Uint32	-	NO		-
6090_h	-	Velocity Encoder Resolution	RO	Uint8	-	NO	ARR	2
	1_h	Encoder Increments per Second	RW	Uint32	-	NO		-
	2_h	Motor Revolutions per Second	RW	Uint32	-	NO		-
6091_h	-	Drive Ratio	RO	Uint8	-	NO	ARR	2
	1_h	Motor Velocity	RW	Uint32	-	NO		-
	2_h	Output Shaft Velocity	RW	Uint32	-	NO		-
6092_h	-	Feed Constant	RO	Uint8	-	NO	ARR	2
	1_h	Feed	RW	Uint32	-	NO		-

	2_h	Drive Shaft Velocity	RW	Uint32	-	NO		-
*6093_h	-	Position Factor	RO	Uint8		NO	ARR	2
	1_h	Position Factor's Numerator	RW	Uint32	-	NO		-
	2_h	Position Factor's Divisor	RW	Uint32	-	NO		-
*6094_h	-	Velocity Encoder Factor	RO	Uint8	-	NO	ARR	2
	1_h	Velocity Encoder Factor's Numerator	RW	Uint32	-	NO		-
	2_h	Velocity Encoder Factor's Divisor	RW	Uint32	-	NO		-
6095_h	-	Velocity Factor 1	RO	Uint8	-	NO	ARR	2
	1_h	Velocity Factor 1's Numerator	RW	Uint32	-	NO		-
	2_h	Velocity Factor 1's Divisor	RW	Uint32	-	NO		-
6096_h	-	Velocity Factor	RO	Uint8	-	NO	ARR	2
	1_h	Numerator	RW	Uint32	-	NO		-
	2_h	Divisor	RW	Uint32	-	NO		-
6097_h	-	Acceleration Factor	RO	Uint8	-	NO	ARR	2
	1_h	Acceleration Factor's Numerator	RW	Uint32	-	NO		-
	2_h	Acceleration Factor's Divisor	RW	Uint32	-	NO		-
6098_h	-	Homing Method	RO	int8	-	NO	VAR	
6099_h	-	Homing Speed	RO	Uint8	-	NO	ARR	2
	1_h	Speed During Search for Switch	RW	Uint32	puu/s	NO		-
	2_h	Speed During Search for Zero	RW	Uint32	puu/s	NO		-
$609A_h$	-	Homing Acceleration/Deceleration	RW	Uint32	puu/s^2	NO	VAR	-
$60B0_h$	-	Position Offset	RW	int32	puu	NO	VAR	-
$60B1_h$	-	Velocity Offset	RW	int32		RPDO	VAR	-
$60B2_h$	-	Torque Offset	RW	int16		RPDO	VAR	-
$60C0_h$	-	Interpolation Sub-mode Selection	RW	int16	-	NO	VAR	-
$60C1_h$	-	Interpolation Data Record	RO	Uint8	-	NO	ARR	2
	1_h	Interpolation Displacement	RW	int32	-	RPDO		-
	2_h	Interpolation Cycle	RW	int32		RPDO		-
$60C2_h$	-	Interpolation Cycle Period	RO	Uint8		NO	REC	2

	1_h	Interpolation Cycle Time Constant	RW	Uint8		RPDO		-
	2_h	Ordinal Number for Power of Ten	RW	int8	-	RPDO		-
$60C3_h$	-	Definition of Synchronous Interpolation	RO	Uint8	-	NO	ARR	2
	1_h	Synchronous Grouping	RW	Uint8	-	NO		-
	2_h	Times of Interpolation's Synchronous Triggering	RW	Uint8	-	NO		-
$60C4_h$	-	Interpolation Data Configuration	RO	Uint8	-	NO	REC	6
	1_h	Array's Max Value	RO	Uint32	-	NO		-
	2_h	Array's Actual Value	RW	Uint32	-	NO		-
	3_h	Array Buffer	RW	Uint8	-	NO		-
	4_h	Array Position	RW	Uint16	-	NO		-
	5_h	Size of Recorded Data	WO	Uint8	-	NO		-
	6_h	Clear Array	WO	Uint8	-	NO		-
$60C5_h$	-	Max Acceleration	RW	Uint32	puu/s^2	NO	VAR	-
$60C6_h$	-	Max Deceleration	RW	Uint32	puu/s^2	NO	VAR	-
$60E3_h$	-	Homing Method Supported	RO	Uint8	-	NO	ARR	-
$60F2_h$	-	Position Selection	RW	Uint16	-	NO		-
$60F4_h$	-	Position Actual Value's Deviation	RO	int32		TPDO	VAR	-
$60FA_h$	-	Position Loop Control	RO	int32	-	TPDO	VAR	-
$60FC_h$	-	Position Demand Value	RO	int32	puu	TPDO	VAR	-
$60FD_h$	-	Digital Input	RO	Uint32	-	TPDO	VAR	-
$60FE_h$	-	Digital Output	RO	Uint8	-	NO	VAR	2
	1_h	Physical Outputs	RW	Uint32	-	RPDO		-
	2_h	Logical Outputs	RW	Uint32	-	YES		-
$60FF_h$	-	Target Velocity	RW	int32	puu/s	RPDO	VAR	-
6502_h	-	Servo-supported Operation Mode	RO	Uint32	-	NO	VAR	CF_h

4.5 Detailed Description of Object Dictionary

4.5.1 Detailed Description of Communication Objects 1XXX$_h$ in Object Dictionary

Index 1000$_h$	Name	Device Type					Data Structure	VAR	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	20192$_h$
Device type parameter is used to describe the sub-protocol or application specification of the device used										

Index 1001$_h$	Name	Error Register					Data Structure	VAR	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint8	Ex-works Setting	---

Error information is described bit by bit as in the table below:

Bit	Meaning	bit	Meaning
0	Conventional	4	Communication
1	Current	5	Reserved
2	Voltage	6	Reserved
3	Temperature	7	Reserved

Occurrence of an error will set the corresponding bit at 1

Index 1003$_h$	Name	Pro-defined Error Field					Data Structure	ARR	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	---
Subindex 00$_h$	Name	Number of Errors					Data Structure	-	Data Type	Uint8
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	0
Only 0 can be written in, and all error records are cleared in this case.										
Subindex 01$_h$	Name	Standard Error Field					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	0

In case of driver warning, the error will be stored in the format shown below:

31 16	15 0
Manufacturer's error code	Standard error code
MSB	LSB

Index	Name	Auto Operation	Data	VAR	Data Type	Uint16

1004h						Structure			
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint16	Ex-works Setting	55AAh
Write in 55AAh and turn on the driver, it will automatically enter operational state.									

Index 1005h	Name	COB-ID SYNC Message				Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	80h
Default value is 80h, currently change to it is not supported.									

Index 1006h	Name	Communication Cycle Period				Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	0
It is only used for synchronous generators, and the unit is μs.									

Index 1008h	Name	Manufacturer Device Name				Data Structure	VAR	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	4C532020h

Index 1009h	Name	Manufacturer Hardware Version				Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	20081201h

Index 100Ah	Name	Manufacturer Software Version				Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	20081201h

Index 1010h	Name	Store Parameters				Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

"Store parameters" means storing the current parameter value into EEPROM, and the stored value will be loaded when the driver is reswitched on, the node is reset or communication is reset.

When parameters need be stored, the user shall specify the subindex corresponding to the storing area and write the ASCII characters of "save", otherwise the saving will not succeed.

The relationship between ASCII characters and hexadecimal values:

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

Type of the parameter to be stored is determined by the subindex.

Subindex 00h	Name	Largest Subindex Supported	Data Structure	-	Data Type	Uint8

	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	4
Subindex 01_h	Name	Save All Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to save all parameters in the list of object dictionary										
Subindex 02_h	Name	Save Communication Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to save parameters in group 1000_h in the list of object dictionary										
Subindex 03_h	Name	Save Application Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to save parameters in group 6000_h in the list of object dictionary.										
Subindex 04_h	Name	Save Manufacturer Defined Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to save parameters in group 2000_h in the list of object dictionary.										

Index 1011_h	Name	Restore Default parameters					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

"Restore Default parameters" means restoring the default parameters to EEPROM, but the restored default value (i.e. ex-works setting) won't be loaded immediately, but will be loaded when the driver is reswitched on, node is reset or communication is reset.

When restoring default parameters is needed, the user shall specify the subindex corresponding to the restoring area, and write in the ASCII characters of "load", otherwise restoring of the default value will not succeed.

The relationship between ASCII characters and hexadecimal values:

	MSB			LSB
ASCII	d	a	o	l
Hexadecimal	64_h	61_h	$6F_h$	$6C_h$

Type of the parameter to be stored is determined by the subindex

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	4
Subindex 01_h	Name	Restore All Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to restore all the parameters in the list of object dictionary.										

Subindex 02h	Name	Restore Communication Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to restore parameters in group 1000h in the list of object dictionary										

Subindex 03h	Name	Restore Application Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to restore parameters in group 6000h in the list of object dictionary										

Subindex 04h	Name	Restore Manufacturer Defined Parameters					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0
It is used to restore parameters in group 2000h in the list of object dictionary										

Index 1014h	Name	COB-ID Emergency Message					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	80h+Node_ID

各位(bit)的含义如下表：Meanings of all bits are shown in the table below:

Bit	Function	Description
31	Emergency(EMCY) function	0: Emergency(EMCY) function is activated (the servo driver will send EMCY commands) 1: Emergency(EMCY) function is suspended (the servo driver will not send EMCY commands)
30~11	Reserved	-----
10~0	Define 11-bit COD-ID	80h+ Node-ID

When an emergency protocol packet takes effect, its COB-ID must remain consistent with this object.

Index 1015h	Name	Inhibit Time EMCY					Data Structure	VAR	Data Type	Uint16
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint16	Ex-works Setting	---
It is used to set EMCY message's inhibiting time (unit: 100μs).										

Index 1016h	Name	Consumer Heartbeat Time					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

The data include the node address monitored and actual consumer time (unit: ms).
Data composition is as shown below:

31 24	23 16	15 0
Reserved	Node-ID	Fixed heartbeat time
MSB		LSB

In the setting, the upper device's heartbeat time must be longer than servo driver's heartbeat time.										
Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-1	Ex-works Setting	1
Subindex 01_h	Name	Consumer Heartbeat Time					Data Structure	Uint32	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	0

Index 1017_h	Name	Producer Heartbeat Time					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
It's used to set the time for the servo driver to report heartbeat; when its value is set as 0, this function is suspended. Unit: ms										

Index 1018_h	Name	Identity Object					Data Structure	REC	Data Type	-
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
This object contains information related to the driver.										
Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	4
Subindex 01_h	Name	Vendor-ID					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	$0AB6_h$
Subindex 02_h	Name	Product Code					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	$4C532020_h$
Subindex 03_h	Name	Revision Number					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0004_h
Subindex 04_h	Name	(Error Behavior)					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	0001_h

Index 1200_h	Name	SDO Server Parameter					Data Structure	REC	Data Type	SDO Parameter
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
This object is read-only (i.e. unsettable). By using this object, the user can read, send and receive										

SDO's COB-ID.

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-2	Ex-works Setting	2
Subindex 01_h	Name	COB-ID Client -> Server					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	600_h+ Node-ID
Subindex 02_h	Name	COB-ID Server -> Client					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	580_h+ Node-ID

Index 1400_h ~ 1403_h	Name	RPDO Communication Parameter					Data Structure	REC	Data Type	RPDO Parameter
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

It's used to receive PDO's communication setting.

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-2	Ex-works Setting	2
Subindex 01_h	Name	COB-ID Used by RPDO					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	See below

Corresponding indexes' setting is shown below:

1400_h: 00000200_h + Node-ID
1401_h: 00000300_h + Node-ID
1402_h: 00000400_h + Node-ID
1403_h: 00000500_h + Node-ID

Subindex 02_h	Name	RPDO Transmission Type					Data Structure	-	Data Type	Uint8
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint8	Ex-works Setting	FF_h

Different values stand for different PDO transmission types, see the table below for details:

Value	Meaning
0_h	Synchronous non-cycling
0_h ~$F0_h$	Synchronous cycling
FE_h、FF_h	Asynchronous non-cycling

The value may be changed only when PDO is ineffective.

Index 1600_h ~ 1603_h	Name	RPDO Mapping Parameter					Data Structure	REC	Data Type	RPDO Parameter
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

It is the mapping setting for receiving PDOs. The total length of mapped objects in one group of

PDOs may not exceed 64bits.

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	0

0: suspend the function and clear other indexes' mapped objects; 1-8: PDO mapping function is turned on and mapping number is set.

Subindex 01_h ~ 04_h	Name	Application Object					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	---

Indexes and subindexes of the mapped objects must be stored in the list of object dictionary; they must be writable and mappable.
Corresponding subindexes must be written in in the format shown below:

31 16	15 8	7 0
Index	Subindex	Object length

MSB LSB

Index 1800_h ~ 1803_h	Name	TPDO Communication Parameter					Data Structure	REC	Data Type	TPDO Parameter
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

It's the communication setting for sending PDOs.

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	0-2	Ex-works Setting	5
Subindex 01_h	Name	TPDO的COB-ID(COB-ID Used by TPDO)					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	See below

Setting of corresponding indexes is as shown below:

1800_h: 00000180_h + Node-ID
1801_h: 00000280_h + Node-ID
1802_h: 00000380_h + Node-ID
1803_h: 00000480_h + Node-ID

Subindex 02_h	Name	RPDO Transmission Type					Data Structure	-	Data Type	Uint8
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint8	Ex-works Setting	FF_h

Different values stand for different PDO transmission types, see the table below for details:

Value	Meaning
0_h	Synchronous non-cycling
0_h ~$F0_h$	Synchronous frame triggering
FE_h	Value change triggering
FF_h	Synchronous cycling, time interval to be set by subindex 05_h (event timer).

It may be set as per actual needs.

Subind	Name	Inhibit Time	Data	-	Data Type	Uint16

ex 03_h							Structure			
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint16	Ex-works Setting	32_h
Its unit is 100μs. If it's 0, it means the inhibit time is ineffective.										
Subindex 05_h	Name	Event Timer					Data Structure	-	Data Type	Uint16
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint16	Ex-works Setting	32_h
Its unit is 1ms. If it's 0, it means the event timer is ineffective.										

Index $1A00_h$ ~ $1A03_h$	Name	TPDO Mapping Parameter					Data Structure	REC	Data Type	TPDO Parameter
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
It's the mapping parameter setting for sending PDOs. The total length of mapped objects in one group of PDOs must not exceed 64bits.										
Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	0-4	Ex-works Setting	0
0: suspend the function and clear other indexes' mapped objects; 1-8: PDO mapping function is turned on and mapping number is set.										
Subindex 01_h ~ 04_h	Name	Application Object)					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	Uint32	Ex-works Setting	---

Content indexes and subindexes of the mapped objects must be stored in the list of object dictionary; they must be readable, writable and mappable.

Corresponding subindexes must be written in in the format shown below:

31 16	15 8	7 0
Index	Subindex	Object length

MSB LSB

4.5.2 Detailed Description of User-defined Parameters 2XXXh in Object Dictionary

Object groups 2000_h ~ 4000_h are object parameters defined by Beijing HollySys Electric Technology Co., Ltd., including Fn function parameters and Dn status parameters.

Fn function parameters' offset addresses are 2000_h and 3000_h, the former is for writing to be stored, the latter for writing not to be stored. Dn status parameters' offset address is 4000_h.

Matching Relationship		Remarks
Object Dictionary	User-defined Parameter	
2000_h---$21FF_h$	Fn 000 ---Fn 1FF	It's for servo operation parameter register's Fn parameters. Written values are saved after reenergization.
3000_h---$31FF_h$	Fn 000 ---Fn 1FF	It's for servo operation parameter register's Fn parameters. Written values are not saved after reenergization.
4000_h---$40FF_h$	Dn 00 ---Dn FF	It's for servo operation status register's Dn parameters.

For detailed description of the parameters, please refer to *Chapter IV Setting* of *LS- and DM-Series Low-voltage Servo Driver Manual*.

4.5.3 Detailed Description of Sub-protocol Parameters 6XXXh in Object Dictionary

Index 6007_h	NameName	Abort Connection Option Code					Data Structure	VAR	Data Type	int16
	Accessibility	RW	Mappability	NO	Related Modes	ALL	Data Range	int16	Ex-works Setting	---

This object is used to set what action is to be performed when connection with the network is cut off.
Meanings of different values are in the table below:

Value	Meaning
0_h	No action
1_h	Fault
2_h	Device control command "Prohibit voltage"
3_h	Device control command "Quick stop"

Parameter values can be set as per actual needs.

Index $603F_h$	NameName	Error Code					Data Structure	VAR	Data Type	Uint16
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	Uint16	Ex-works Setting	0

When the driver has warning or a fault, the corresponding warning or fault information code can be got through index $603F_h$. Detailed fault information is in *LS- and DM-Series Low-voltage Servo Driver Manual*.

Index 6040_h	NameName	Control Word					Data Structure	VAR	Data Type	Uint16
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint16	Ex-works Setting	0

Control command has many functions, such as servo on, command triggering, error reset, and emergency stop.

位(bit)	Name	Description

0	Switch on	0- ineffective 1- effective
1	Enable voltage	0- ineffective 1- effective
2	Quick stop	1- ineffective 0- effective
3	Enable operation	0- ineffective 1- effective
4~6	Operation mode specific	It's related to each servo driver's operation modes, details of which is in the table below.
7	Fault reset	It performs fault reset for resettable faults and warning. It becomes effective on the rising edge of bit7;
8	Halt	0- ineffective 1- effective
9~10	Reserved	Preserved
11~15	Manufacturer specific	Preserved without definition

Note: Value assignment to a single bit of the control word is meaningless, a control command can be made only by combination of multiple bits.

Meanings of bit0~bit3 and bit7 remain unchanged for different modes of the servo driver.

Relationship between bit4~bit6 and the servo driver's modes is as shown in the table below:

Bit	Control Mode		
	Speed Mode Torque Mode	Position Mode	Homing Mode
4	Reserved	New set-point (positive edge triggering)	Home operation start (positive edge triggering)
5	Reserved	Change set immediately	Reserved
6	Reserved	1 - Relative position command 0 - Absolute position command	Reserved

These commands must be sent in order, otherwise the servo driver can't enter the desired state.

Index 6041_h	NameName	(Status Code)					Data Structure	VAR	Data Type	Uint16
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	Uint16	Ex-works Setting	0

It reflects status of the servo driver:

位bit	NameName	描述
0	Ready to switch on	0 - Not ready 1 –Ready
1	Switched on	0 - Not waiting to be switched on 1 –Waiting to be switched on
2	Operation enabled	0 - Not enabled 1 - Enabled
3	Fault	0 - No fault 1 - In fault
4	Voltage enabled	0 - Main circuit power not on 1 - Main circuit power on
5	Quick stop	0 - Stopped

		1 - Not stopped
6	Switch on disabled	0 - Switch on not disabled 1 - Switch on disabled
7	Warning	0 - No warning 1 - Warning given
8	Manufacturer	Reserved, always at 0
9	Remote	Always at 0
10	Target reached	0 - Target not reached 1 - Target reached
11	internal limit active	0 - No position limit 1 - Position limit functioning
12~13	Operation mode specific	It's related to servo drivers' modes Details are in the table below
14~15	Manufacturer specific	Reserved

Note: Value assignment to a single bit of the control word is meaningless, a status word to describe the servo driver's current state can be made only by combination of multiple bits.

Meanings of bit0~bit9 remain unchanged for different modes of the servo driver.

Relationship between bit12~bit13 and the servo driver's modes is as shown in the table below:

Bit	Control Mode			
	Speed Mode	Torque Mode	Position Mode	Homing Mode
12	Zero speed state	Reserved	Set-point Acknowledge	Homing attained
13	Reserved	Reserved	Following error	Homing error

The states in the table are to be fed back by the servo driver.

Index 6060_h	Name	Mode of Operation					Data Structure	VAR	Data Type	int8
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	0-6	Ex-works Setting	0

It is used to select the servo driver's mode of operation:

Value	Description	Note
0	---	Reserved
1	Position mode	Refer to 3.4
2	---	Reserved
3	Speed mode	Refer to 3.5
4	Torque mode	Refer to 3.6
5	---	Reserved
6	Homing mode	Refer to 3.7

If an unsupported servo mode is selected through SDO or PDO, the driver will be unable to work normally.

Index 6061_h	Name	Mode of Operation Display					Data Structure	VAR	Data Type	int8
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	0-6	Ex-works Setting	0

It is used to display the servo driver's current control mode:

Value	Description	Note
0	---	Reserved
1	Position mode	Refer to 3.4
3	Speed mode	Refer to 3.5
4	Torque mode	Refer to 3.6
6	Homing mode	Refer to 3.7

Only after the servo driver enters enabled state can 6061_h display the servo driver's current control mode.

Index 6063_h	Name	Position Actual Internal Value					Data Structure	VAR	Data Type	int32
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int32	Ex-works Setting	-
It is used to display feedback of the motor's real-time absolute position, the unit is number of pulses.										

Index 6064_h	Name	Position Actual Value					Data Structure	VAR	Data Type	int32
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int32	Ex-works Setting	-
It is used to display feedback of the motor's real-time absolute position, the unit is number of pulses.										

Index 6069_h	Name	Velocity Sensor Actual Value					Data Structure	VAR	Data Type	int32
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int32	Ex-works Setting	-
It is used to display the actual speed calculated by the motor's internal sensor, the unit is number of pulses per second.										

Index $606A_h$	Name	Sensor Selection Code					Data Structure	VAR	Data Type	int16
	Accessibility	RW	Mappability	TPDO	Related Modes	ALL	Data Range	int16	Ex-works Setting	1

It is used to select the velocity sensor.

Value	Description
0	Actual speed is obtained through position encoder
1	Actual speed is obtained through speed encoder

Default value is 1.

Index $606B_h$	Name	Velocity Demand Value					Data Structure	VAR	Data Type	int32
	Accessibility	RO	Mappability	TPDO	Related Modes	PV	Data Range	int32	Ex-works Setting	--
It is speed command to be given under speed mode; the unit is puu/s.										

Index $606C_h$	Name	Velocity actual value	Data Structure	VAR	Data Type	int32

	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int32	Ex-works Setting	--
It displays the motor's actual speed to facilitate the user's monitoring; the unit is number of pulses per second.										

Index 6071_h	Name	Target Torque					Data Structure	VAR	Data Type	int16
	Accessibility	RW	Mappability	YES	Related Modes	PT	Data Range	int16	Ex-works Setting	--
It is the motor target output torque under torque mode; the unit is thousandth of rated torque. If the object is set at 1000, the target torque is the same as the motor's rated torque.										

Index 6072_h	Name	Max Torque					Data Structure	VAR	Data Type	Uint16
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint16	Ex-works Setting	--
It is the maximum torque output by the driver; the unit is thousandth of rated torque.										

Index 6073_h	Name	Max Current					Data Structure	VAR	Data Type	Uint16
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint16	Ex-works Setting	--
It is the driver's current when it's outputting its maximum torque; the unit is thousandth of rated current.										

Index 6074_h	Name	Torque Demand Value					Data Structure	VAR	Data Type	int16
	Accessibility	RO	Mappability	TPDO	Related Modes	PT	Data Range	int16	Ex-works Setting	--
It is used to describe the torque command under torque mode; the unit is thousandth of rated torque.										

Index 6075_h	Name	Motor Rated Current					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	--
It is used to describe the applicable motor's rated current; the unit is mA.										

Index 6076_h	Name	Motor Rated Torque					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	--
It is used to describe the applicable motor's rated torque; the unit is 0.001N.m.										

Index 6077_h	Name	Torque Actual Value					Data Structure	VAR	Data Type	int16
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int16	Ex-works Setting	--
It is used to describe the motor's actual output torque; the unit is thousandth of rated torque.										

Index 6078_h	Name	Current Actual Value					Data Structure	VAR	Data Type	int16
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	int16	Ex-works Setting	--
It is used to describe the motor's actual output current; the unit is mA.										

Index 6079_h	Name	DC Link Circuit Voltage					Data Structure	VAR	Data Type	Uint32
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	--
It is used to describe the driver's DC link voltage; the unit is mV.										

Index $607A_h$	Name	Target Position					Data Structure	VAR	Data Type	int32
	Accessibility	RW	Mappability	YES	Related Modes	PP	Data Range	int32	Ex-works Setting	--

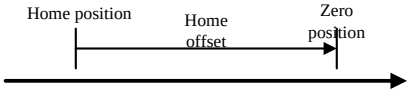
It is used to set the servo driver's target position under position mode:

6040_h的 bit6 6040_h's bit6	Description
0	Under absolute position mode, $607A_h$ is current stage's target absolute position; when the command's execution is completed, the (absolute) Position Actual Value 6064_h = $607A_h$.
1	Under relative position mode, $607A_h$ is current stage's target incremental position; when the command's execution is completed, the user's displacement increment = $607A_h$.

Unit: number of pulses

Index $607C_h$	Name	Home Offset					Data Structure	VAR	Data Type	int32
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	int32	Ex-works Setting	--

It is the home reference point to be located during homing; when the home reference point is located, the position whose coordinates are determined by the offset from that point is the home defined by the user.



Unit: number of pulses.

Index $607E_h$	Name	Polarity					Data Structure	VAR	Data Type	Uint8
	Accessibility	RW	Mappability	YES	Related Modes	-	Data Range	Uint8	Ex-works Setting	0

It is used to set the position command's or speed command's polarity.

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

Position polarity	Speed polarity	reserved

Bit7=1 means by multiplying position command ($607A_h$) by-1 under position mode, the user can reverse rotation of the motor.
Bit6=1 means by multiplying speed command ($60FF_h$) by -1 under speed mode, the user can reverse rotation of the motor.

Index $607F_h$	Name	Max Profile Velocity					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	--
It is used to set the user's max profile velocity; the unit is number of pulses per second.										

Index 6080_h	Name	Max Motor Velocity					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	3600
It is used to set the max motor velocity during its operation; the unit is RPM.										

Index 6081_h	Name	Profile Velocity					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	PP	Data Range	Uint32	Ex-works Setting	--
It is the motor's target speed under position mode; the unit is number of pulses per second.										

Index 6083_h	Name	Profile Acceleration					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	$7FFFFFFF_h$
It is the acceleration during the acceleration stage of the command's execution; the unit is number of pulses per square second.										

Index 6084_h	Name	Profile Deceleration					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	$7FFFFFFF_h$
It is the deceleration during the deceleration stage of the command's execution; the unit is number of pulses per square second.										

Index 6086_h	Name	Motion Profile Type					Data Structure	VAR	Data Type	int16
	Accessibility	RW	Mappability	YES	Related Modes	ALL	Data Range	int16	Ex-works Setting	0
It is used to set motion profile type of the motor's position command. -1: user-defined S profile; 0: T-type acceleration and deceleration profile.										

Index 6087_h	Name	Torque Slope					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	YES	Related Modes	PT	Data Range	Uint32	Ex-works Setting	0

It is used to set the time slope as the time needed for transition from 0 to 100% rated torque. The unit is thousandth of rated torque per second.

Index 608F$_h$	Name	Position Encoder Resolution					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
It is used to set the position encoder's counting increment for one turn of the motor's rotation. Position encoder resolution = position encoder counting increment / number of turns of motor rotation										
Subindex 00$_h$	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01$_h$	Name	Encoder Increments					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	2710$_h$
It is used to display the increment of position encoder's counting after the motor has rotated a number of turns.										
Subindex 02$_h$	Name	Motor Revolutions					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to display the number of turns the motor has rotated.										

Index 6090$_h$	Name	Velocity Encoder Resolution					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
It is used to display the per second increment of encoder for each second's number of turns of the motor's rotation. Speed encoder resolution = encoder's per second increment / motor's per second rotation turns										
Subindex 00$_h$	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01$_h$	Name	Encoder Increments Per Second					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	2710$_h$
It is used to display the encoder's per second increment.										
Subindex 02$_h$	Name	Motor Revolutions Per Second					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to display the motor's number of turns per second.										

Index 6094$_h$	Name	Velocity Encoder Factor					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-

It is the velocity encoder factor, used to establish the proportion between the user-specified load sped and motor position increment. 606C

Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01_h	Name	Numerator					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the motor's rotation speed.										
Subindex 02_h	Name	Divisor					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the load's speed.										

Index 6095_h	Name	Velocity Factor 1					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	-
Velocity factor 1 is used to establish the proportional relationship between the load's speed and motor speed										
Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01_h	Name	Numerator					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the load's speed.										
Subindex 02_h	Name	Divisor					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the motor's speed.										

Index 6098_h	Name	Homing Method					Data Structure	VAR	Data Type	int8
	Accessibility	RW	Mappability	YES	Related Modes	HM	Data Range	0-35	Ex-works Setting	0

Select the method of homing:

Value (decimal)	Description
1	Homing upon touching backward-rotation prohibiting switch and Z pulse
2	Homing upon touching forward-rotation prohibiting switch and Z pulse
3	Backward-rotating homing upon touching home switch's upper edge and Z pulse
4	Forward-rotating homing upon touching home switch's upper edge and Z

	pulse
5	Forward-rotating homing upon touching home switch's lower edge and Z pulse
6	Backward-rotating homing upon touching home switch's lower edge and Z pulse
7	Backward-rotating homing upon touching home switch's upper edge, forward-rotation prohibiting switch and Z pulse
8	Forward-rotating homing upon touching home switch's upper edge, forward-rotation prohibiting switch and Z pulse
9	Backward-rotating homing upon touching home switch's lower edge, forward-rotation prohibiting switch and Z pulse
10	Forward-rotating homing upon touching home switch's lower edge, forward-rotation prohibiting switch and Z pulse
11	Forward-rotating homing upon touching home switch's upper edge, backward-rotation prohibiting switch and Z pulse
12	Backward-rotating homing upon touching home switch's lower edge, backward-rotation prohibiting switch and Z pulse
13	Forward-rotating homing upon touching home switch's upper edge, backward-rotation prohibiting switch and Z pulse
14	Backward-rotating homing upon touching home switch's upper edge, backward-rotation prohibiting switch and Z pulse
17	Homing upon touching backward-rotation prohibiting switch
18	Homing upon touching forward-rotation prohibiting switch
19	Backward-rotating homing upon touching home switch's upper edge
20	Forward-rotating homing upon touching home switch's upper edge
21	Forward-rotating homing upon touching home switch's lower edge
22	Backward-rotating homing upon touching home switch's lower edge
23	Backward-rotating homing upon touching home switch's upper edge and forward-rotation prohibiting switch
24	Forward-rotating homing upon touching home switch's upper edge and forward-rotation prohibiting switch
25	Backward-rotating homing upon touching home switch's lower edge and forward-rotation prohibiting switch
26	Forward-rotating homing upon touching home switch's lower edge and forward-rotation prohibiting switch
27	Forward-rotating homing upon touching home switch's lower edge and backward-rotation prohibiting switch
28	Backward-rotating homing upon touching home switch's lower edge and backward-rotation prohibiting switch
29	Forward-rotating homing upon touching home switch's upper edge and backward-rotation prohibiting switch
30	Backward-rotating homing upon touching home switch's upper edge and backward-rotation prohibiting switch
33	Backward-rotating homing upon touching Z pulse
34	Forward-rotating homing upon touching Z pulse
35	Current position is home

When 6098h = 15, 16, 31, 32, the value assignment is meaningless, and the servo driver will not perform homing action.

Index 6099_h	Name	Homing Speed					Data Structure	ARR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	HM	Data Range	-	Ex-works Setting	-
It is used to set 2 speed values under homing mode: 1. speed during search for deceleration point signal; 2. speed during search for home signal;										
Subindex 00_h	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01_h	Name	Speed During Search For Switch					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the speed during search for deceleration point signal; it could be set as a big value to prevent the homing time from being too long and to prevent the consequent homing time-out fault. The unit is pulses per second										
Subindex 02_h	Name	Speed During Search For Zero					Data Structure	-	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	-	Data Range	-	Ex-works Setting	1
It is used to set the speed during search for home signal; it could be set as a small value to prevent the servo driver's overshot when it's stopping from a high speed and to prevent the consequent big deviation between the stopping position and the set machine home. The unit is pulses per second.										

Index $609A_h$	Name	Homing Acceleration)					Data Structure	VAR	Data Type	Uint32
	Accessibility	RW	Mappability	NO	Related Modes	HM	Data Range	Uint32	Ex-works Setting	0
It is used to set the acceleration under homing mode. The unit is pulses per square second.										

Index $60FC_h$	Name	Position Demand Value					Data Structure	VAR	Data Type	int32
	Accessibility	RO	Mappability	YES	Related Modes	ALL	Data Range	int32	Ex-works Setting	0
It is used to describe the motor's real-time position command. User Position Command (6062_h) ×Position Factor (6093_h) = Position Demand Value $60FC_h$										

Index $60FD_h$	Name	Digital Input					Data Structure	VAR	Data Type	Uint32
	Accessibility	RO	Mappability	TPDO	Related Modes	ALL	Data Range	Uint32	Ex-works Setting	0

The driver's current digital input logic is: 0: - the logic is ineffective; 1- the logic is effective.
The digital input signals stood for by all the bits are shown below:

31~16	15~4	3	2	1	0
Manufacturer-defined	Reserved	Undefined	Home	Forward-rotation	Backward-rotation

				switch	prohibiting switch	prohibiting switch
MSB						LSB

Index 60FE$_h$	Name	Digital Output					Data Structure	ARR	Data Type	Uint32
	Accessibility	RO	Mappability	RPDO	Related Modes	ALL	Data Range	-	Ex-works Setting	-
It is used to describe the driver's current digital output.										
Subindex 00$_h$	Name	Largest Subindex Supported					Data Structure	-	Data Type	Uint8
	Accessibility	RO	Mappability	NO	Related Modes	-	Data Range	2	Ex-works Setting	2
Subindex 01$_h$	Name	Physical Outputs					Data Structure	-	Data Type	Uint32
	Accessibility	RO	Mappability	RPDO	Related Modes	-	Data Range	Uint32	Ex-works Setting	1

It is used to describe the logic value of the driver's current digital output: 0- the logic is ineffective; 1- the logic is effective.
The digital output signals stood for by all the bits are shown below:

31~16	15~1	0
Manufacturer-defined	Reserved	Brake output
MSB		LSB

Index 60FF$_h$	Name	Target Velocity					Data Structure	VAR	Data Type	int32
	Accessibility	RW	Mappability	YES	Related Modes	PV	Data Range	int32	Ex-works Setting	0
It is used to set the user's speed command under speed mode; the unit is number of pulses per second.										

Index 6502$_h$	Name	Status Word					Data Structure	VAR	Data Type	Uint32
	Accessibility	RO	Mappability	TPDO	Related Modes	-	Data Range	0~2^{32}-1	Ex-works Setting	CF$_h$

It is used to describe the servo operation modes supported by the driver:

Bit	Description	Support or not: 0- not supported; 1-supported
0	pp (position mode)	1
1	Reserved	1
2	pv (speed mode)	1
3	tq (torque mode)	1
4	Reserved	0
5	hm (homing mode)	1
6	ip (interpolation mode)	——
7	csp (cycling synchronous position mode)	0
8	csv (cycling synchronous velocity mode)	0
9	cst (cycling synchronous torque mode)	0
10~31	Reserved	0

If the CANopen device supports object dictionary 6502$_h$, the user can use the object dictionary 6502$_h$ to know what servo modes are supported by the driver.

Chapter V Annex

5.1 Examples of Commonly Used Commands

The examples below are based on assumption that CAN-ID is 1.

5.1.1 Commands for Saving Parameters

1010_h is used for saving relevant setting.

Note: As the time for saving parameters is long, saving will succeed only after reception of the command-saving SDO's response telling the writing is successful. Saving has not succeeded before that response from the SDO is received.

5.1.1.1 Save All Parameters

Frame ID	Data
601	23 10 10 01 73 61 76 65

5.1.1.2 Save Communication Parameters

Frame ID	Data
601	23 10 10 02 73 61 76 65

5.1.1.3 Save Application Parameters

Frame ID	Data
601	23 10 10 03 73 61 76 65

5.1.2 Command for Restoring Parameters

1011_h is used for restoring relevant setting.

5.1.2.1 Restore All Default Parameters

Frame ID	Data
601	23 11 10 01 6C 6F 61 64

5.1.2.2 Restore Default Communication Parameters

Frame ID	Data
601	23 11 10 02 6C 6F 61 64

5.1.2.3 Restore Default Application Parameters

Frame ID	Data
601	23 11 10 03 6C 6F 61 64

5.1.3 Command to Remove Faults

Bit7 of 6040_h - 00_h is for removing faults.

Fault reset: The user can use fault reset for resettable faults and warning. The resetting function is effective on bit7's rising edge, when bit7's value is maintained at 1, other control commands are ineffective.

After the fault is removed, the driver will remain in enabled state:

Frame ID	Data
601	2B 40 60 00 8F 00 00 00

5.2 Servo Fault Codes

Error Code	LED Display	Fault Description
1200	E brSL	Discharging resistor's resistance value is too small
1301	E coUt	Communication time-out
1320	E cotS	Stop caused by disconnection from network
1330	E cStP	Self-activated stop
1401	E dnRE	Error warning that the driver has no response
1500	E EnAb	Encoder AB signal alarm
1510	E EncU	Encoder UVW signal alarm
1511		

1600	E_FrAE	FRAM data writing operation check error
1700	E GEAr	Abnormal electronic gear parameters
2200	E LUdc	Undervoltage of power supply warning
2500	E oc-A	Phase-A overcurrent warning
2501	E oc-b	Phase-B overcurrent warning
2502	E oc-C	Phase-C overcurrent warning
2510	E oLod	Overload fault warning
2520	E oSPE	Overspeed warning
2530	E oUdc	Power supply overvoltage warning
2540	E oU-P	Abnormal phase voltage
2600	E PArA	FRAM's error of parameter overflow
2605	E orEr	Homing time-out
2610	E PEOU	Overflow of the position deviation counter
2630	E PHot	Power parts overheating
2645	E PosE	Over-travel warning
2660	E PS1E	1-phase current ADC null point exception warning
2661	E PS2E	2-phase current ADC null point exception warning
2900	E SPEE	Stall warning
3110	E USPn	Motor model code not supported
3600	E 2LoS	Warning of Z pulse loss of the encoder
3601	E 2EtE	Error alarm of too many Z pulses of the encoder

Manufacturer: Beijing HollySys Electric Technology Co., Ltd.
(formerly Stone Motor)
Add: 10th Floor, Block A, Huizhi Mansion, 9 Xueqing Road, Haidian District, Beijing
Postal Add: Postbox 2877, Beijing, 100085
Central exchange: (010) 62932100
Extensions: sales－100; technology support－810
Sales hotline: (010) 62927938
Fax: (010) 62927946
Website: www. syn-tron. com

Nanjing office: (025) 84293632/37/52/53
Shenzhen branch: (0755) 26581960/61/62