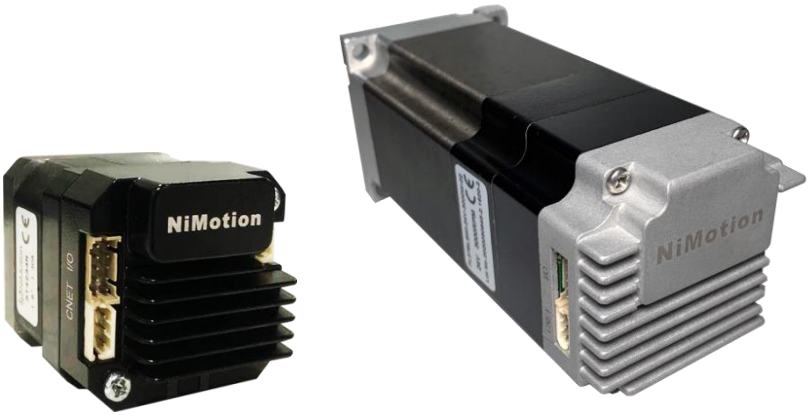
PMM60L、STM-M、BLM 、TSM Series Integrated Motor

CANopen Communication User Manual

Version No.A02



Beijing NiMotion Control Technology Co., Ltd.

Catalog

1 About this Manual....1
1.1 Brief Introduction....1
1.2 The implication of the numerical values....1
1.3 Bit....1
2 Terms and Abbreviations....2
3 Communication Network....3
3.1 Overview of CANopen Protocol....3
3.2 SDO Access....3
3.3 PDO Message....6
3.4 EMCY Message....11
3.5 Network Management (NMT service)....12
3.6 System Setting....14
4 Control of Equipment....16
4.1 The CiA402 State Machine....16
5 Unit Conversion....22
5.1 $608F_h$:Resolution of position encoder....22
5.2 6091_h:Transmission Ratio....23
5.3 607Eh: Polarity....24
5.4 Example....25
6 Operation Mode....26
6.1 Overview of Operation Mode....26
6.2 Position Control Function....27
6.3 Profile Position Mode（PP）....30
6.4 Velocity Mode....39
6.5 Profile Velocity Mode (PV)....43
6.6 Profile Torque Mode(PT)....46
6.7 Homing Mode....48
6.8 Interpolated Position Mode....60
6.9 Cyclic Sync Position Mode (CSP)....63
6.10 Cyclic Sync Velocity Mode(CSV)....66
6.11 Cyclic Sync Torque Mode (CST)....68
6.12 NiMotion Mode....70
6.13 NiMotion Position Mode....70
6.14 NiMotion Velocity Mode....73
6.15 NiMotion Torque Mode....76
6.16 Multi-Segment Position Mode....79
6.17 Multi-velocity mode....83
7 Special Functions....88
7.1 Digital Inputs and Output....88
7.2 Analog Input....92

7.3 Brake Setting 101
7.4 I^2t Overload Protection of Motor 103
7.5 Brake Configuration 103
7.6 Position Recovery 105
7.7 Parameter Saving and Restore 106
7.8 Power Limit Function 107
7.9 Modify CAN-ID Online 107
7.10 Absolute Value Encoder 108
7.11 Communication Watchdog Function 108
8 Adjust 110
8.1 Summary 110
8.2 Manual Gain Adjustment 110
8.3 Resonance Suppression 116
9 Error Management 119
9.1 Alarm Codes 119
9.2 Fault action settings 121
9.3 Fault Reset 123
9.4 Fault detection instructions 123
10 Object Dictionary 127
10.1 Communication Parameter Group 1000_h Description 129
10.2 Manufacturer Defined Parameter Group 2000_h Description 135
10.3 Sub-protocol Definition Parameter Group 6000_h Description 168
11 Version Information 176

1 About this Manual

1.1 Brief Introduction

This manual explains how to program and operate the integrated low-voltage Servo motor manufactured by Beijing NiMotion Control Technology Co., Ltd.

1.2 The implication of the numerical values

In this manual, numerical values are usually in decimal notation. If they have to be in hexadecimal notation, the subscript "h" at the end of the number is used, for example: 1003_h.

Directory index and sub-index of object are represented as follows:

<index>:<sub-index>

For example, sub-index 02 of object 1003_h is represented as **1003_h:02_h**.

1.3 Bit

Bits of an object always start from 0 at LSB.

	MSB							LSB
Bit Number	7	6	5	4	3	2	1	0
Bits	0	1	0	1	0	1	0	1

$\triangleq 55_{hex} \triangleq 85_{dec}$

2 Terms and Abbreviations

CAN	Controller area network
CAN-ID	Controller area network Identifier
COB	Communication object
COB-ID	Communication object Identifier
CIA	CAN in Automation
LSB	Least significant bit
MSB	Most significant bit
PDS	Power drive system
FSA	Finite state automaton
PDO	Process data object
SDO	Service data object
RPDO	Receive process data object
TPDO	Transmit process data object

3 Communication Network

3.1 Overview of CANopen Protocol

CANopen-bus integrated motor of Beijing NiMotion Control Technology Co., Ltd. adopts CANopen communication control,which is an application-layer protocol of network transmission system based on CAN serial bus,following ISO/OSI standard model. Different devices in the network exchange data by object dictionaries or objects. By process data objects (PDO) or service data objects (SDO),the master node can obtain or modify the data in object dictionaries of other node. The integrated low voltage servo motor driver is designed according to CiA402 protocol. The CANopen device model is shown in the figure.

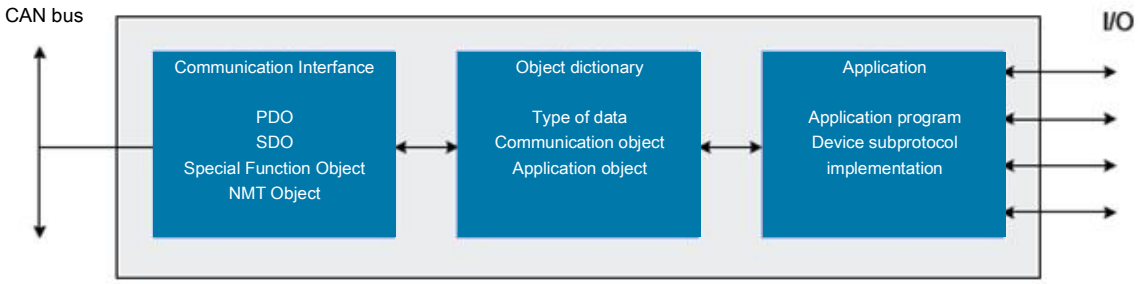


Figure 3-1

The communication objects are as follows:

Table 3-1

SDO	Service Data Object	Set the general parameters of the controller
PDO	Process Data Object	Rapid exchange process data(e.g. actual location)
EMCY	Emergency Message	Transfer drive fault information
NMT	Network Management	CANopen network manage
HEARTBEAT	Error Control	monitor the node's life status

3.2 SDO Access

The service data object (SDO) establishes contact with the object dictionary by the object index and sub-index. By SDO, the object contents in the object dictionary can be read, or be modified if allowed.

SDO Transport Framework

SDO transmission mode follows the client-server mode, that is, one response one answer. Initiated by SDO client in CAN-bus network, respond by SDO server. Therefore, the data exchange between SDO requires two CAN messages at least, and the identifiers of the two CAN messages are different. The SDO transmission model is shown in the figure below:

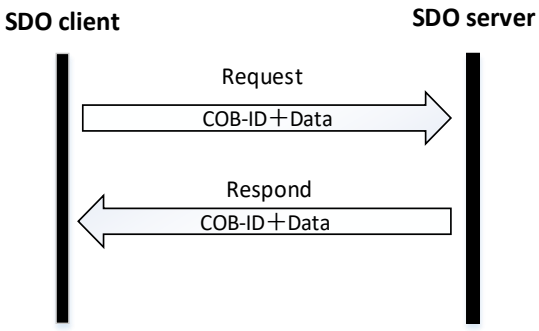


Figure 3-2 The SDO transmission model

3.2.1 SDO Transmission Message

SDO transmission message is composed of COB-ID and data segments. The data segment adopts little endian mode,that is, the lower bit in the front,and the high bit in the back. All SDO message data segments must be 8-bytes long.

The format of data message written by SDO is as follows:

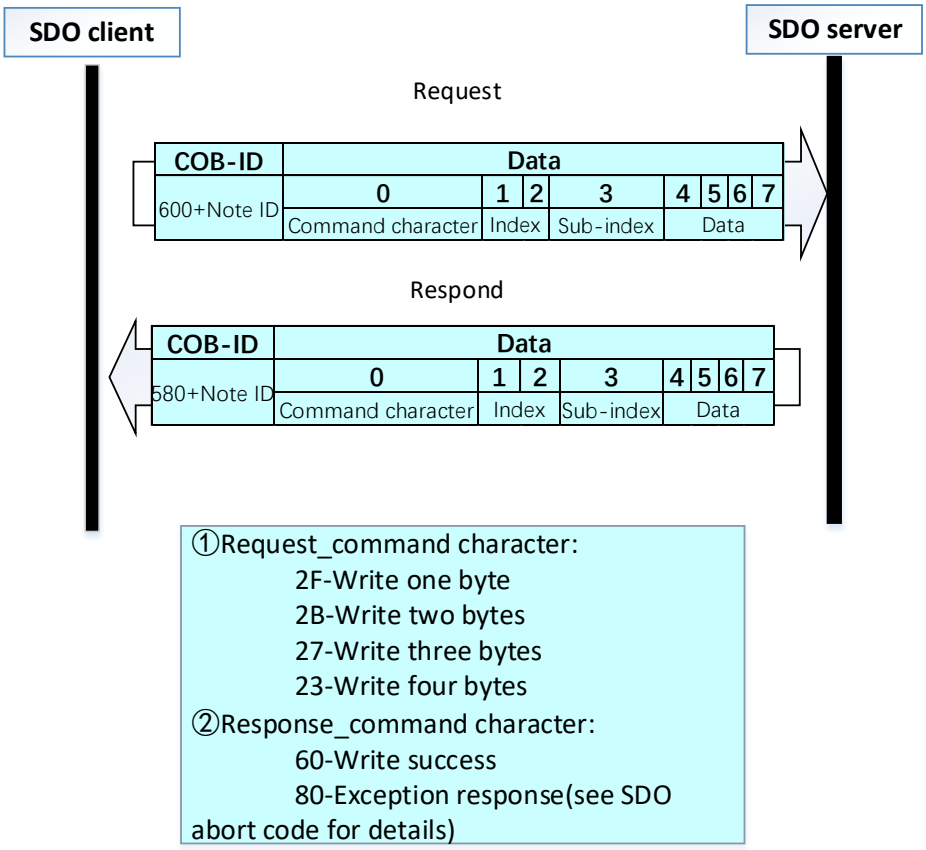


Figure 3-3

The format of data message read by SDO is as follows:

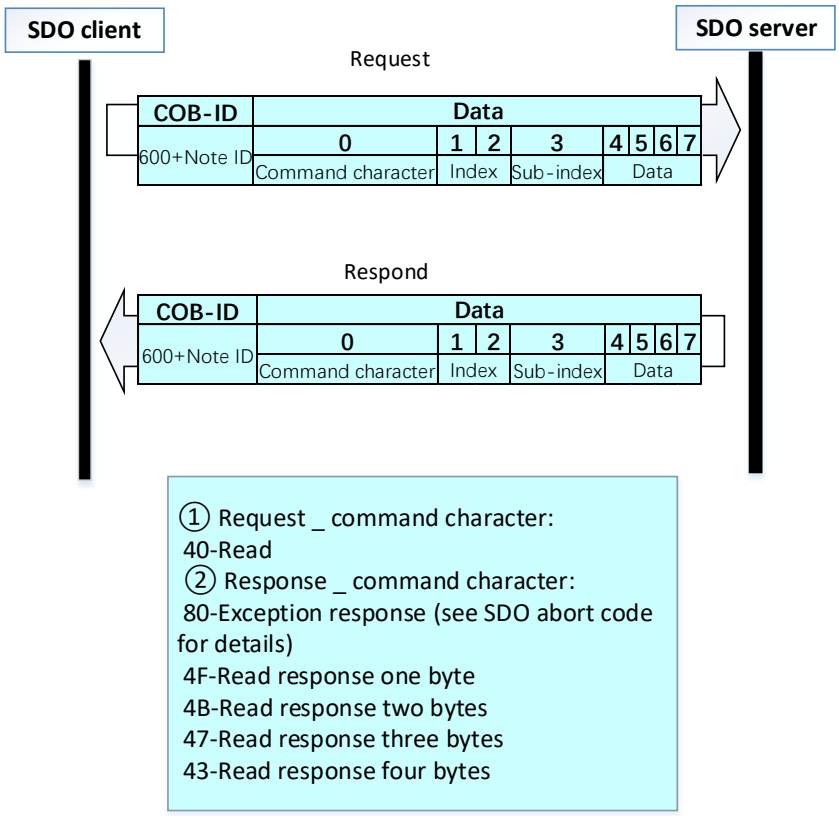


Figure 3-4

3.2.2 SDO Error Message

When an error occurs during reading or writing (for example,due to the object dictionary does not exist), the controller will not confirm it, but send a error report . As shown in the figure below, the object in 6041_h is read-only,when the object is write,it returns an error code of 0x06010002 (the lower bits first, the higher bits second).

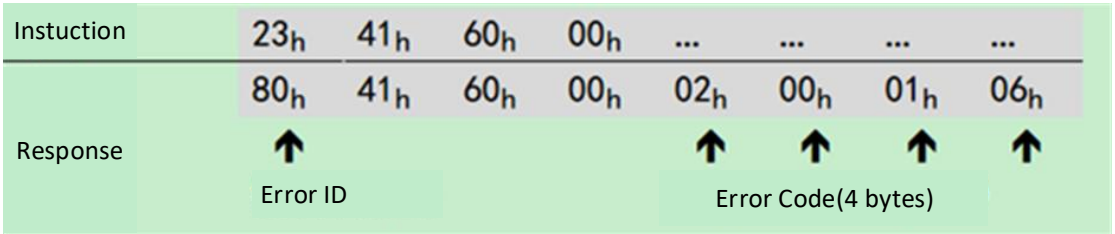


Figure 3-5 Examples of SDO error reports

SDO abnormal response aborted code (Hex):

Table 3-2

Interrupt Code	Description
05 03 00 00	Trigger bit does not change alternately
05 04 00 00	SDO protocol timeout
05 04 00 01	Illegal or unknown Client / Server, command word
05 04 00 02	Invalid block size (Block transfer mode only)
05 04 00 03	Invalid sequence number (Block transfer mode only)

Interrupt Code	Description
05 04 00 04	CRC error (Block transfer mode only)
05 04 00 05	Out of memory
06 01 00 00	Object does not addressable
06 01 00 01	Attempt to read write-only objects
06 01 00 02	Attempt to write read-only object
06 02 00 00	Object does not exist in object dictionary
06 04 00 41	Object cannot be mapped to PDO
06 04 00 42	Number or length of objects exceeds PDO length
06 04 00 43	General parameters incompatible
06 04 00 47	General equipment internal incompatibility
06 06 00 00	Object access failed due to hardware error
06 07 00 10	Data type mismatch, service parameter length mismatch
06 07 00 12	Data type mismatch, service parameter length is too large
06 07 00 13	Data type mismatch, service parameter length is too short
06 09 00 11	Sub-index does not exist
06 09 00 30	Out of parameter value range
06 09 00 31	Write parameter value is too large (during writing)
06 09 00 32	Write parameter value is too small (during writing)
06 09 00 36	Max value less than min value
08 00 00 00	General error (e.g. bytes number wrong)
08 00 00 20	Data cannot be transferred or saved to the application program
08 00 00 21	Data cannot be transferred or saved to the application due to local control
08 00 00 22	Data cannot be transferred or saved to the application due to the current status of the device
08 00 00 23	The object dictionary dynamically generates an error or does not exist

3.3 PDO Message

Process data object (PDO) is used to transmit real-time data, which is the most important data transmission mode in CANopen. Through PDO, data can be transferred in the way of event- control or periodic. Because PDO transmission does not need response, and PDO length can be less than 8-bytes, so the transmission is fast.

3.3.1 PDO Transmission Framework

PDO transmission follows the producer-consumer model, that is, TPDOs generated by producers in CAN-bus network can be received by one or more consumer RPDOs in the network according to COB-ID, the transmission model is shown in the following figure:

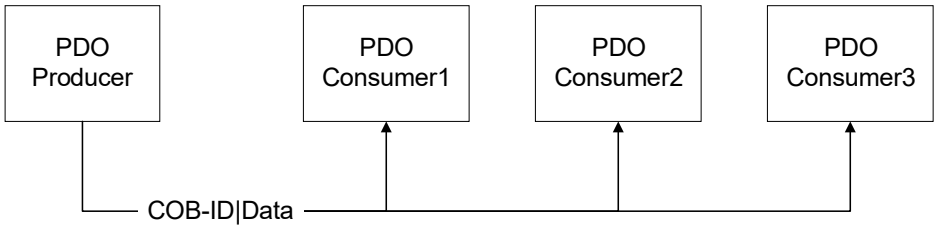


Figure 3-6 PDO Transmission model

3.3.2 PDO Object

According to the difference between receiving and sending, PDO can be divided into RPDO and TPDO. Each PDO is determined by the communication parameters and mapping parameters. The controller uses 4 RPDOs and 4 TPDOs to realize PDO transmission. The related objects are listed as follows:

Table 3-3

PDO		Identifier(COB-ID)	Communication object	Mapping object
RPDO	1	0x200+Node ID	1400_h	1600_h
	2	0x300+NodeID	1401_h	1601_h
	3	0x400+NodeID	1402_h	1602_h
	4	0x500+NodeID	1403_h	1603_h
TPDO	1	0x180+NodeID	1800_h	$1A00_h$
	2	0x280+NodeID	1801_h	$1A01_h$
	3	0x380+NodeID	1802_h	$1A02_h$
	4	0x480+NodeID	1803_h	$1A03_h$

3.3.3 PDO Communication Parameters

1 PDO Identifier COB-ID

PDO Identifier contains control bits and identification data to determine the bus priority of the PDO. COB-ID is in sub-index 01 of the communication parameters (RPDO：1400_h~1403_h，TPDO： 1800_h~1803_h) , and the highest bit determines whether the PDO is valid.

For example: for the motor with node address 1,the COB-ID in TPDO1 is 80000181_h, indicating that the PDO is disabled and in an invalid state. When its COB-ID is 181_h, indicating that the PDO is activated.

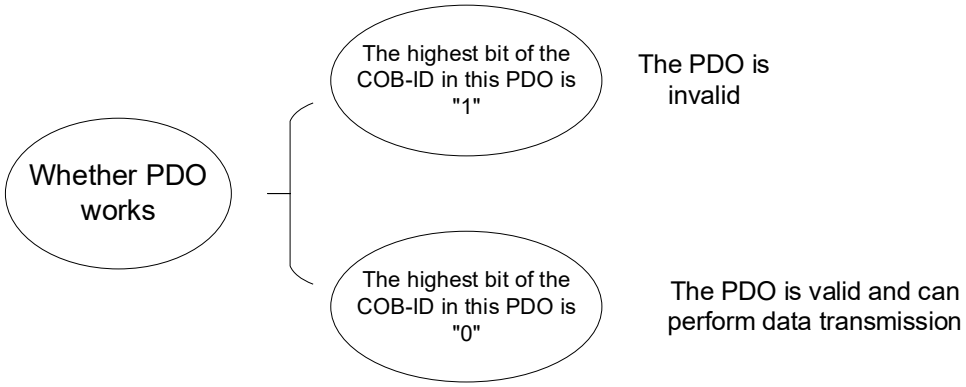


Figure 3-7

2 Transmission type of PDO

The transmission type of PDO is in sub-index 02 of the communication parameters (RPDO：1400_h~1403_h ， TPDO：1800_h~1803_h), which determines the PDO transmission mode.

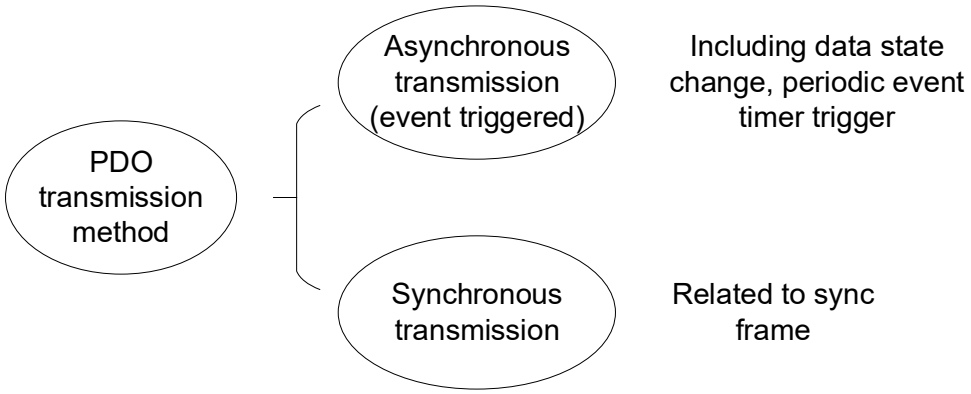


Figure 3-8 PDO transmission method

The transmission trigger type is as follows:

Table 3-4

Transfer type value	synchronization		Synchronization (event triggered)
	Cycle circulation	Acyclic non circulation	
0		√	
1-240	√		
241-253	Reserve		
254、255			√

- When the transmission type of TPDO is 0, if the mapping data changes and a synchronous frame is received, the TPDO is sent;

  When the transmission type of TPDO is from 1 to 240 and the corresponding number of synchronous frames are received, the TPDO is sent;

  When the transport type of TPDO is 254 or 255, the TPDO is sent when the mapping data

changes or the event timer arrives.

- When the transmission type of RPDO is 0-240, as long as a synchronous frame is received, the latest data of the RPDO will be updated to the application; when the transmission type of RPDO is 254 or 255, the received data will be directly updated to the application.

3 Prohibition Time

The prohibition time is for TPDO, which is stored in the sub index 03 of the communication parameter (1800_h ~ 1803_h), so as to avoid the problem that data with lower priority cannot compete for the bus due to the large amount of PDO data with higher priority and the continuous occupation of the bus. The parameter is defined as a 16-bits unsigned integer in 100us. After setting the value, the transmission interval of the same TPDO must not be less than the corresponding time of the parameter.

For example, if the prohibition time of TPDO1 is set to 100, the transmission interval of TPDO1 will not be less than 10ms.

4 Event Timer

For asynchronous transmission (transmission type is 254 or 255) TPDO, define event timer, which is in sub index 05 of communication parameter (1800_h ~ 1803_h). The event timer can also be regarded as a trigger event, which will trigger the corresponding TPDO transmission. If other events such as data changes occur during the timer cycle, TPDO will also be triggered and the event counter will be reset immediately. The event timer parameter is defined by a 16-bit unsigned integer,unit:1ms.

3.3.4 PDO Mapping

PDO mapping parameters contain pointers to process data corresponding to PDO to be sent or received by PDO, including index, sub index and mapping object length. Each PDO data length can be up to 8-bytes, and one or more objects can be mapped at the same time. Sub index 0 records the number of objects mapped by this PDO, and sub indexes 1-8 are the mapping contents. The mapping parameters are defined as follows:

Table 3-5

Bit	31-16（2 bits）	15-8（1 bit）	7-0（1 bit）
Implication	Index	Sub index	Object length

The index and sub index jointly determine the position of the object in the object dictionary. The object length indicates the specific length of the object, which is expressed in hexadecimal, that is:

Table 3-6

Object length	Length of bits
0x08	8 bit
0x10	16 bits
0x20	32 bits

For example: RPDO1 maps two parameters, 6040_h and 6060_h; then the total mapping length is 3 bytes (2 + 1), that is, the data segment of RPDO1 has 3 bytes during the transmission, the mapping relationship is shown in the following figure:

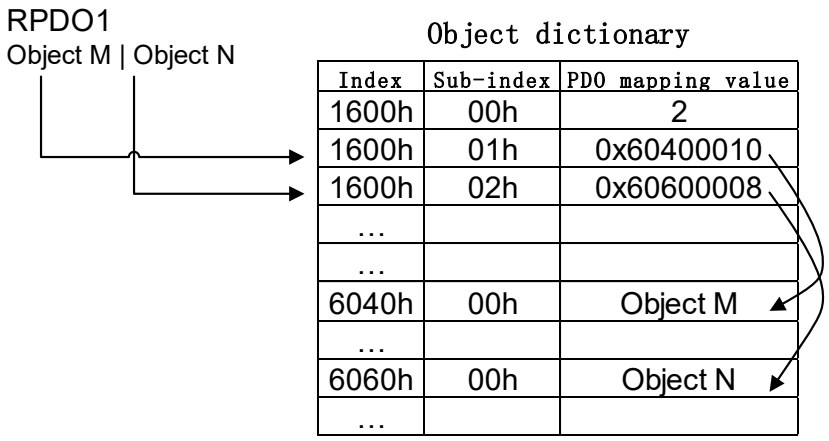


Figure 3-9 RPDO1 mapping relationship

TPDO is mapped in the same way as RPDO, in the opposite direction. RPDO decodes the input according to the mapping relationship, and TPDO encodes the output according to the mapping relationship.

For example: TPDO1 maps two parameters, 6041_h and 6061_h; then the total mapping length is 3 bytes (2 + 1), that is, the data segment of TPDO1 has 3 bytes during the transmission, the mapping relationship is shown in the following figure :

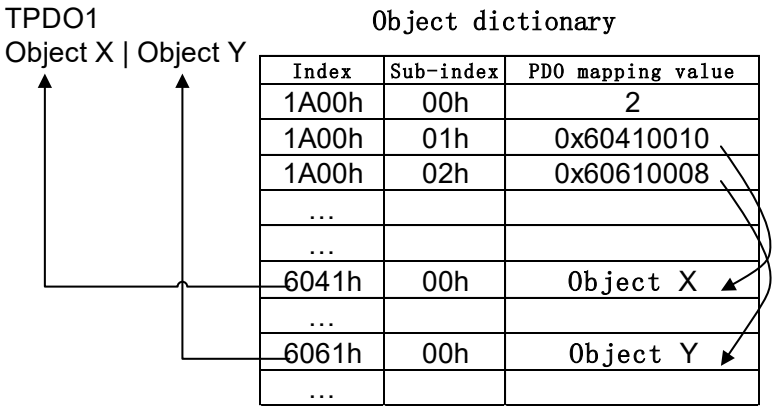


Figure 3-10

If the device supports dynamic modification of PDO mapping parameters, SDO messages can be used to configure PDO mapping parameters. This device supports dynamic modification of PDO mapping parameters. The following process must be followed to modify PDO mapping:

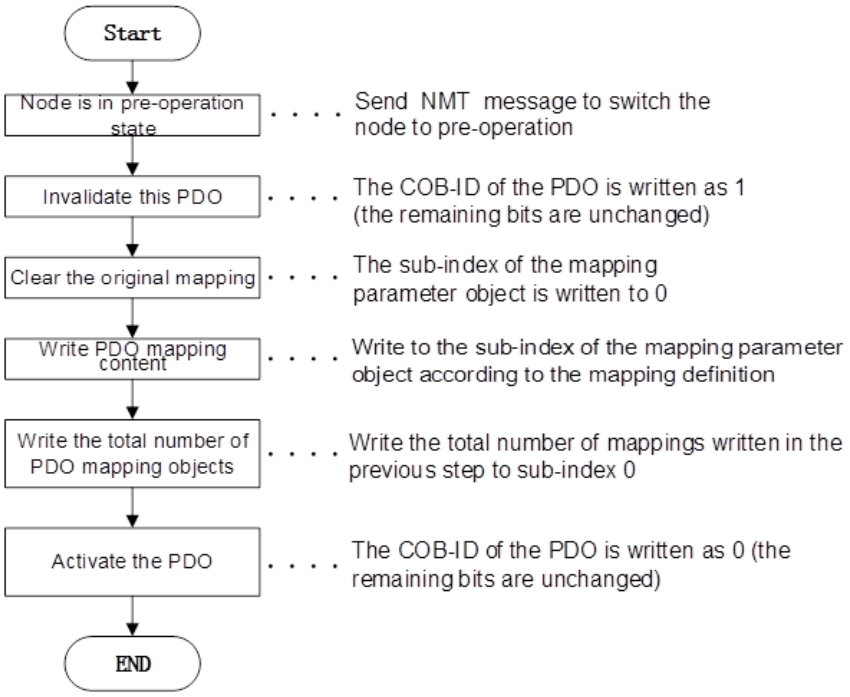


Figure 3-11

3.4 EMCY Message

When the node has errors, according to the standardization mechanism, the node will send a frame of emergency message,which follows the producer-consumer model. After the node error is sent, other nodes in the CAN network can choose to handle the error. This device is only an emergency message producer and does not handle emergency messages from other nodes.

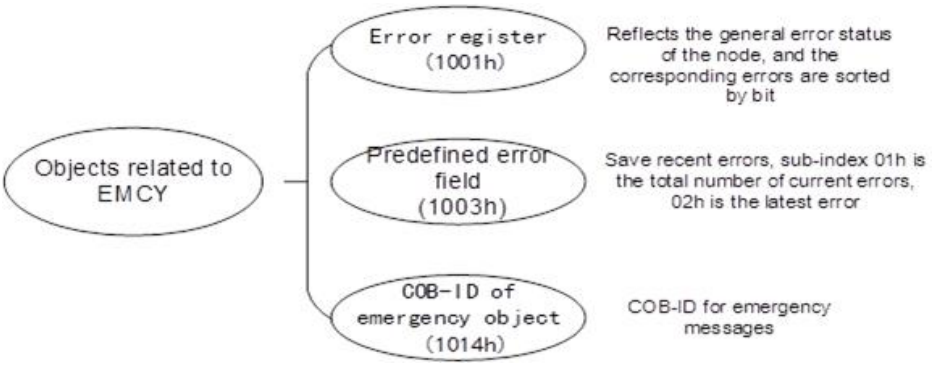


Figure 3-12

When a node is faulty, the error register and predefined error field need to be updated. The contents of emergency message shall be in accordance with the following specifications:

Table 3-7

COB-ID	Message (byte)							
	0	1	2	3	4	5	6	7
0x80+Node ID	Error code		Error register	Retain	Auxiliary byte			

Auxiliary byte displays the user specified error code. Please refer to "Fault management" for details of error code and auxiliary byte definition.

3.5 Network Management (NMT service)

The network management system (NMT) is responsible for initializing, starting and stopping the network and devices in the network, belonging to the master / slave system .There is only one NMT host in CANopen network, which can configure the whole CANopen network including itself.

3.5.1 NMT Services

CANopen performs the corresponding work according to the state machine specified in the protocol. Some of them are internal automatic conversion, and some of them must be converted by NMT host sending NMT message, as shown in the figure below:

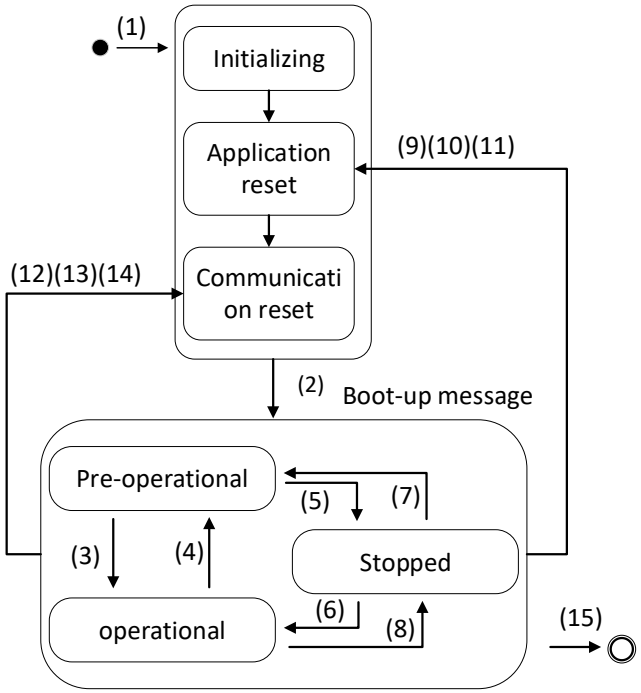


Figure 3-13

(1) power on; (2) automatically switch to Pre-operational state; (3) switch to Operational state; (4) switch to Pre-operational state;

(5), (8) switch to the stopped state; (9), (10), (11) switch to the Application reset state;

(12), (13), (14) switch to Communication reset state; (15) power failure or hardware reset.

The conversion in the figure above is realized by NMT message, and only NMT host can send NMT control message. The format of message message is shown in the table below:

Table 3-8

COB-ID	Data/Byte	
	0	1
0x000	Command word	Node ID

The COB-ID of NMT message is fixed to "0x000". The data area is composed of two bytes: the first byte is the command word, indicating the control function of the frame as shown in the following table; the second byte is the address of CANopen node, when it is "0", it is a broadcast message, and all the slave devices in the network are valid.

Table 3-9

Command word	Description
0x01	Switch to Operational state
0x02	Switch to Stopped state
0x80	Switch to Pro-operational state
0x81	Reset node command
0x82	Reset communication command

After the device is powered on, it will automatically switches the Initializing state, including initializing, resetting nodes and resetting communication. After that, the device sends boot-up and automatically enters the Pre-operational state, which is the main configuration node state. After the configuration is completed, the node needs the NMT host to send the NMT message to enter the Operational state. The Operational state is the normal working state of CANopen, and all modules should work normally.

The CANopen services supported in various NMT states are shown in the following table:

Table 3-10

Service type	（Pre-op）	（op）	（Stopped）
Process data object (PDO)	No	Yes	No
Service data object(SDO)	Yes	Yes	No
Synchronous message	Yes	Yes	No
Emergency message(EMCY)	Yes	Yes	No
Network Management System	Yes	Yes	Yes
Error control	Yes	Yes	Yes

3.5.2 Error Control

NMT error control is mainly used to detect whether the equipment in the network is online and the status of the equipment, including node protection, life protection and heartbeat. This device only supports heartbeat.

The heartbeat mode adopts the producer-consumer model. CANopen device can send heartbeat message according to the period set by the producer heartbeat interval object 1017_h, unit: ms. The format of heartbeat message is as follows:

Table 3-11

COB-ID	Data (1bit)
0x700+Node ID	Node state

Node status: 0x04: Stopped state
0x05: Operational operation state
0x7F: Pre-operational state

3.5.3 Start-up Agreement

After starting the power supply or resetting, the motor will report the initialization phase through a start message, and then automatically enter the Pre-operational state.

The start message is almost the same as the heartbeat message, except that NMT status is not sent, but 0 is sent. Message composition of start message:

Table 3-12

COB-ID	Data(1 Bit)
0x700+Node ID	0

3.6 System Setting

In order to make the integrated motor connect to CANopen field bus network accurately, it is necessary to set the related objects of the integrated motor driver.

3.6.1 Baud-Rate Setting

The CAN communication baud-rate is changed by setting object $200C_h$: 03_h, and the transmission includes the value of CAN communication baud-rate to be set.

Table of setting value and baud-rate:

Table 3-13

Value	Baud rate
0x00	10 kBit/sec
0x01	20 kBit/sec
0x02	50 kBit/sec
0x03	100 kBit/sec
0x04	125 kBit/sec
0x05	250 kBit/sec
0x06	500 kBit/sec
0x07	800 kBit/sec
0x08	1 MBit/sec

3.6.2 Node ID

Change the node ID through the setting of object $200C_h$: 02_h. The setting range is 1-127.

3.6.3 Connection of Communication Line

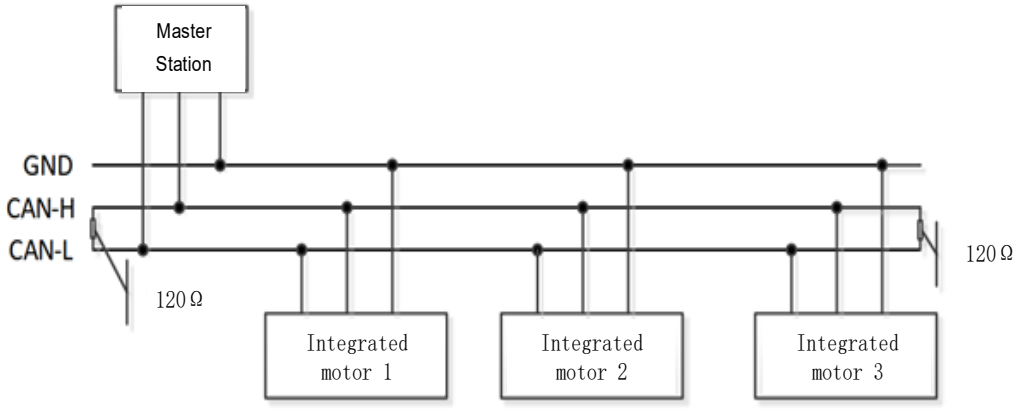


Figure 3-14

4 Control of Equipment

4.1 The CiA402 State Machine

The operating state of integrated low-voltage motor is switched by a state machine, which is defined in the CiA402 standard. The motor drive is controlled by the control word 6040_h, and the actual state of the device can be read back from the status word object 6041_h.

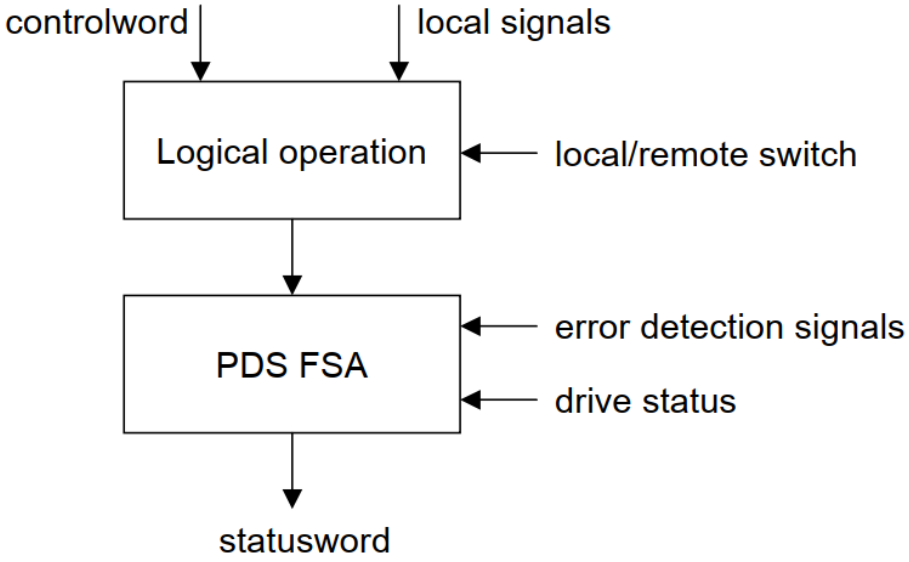


Figure 4-1

4.1.1 The State Transition Diagram

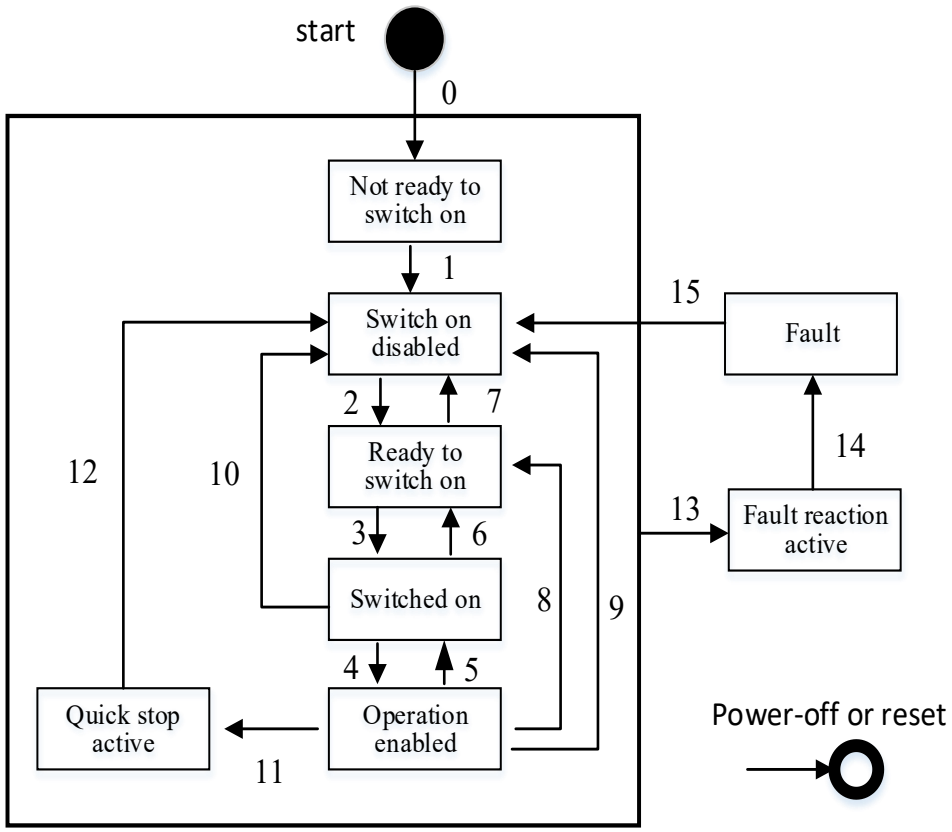


Figure 4-2

The motor states corresponding to each state are in this table below:

Table 4-1

Function	FSA states							
	Not ready to switch on	**Switch on disabled**	**Ready to switch on**	**Switched on**	**Operation enabled**	**Quick stop active**	**Fault reaction active**	**Fault**
Brake applied, if present	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes
Low-level power applied	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High-level power applied	Yes/No	Yes/No	Yes/No	Yes	Yes	Yes	Yes	Yes/No
Drive function enabled	No	No	No	No	Yes	Yes	Yes	No
Configuration allowed	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

The control word and switching of state machine are in this table below:

Table 4-2

Command	Bits of the control word					Transitions
	Bit7	Bit3	Bit2	Bit1	Bit0	
Shutdown	0	×	1	1	0	2,6,8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3+4（NOTE）
Disable voltage	0	×	×	0	×	7，9，10，12
Quick stop	0	×	0	1	×	7，10，11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4，16
Fault reset		×	×	×	×	15

NOTE: Automatic transition to Enable operation state after executing Switch on state functionality.

The status word in each state are as follows:

Table 4-3

Status word	PDS FSA state
xxxx xxxx x0xx 0000b	Not ready to switch on
xxxx xxxx x1xx 0000b	Switch on disabled
xxxx xxxx x01x 0001b	Ready to switch on
xxxx xxxx x01x 0011b	Switched on
xxxx xxxx x01x 0111b	Operation enabled
xxxx xxxx x00x 0111b	Quick stop active
xxxx xxxx x0xx 1111b	Fault reaction active
xxxx xxxx x0xx 1000b	Fault

Notes: x means that this bit does not been care,bit 8 ~ bit15 has different meaning in different modes. Please check its specific status in each mode.

4.1.2 Control Word

The bits of the control word are defined as follows:

15	10	9	8	7	6	3	2	1	0
11					4				
ms	r	oms	h	fr	oms	eo	qs	ev	so
MSB									LSB

LEGEND: ms = manufacturer-specific; r = reserved; oms = operation mode specific; h = halt; fr = fault reset; eo = enable operation; qs = quick stop; ev = enable voltage; so = switch on

Notes：

It is meaningless to assign each bit of the control word individually,It must be combined with other bits to form a control instruction. Bit 0 ~bit 3 and bit 7 have the same meaning in each mode, Only by sending commands sequentially can the motor drive be guided into the predicted state according to the switching process of the CiA402 state machine,and each command corresponds to 1 definitive state. Bit4 ~ bit6 are related to each operation mode (please check the control instructions in different modes)

4.1.3 Status Word

The bits of the status word are defined as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms		oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Table 4-4

Bit	Description
0	ready to switch on
1	switched on
2	operation enabled
3	fault
4	voltage enabled
5	quick stop
6	switch on disabled
7	warning
8	manufacturer specific
9	remote
10	target reached
11	internal limit active
12-13	operation mode specific
14-15	manufacturer specific

Notes：

It is meaningless to assign each bit of the status word individually,It must be combined with other control bits to form the current state feedback of the motor. In each stepping mode, bit 0-bit9 has the same meaning,by which the motor feeds back a certain status after sending commands by the control word 6040_h sequentially. Besides,bit10, bit11 and bit15 have the same meaning in each mode,by which the motor feeds back its status after a certain operation mode is executed.

4.1.4 The Stop Mode

The stop mode is the action of the state machine which leaving the "Operation enabled" state, corresponding settings are as follows:

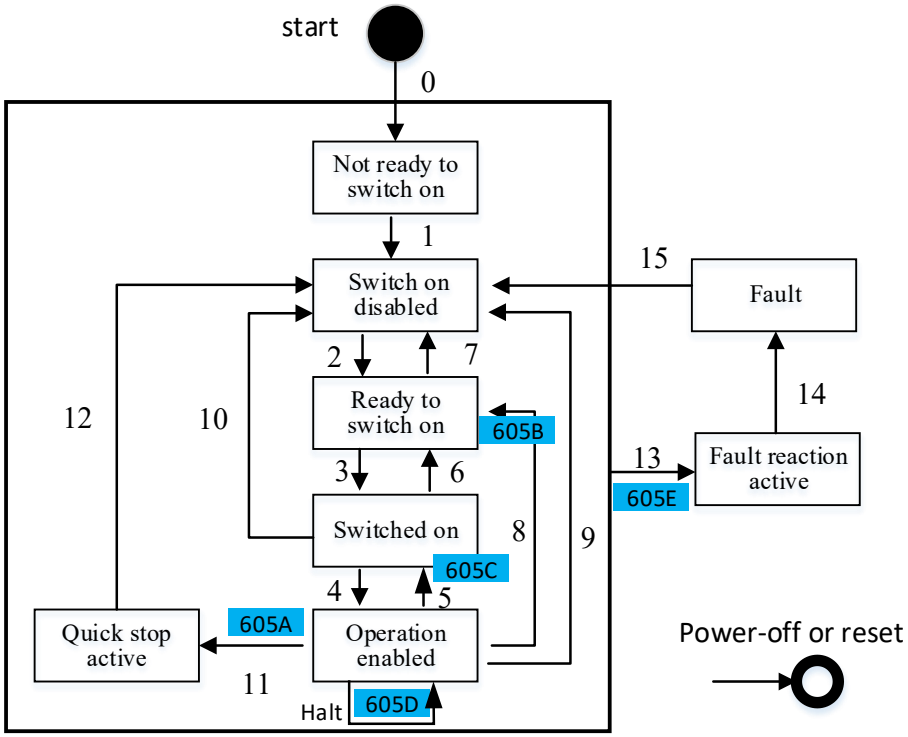


Figure 4-3

- **Fault**

  When a error or a warning occurs, automatically the motor switches to the Fault state which is related to $605E_h$(Fault response option code). When $605E_h$= -1, the fault reaction is related to the reaction code in the fault code which is configurable.When $605E_h$ > 0, the fault reaction is selected by the object dictionary $605E_h$. The relationship between the setting value in $605E_h$ and the fault reaction is as follows:

Table 4-5

Setting Value	Stop Mode
-1	The stop mode is related to fault reaction option code in alarm code. See "9.1 alarm code" for details
0x00	Slow down freely and keep operating freely.
0x01	Slow down on slow down ramp set in 6084_h.
0x02	Slow down on quick slow down ramp set in 6085_h.

- **Quick Stop**

  In non-fault states, when the control word 6040_h = 0x02, Quick stop is executed.The Quick stop mode is selected by the object dictionary $605A_h$.The meaning of the setting value in

605A$_h$ is shown in the table below:

Table 4-6

Setting Value	Stop Mode
0x00	Slow down freely and keep operating freely.
0x01	Slow down on slow down ramp set in 6084h,and keep operating freely.
0x02	Slow down on slow down ramp set in 6085$_h$,and keep operating freely.

- **Halt**

Bit8 of the control word 6040$_h$ is the functional bit of Halt. When the bit8 changes from 0 to 1, Halt stop is executed. The Halt stop mode is chose by the object dictionary 605D$_h$.The meaning of the setting value of 605D$_h$ is shown in the table below:

Table 4-7

Setting Value	Stop Mode
0x00	Reserved
0x01	Slow down on slow down ramp set in 6084$_h$,and stay in Clamped state.
0x02	Slow down on slow down ramp set in 6085$_h$,and stay in Clamped state.

- **Disable Operation**

The Unpower mode is chose by object dictionary 605C$_h$.The meaning of the setting value of 605C$_h$ is shown in the table below:

Table 4-8

Setting Value	Slow Down Mode
0x00	Slow down freely and keep operating freely.
0x01	Slow down on slow down ramp set in 6084$_h$,and keep operating freely.

- **Shutdown**

The Power off mode is chose by object dictionary 605B$_h$.The meaning of the setting value of 605B$_h$ is shown in the table below:

Table 4-9

Setting Value	Slow Down Mode
0x00	Slow down freely and keep operating freely.
0x01	Slow down on slow down ramp set in 6084$_h$,and keep operating freely.

5 Unit Conversion

In order to meet application requirements of all industries, facilitate users to define their own instruction units, the units-conversion module in the device can convert any User Unit (UU) into internal unit(Encoder Increment:shorter form is Inc).

5.1 $608F_h$:Resolution of position encoder

This object shall indicate increment and rotation number of the encoder installed. The formulas to calculate the resolution of position encoder are as follows:

$$\text{Resolution of position encoder}=\frac{\text{increment of encoder}(608F_h:01_h)}{\text{motor turns }(608F_h:02_h)}$$

The object description is specified in the table below:

Table 5-1

Attribution	Description
Index	$608F_h$
Name	Resolution of position encoder
Data structure	Array
Data type	Unsigned32

The sub-indexes description is specified in the table below:

Table 5-2

Attribution	Description
Sub-index	01_h
Description	Increment of encoder
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	4000
Sub-index	02_h
Description	Motor turns
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	1

5.2 6091_h:Transmission Ratio

This object shall indicate the numbers of motor revolutions and drive-shaft revolutions. The formula to calculate Gear ratio is as follows:

$$\text{Gear ratio}=\frac{\text{Motor shaft revolutions}(6091_h:01_h)}{\text{Drive shaft revolutions}(6091_h:02_h)}$$

The object description is specified in the table below:

Table 5-3

Attribution	Description
Index	6091_h
Name	Gear ratio
Data structure	Array
Data type	Unsigned32

The sub-indexes description is specified in the table below:

Table 5-4

Attribution	Description
Sub-index	01_h
Description	Motor shaft revolutions
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	1
Sub-index	02_h
Description	Drive shaft revolutions
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	1

Gear ratio is used to specify the proportional relationship between the drive shaft displacement and the motor shaft displacement.

- Relationship between shaft position feedback (encoder unit) and drive shaft position feedback (instruction unit):

  shaft position feedback=drive shaft position feedback × Gear ratio

- Relationship between motor shaft speed(rpm) and drive shaft speed (unit /s):

$$\text{motor shaft speed (rpm)} = \frac{\text{drive shaft speed} \times \text{gear ratio}\ (6091_h)}{\text{encoder resolution s}} \times 60$$

- Relationship between motor acceleration (rpm/s) and shaft speed (unit /s^2):

$$\text{motor shaft acceleration} = \frac{\text{drive shaft accelerati on} \times \text{gear ratio}(6091_h)}{\text{encoder resolution s}} \times 60$$

5.3 607Eh: Polarity

This object dictionary is configured to determine whether the position command and velocity command are multiplied by 1 or -1. Position polarity is only used in Profile position mode and Cyclic sync position mode, and has no effect in Homing mode. Velocity polarity is only used in Profile velocity mode and Cyclic sync velocity mode. The specific definitions are as follows:

Table 5-5

bit		
7	6	5~0
Position polarity	Velocity polarity	reserved (0)

The bits in the table above: 0 is multiplied by 1, 1 is multiplied by -1. For example, if Position polarity = 0, position demand value will be multiplied by 1. The object description is specified in the table below:

Table 5-6

Attribution	**Description**
Index	$607E_h$
Name	Polarity
Data structure	Variable
Accessibility	rw
Data type	Unsigned8
Default value	0

5.4 Example

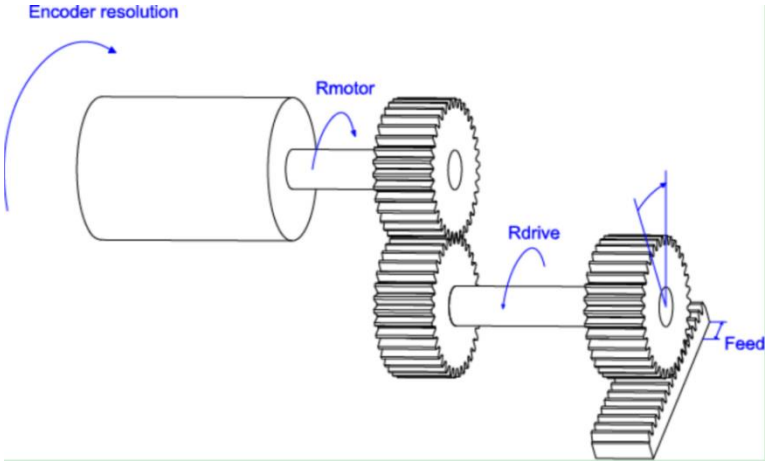


Figure 5-1

As shown in the figure above, R_{motor} is the motor shaft and R_{drive} is the drive shaft. Suppose: the Encoder resolution = 10000 as the motor shaft rotates 1 revolution; R_{motor} rotates 3 revolutions when the R_{drive} rotates 1 revolution; from the conditions above, it can be seen that as the drive shaft rotates 1 revolution, the motor needs to operate 3 * 10000 encoder codes, which means that the motor rotates 3 revolutions. Parameters can be calculated according to the following formulas:

position encoder resolution $608F_h$:

$$\text{Resolution of position encoder} = \frac{\text{Increment of encoder}(608F_h:01_h)}{\text{Motor turns }(608F_h:02_h)} = \frac{10000}{1} = 10000$$

gear ratio 6091_h:

$$\text{Gear ratio} = \frac{\text{Motor shaft revolutions}(6091_h:01_h)}{\text{Drive shaft revolutions}(6091_h:02_h)} = \frac{3}{1} = 3$$

6 Operation Mode

6.1 Overview of Operation Mode

The control mode of integrated-servo is divided into CiA402 mode and NiMotion mode, 2002_h:01_h (control mode option) is used to determine whether the servo is in CiA402 mode or NiMotion mode. The object description is as follows:

Table 6-1

Attribution	Description
Index	2002_h
Name	Basic control parameter
Data structure	Record
Data type	-

Sub index 01_h is to determine the mode:

Table 6-2

Sub-Index	01_h
Description	Control mode option
Accessibility	rw
PDO mapping	No
Data type	Unsigned8
Default value	0
Meaning of Value	0x00: CiA402 mode 0x01: Manufacturer position mode 0x02: Manufacturer velocity mode 0x03: Manufacturer torque mode

After selecting CiA402 mode, 6060_h needs to be set to select specific motion mode (control mode selection, invalid operation state setting). The object description is as follows:

Table 6-3

Sub-Index	6060_h
Description	Mode Option
Accessibility	rw
PDO mapping	NO
Data type	Unsigned8
Default value	0x01
Sub-Index	-
Meaning of Value	0x00: Undefined 0x01: Profile position mode(PP) 0x02: Velocity mode (VM) 0x03: Profile velocity mode

Meaning of Value	0x04: Torque profile mode 0x05: Undefined 0x06: Homing mode (HM) 0x07: Interpolated position mode(IP) 0x08: Cyclic sync position mode(CSP) 0x09: Cyclic sync velocity mode (CSV) 0x0A: Cyclic sync Torque mode (CST)

6.2 Position Control Function

6.2.1 Structure Chart

Target Following Error

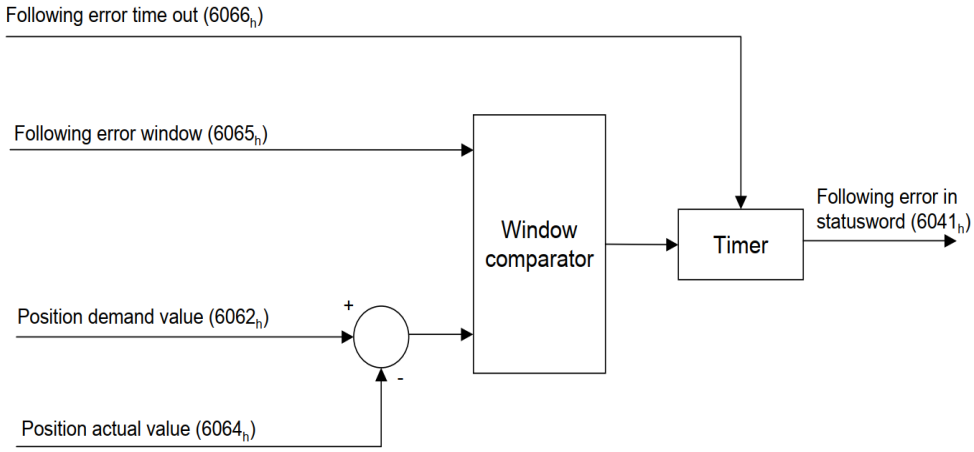


Figure 6-1

Target Reached

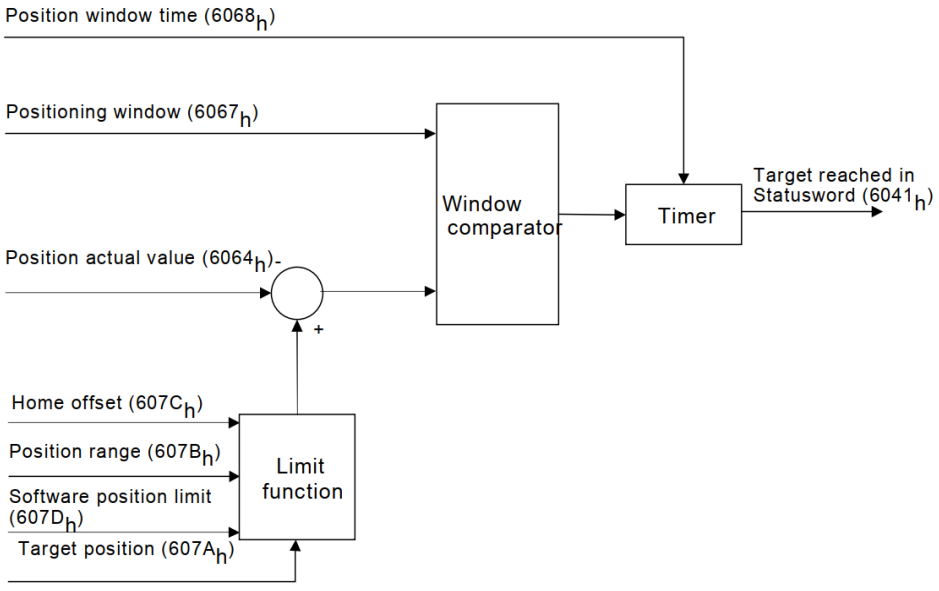


Figure 6-2

6.2.2 Objects related

Target following error

Table 6-4

Object Index	Description
6041_h（Status word）	Status word
6062_h（Position demand value）	Current position demand value inside the drive (user unit)
6064_h（Position actual value）	Current position actual value feedback (user unit)
6065_h（Following Error Window）	This object shall indicate the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients.
6066_h（Following Error Time Out）	Following error time out (unit：ms)

Target Reached

Table 6-5

Object Index	Description
6041_h (Status word)	Status word
6064_h（Position actual value）	Current position value feedback (user unit)
6067_h（Position window）	The configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position shall be regarded as having been reached.
6068_h（Position window time）	Position window time(unit：ms)
$607A_h$（Target position）	Target position(user unit)
$607B_h$（Position range limit）	Position range limit(user unit)
$607C_h$（Home offset）	Difference between the 0-position in the application and the machine home position (user unit)
$607D_h$（Software position limit）	Target position limit (user unit)

6.2.3 Function Description

Target Following Error

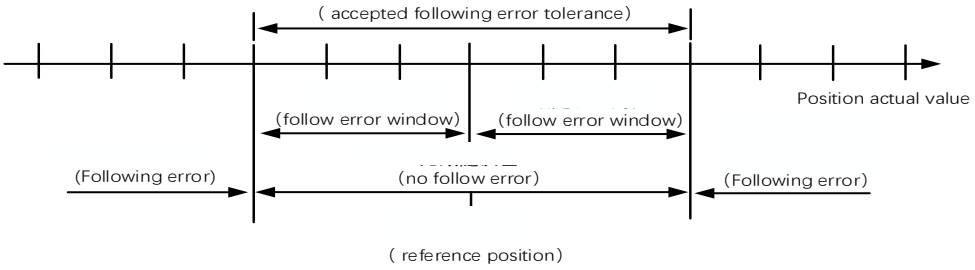


Figure 6-3

Following error refers to the deviation between Position demand value (6062_h) and Position actual value (6064_h).If following error is continually greater than value set in 6065_h (following error windows) within the time set in 6066_h (Following error timeout) (as shown in Figure 6.2, the following error exceeds acceptable range), Bit13 (following error) in 6041_h(status word) will be set to 1,the sequence diagram is shown below:

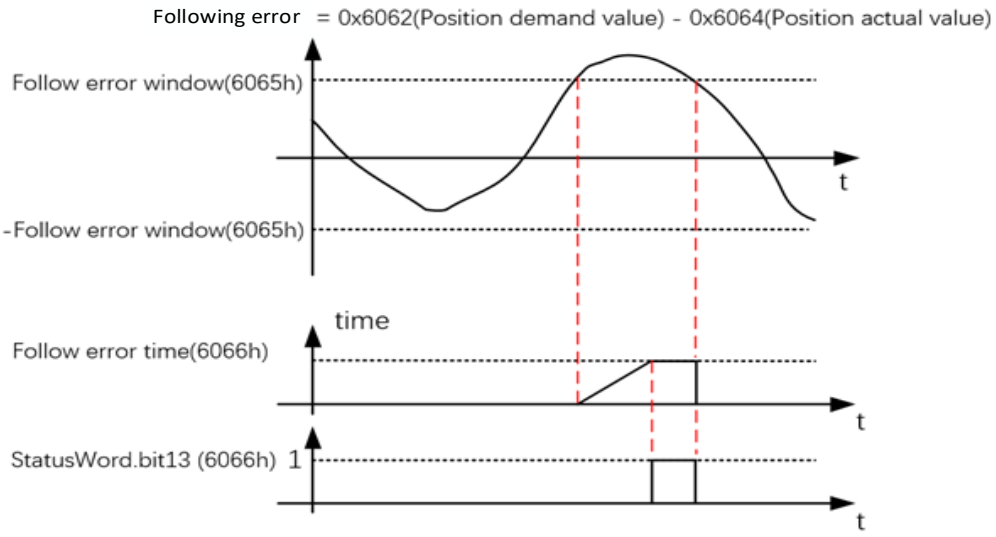


Figure 6-4

Target Reached

Figure of Error Acceptable:

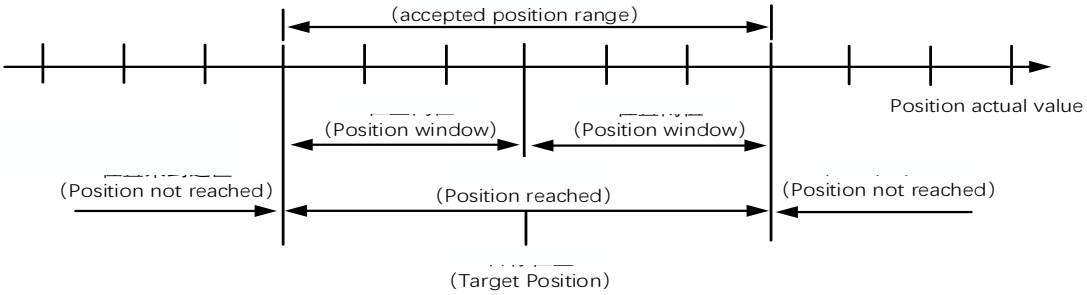


Figure 6-5

Position deviation is the deference between the Target position in $607A_h$ and the Position actual value in 6064_h. If the deviation is in the acceptable range (as shown in the above figure) stablely and span the time set in 6068_h (position windows time), Bit10 (target reached) of the status word will be set to 1, that is, the target position is reached. The sequence diagram is as follows:

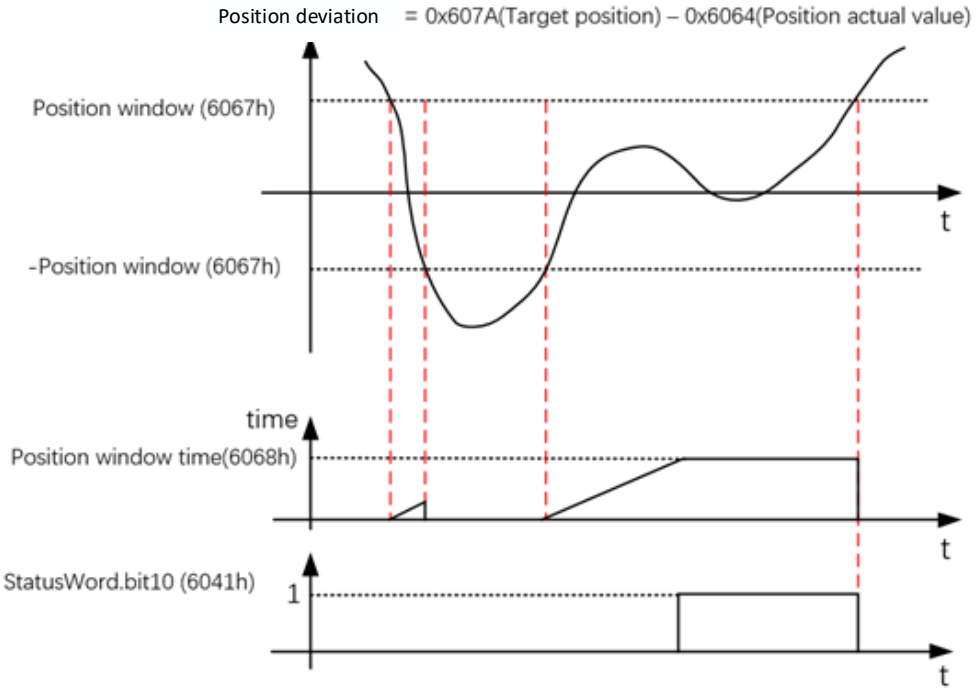


Figure 6-6

6.3 Profile Position Mode（PP）

This mode is mostly used in point-to-point positioning applications. In this mode, the master computer gives the target position (absolute or relative), speed,acceleration and deceleration of the position curve, the track generator in servo will generate target position curve command according to the settings, and the drive will complete the position,speed and torque control internally.

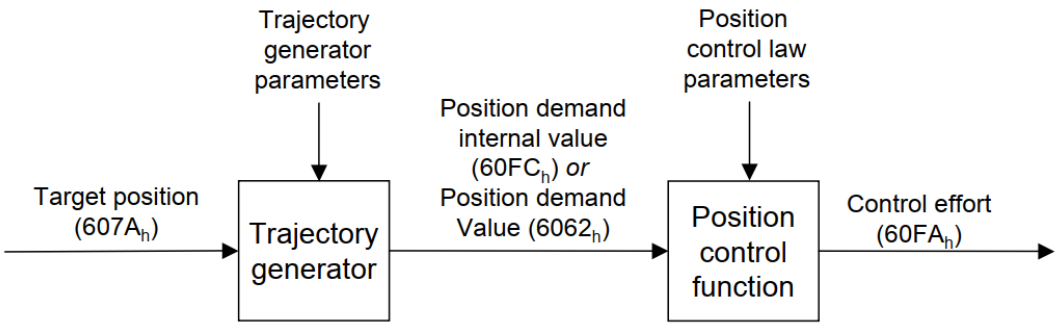


Figure 6-7

6.3.1 Structure Chart

The following figure introduces the detailed structure of the trace generator:

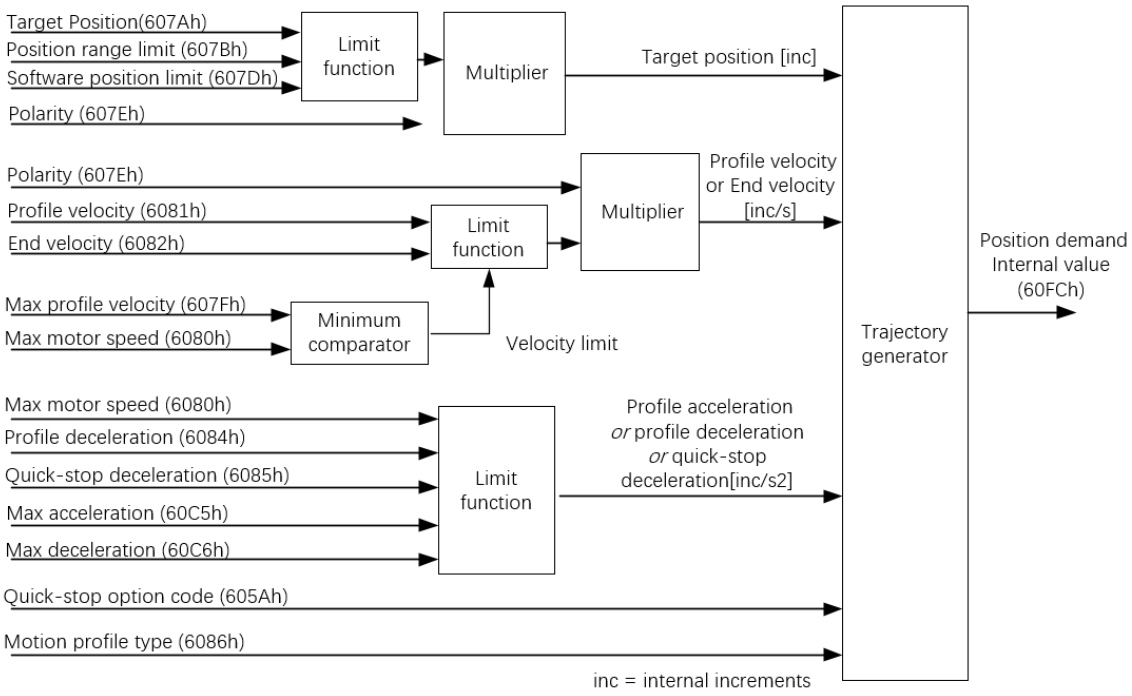


Figure 6-8

6.3.2 Object Related

Pay attention to the following objects in this mode :

Table 6-6

Object Index	Description
$605D_h$ (Halt option code)	Halt option
$605A_h$ (Quick stop option code)	Quick stop option
6062_h (Position Demand Value)	Current position demand value inside the drive (user unit)
6063_h (Position Actual Enc Value)	Current position actual value feedback (encoder unit)
6064_h (Position Actual User Value)	Current actual position (user unit)
$607A_h$ (Target Position)	Target position (user unit)
$607B_h$ (Position Range Limit)	Position range limit (user unit)
$607D_h$ (Software Position Limit)	Target position limit (user unit)
$607E_h$ (Polarity)	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
$607F_h$ (Max Profile Velocity)	Maximum velocity (user unit /s), to limit the velocity
6080_h (Max motor speed)	Max motor speed(rpm)

Object Index	Description
6081_h (Profile Velocity)	Maximum velocity during uniform motion in this profile operation, (user unit / s), I.e. the velocity at the end of the acceleration slope during the positioning.
6082_h (End Velocity)	The velocity when reaching the target position(user unit/s), and the velocity at the end of the slope. Usually, this object is set to 0, so that the velocity when reaching the target position is just down to 0.
6083_h (Profile Acceleration)	The acceleration during the profile operation(user unit / s^2), limited by $60C5_h$.
6084_h (Profile Deceleration)	The deceleration during the profile operation operation, (user unit / s^2), limited by $60C6_h$.
6085_h (Quick Stop Deceleration)	Deceleration during "quick stop" in cia402 state machine.
6086_h (Motion Profile Type)	Choose the type of velocity curve. If the value is "0", the impact (jerk) will be not limited; If the value is "3", the impact (jerk) will be limited only by the value in $60a4_h$: 01_h-02_h in this equipment.
$60C5_h$ (Max Acceleration)	Max acceleration(user unit / s^2)
$60C6_h$ (Max Deceleration)	Max deceleration(user unit / s^2)
$60F2_h$ (Positioning option code)	Location options
$60FC_h$ (Position Demand internal value)	Output of track generator, i.e.the real time position instruction which is internal planned(encoder unit)

The following figure shows the effect of the velocity, acceleration and impact parameters:

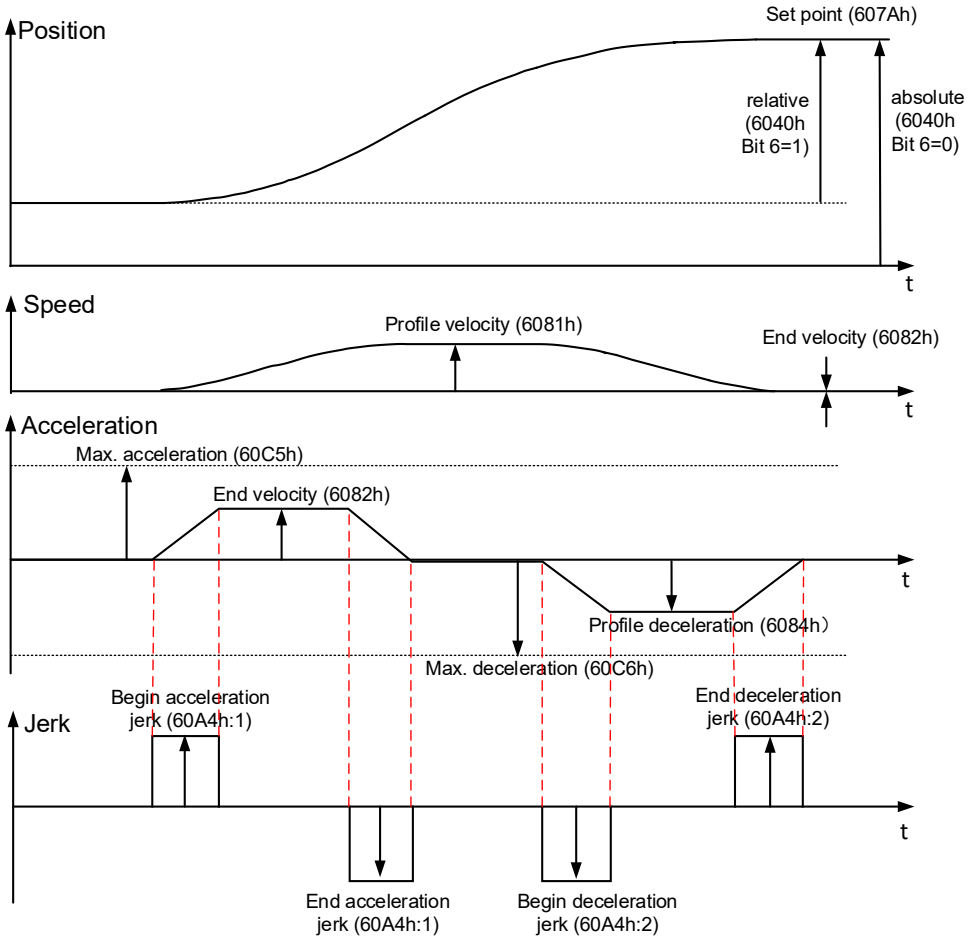


Figure 6-9

6.3.3 Control Instruction and Status Information

Enable

As this mode is enabled, the value in object 2002_h:01_h and object 6060_h (Modes of Operation)must be set to "0" and "1" respectively.

Control Word

15 10	9	8	7	6	5	4	3 0
(see 4.1.2)	Change on set-point	Halt	(see4.1.2)	abs/rel	Change set immediately	New set-point	(see4.1.2)
MSB							LSB

In the Profile position mode, the following bits in object 6040_h (control word) have special functions.

Table 6-7

Bit 5	Bit4	Description
0	0->1	Not Update-now, the next task is started after the running task be completed
1	0-> 1	Update-now, the task triggered by bit4 will be executed immediately.

Table 6-8

Bit	Value	Description
6	0	Target position is an absolute value
	1	Target position is a relative value (depending on object $60F2_h$)
8	0	Positioning is executed or continued
	1	Motor will be stopped accordingly to halt option code (605Dh)

Status Word

15	14	13	12	11	10	9	0
(see 4.1.3)	Following error		Set-point acknowledge	(see 4.1.3)	reached Target	(see 4.1.3)	
MSB							LSB

In Profile position mode, the following bits in object 6041_h (status word) have special functions:

Table 6-9

Bit	Value	Description
10	0	Halt (Bit 8 in control word 6040_h) = 0: Target position not reached Halt (Bit 8 in control word 6040_h) = 1: Axis decelerates
	1	Halt (Bit 8 in control word 6040_h) = 0: Target position reached Halt (Bit 8 in control word 6040_h) = 1: Velocity of axis is 0
12	0	Previous set point already processed, waiting for new set point
	1	Previous set point still in process, set point overwriting shall be accepted
13	0	No following error
	1	Following error

6.3.4 Function Description

It is divided into immediate update and non immediate update to interpret the control instruction sequence:

1 Control Instruction Sequence- Immediate Update

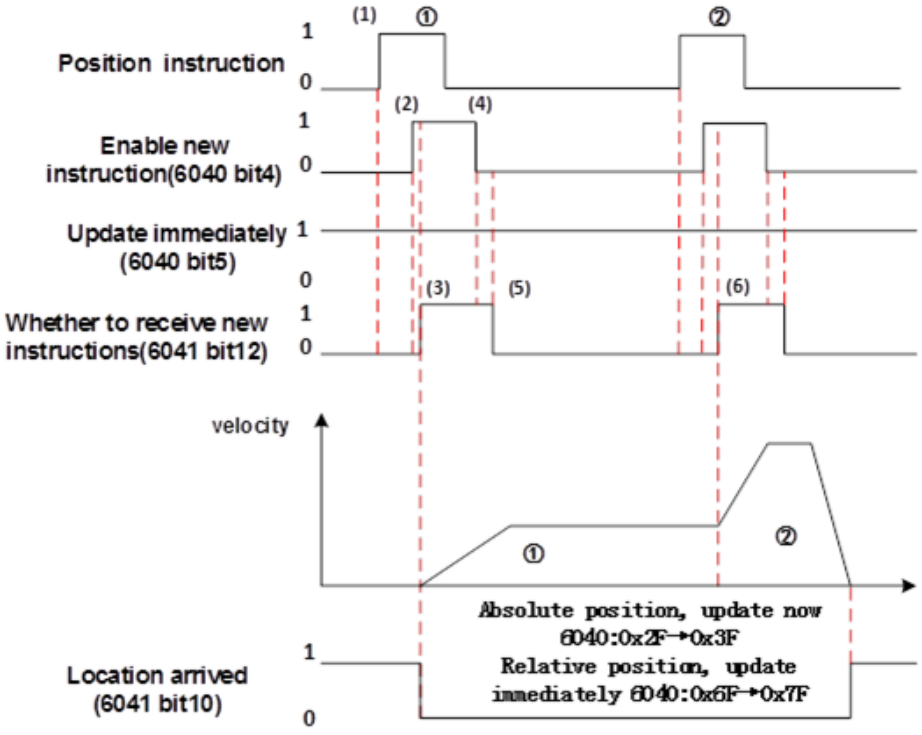


Figure 6-10

Some interprets of the notes in above process is gave in the following table:

Table 6-10

（1）	The master computer updates instruction (including Target position $607A_h$, Acceleration time 6083_h, Deceleration time 6084_h, Profile velocity 6081_h,etc.)
（2）	Change bit4 in 6040_h from 0 to 1, indicating that the slave station has new position instruction (bit5 in 6040_h is set to 1 in the immediate-updating case)
（3）	After receiving the rising edge of bit4 in 6040_h,the slave station judges whether the new position instruction can be received or not: bit12 in 6041_h being 0 ,indicates that the slave station can receive the new instruction ①; The slave station change bit12 in 6041_h from 0 to 1 after receiving the new instruction, which indicates that the instruction ① has been received, and the slave station is unable to receive new instruction. In immediate update mode, the motor executes the position command immediately once the new position command is received (bit12 in 6041_h changes from 0 to 1)
（4）	The master computer shall release position instruction only when receiving the bit12 in 6041_h from the slave station being changed from 1 to 0. And,the master computer changes bit4 in control word 6040_h from 1 to 0,indicating that there is no new position instruction at present.
（5）	The slave station changes bit12 in status word 6041_h from 1 to 0 when it detects that bit4 in control word 6040_h been changed from 1 to 0, indicating that the slave station is ready to receive new position instruction.

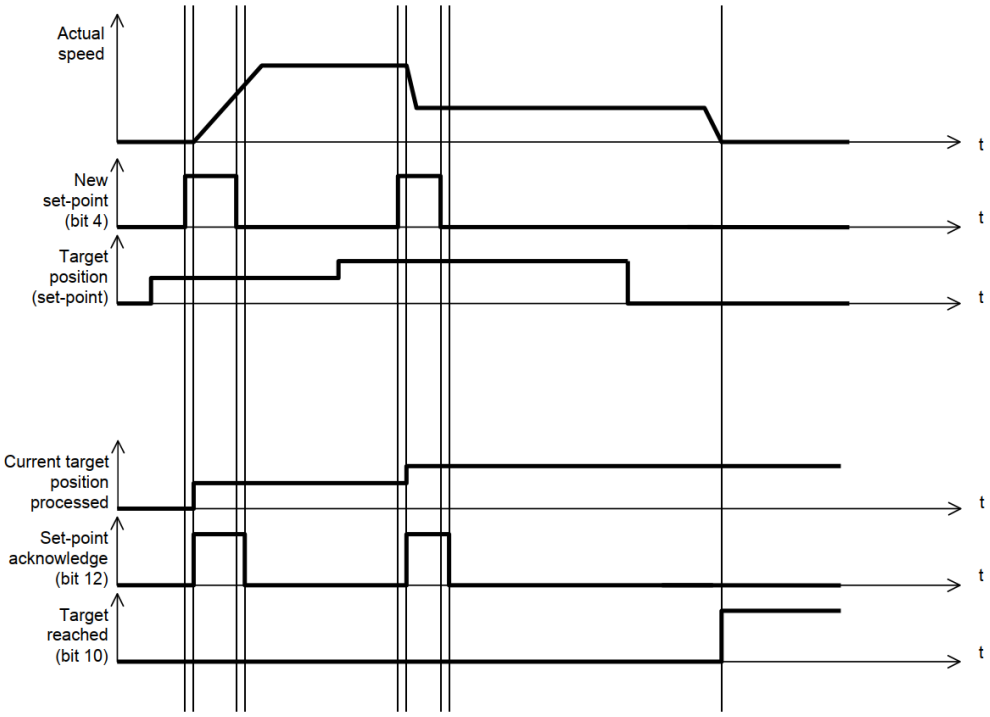


Figure 6-11

2 Control Instruction Sequence Non Immediate Update

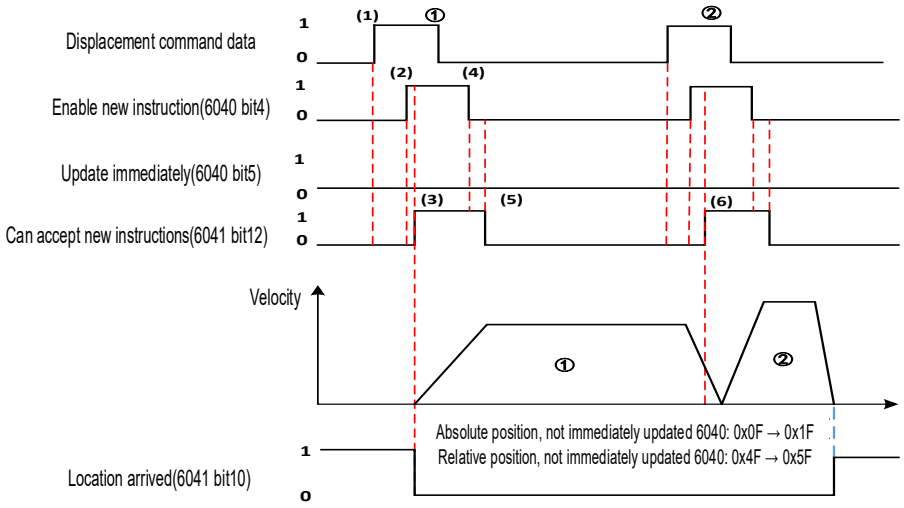


Figure 6-12

The notes (1) - (5) in the figure below are similar to the immediate-update mode. It can be seen from the comparison of position (6) in the figure,the difference is that in non immediate update mode, although the new target position ② has been accepted, it needs to reach position ① first, slow down and stop, and then move to new position ② .

In non immediate-update mode,the motor has 5 buffers, i.e. it can have 5 target positions in the sequence at the same time and the motor will operate to these positions in turn. If the buffers is full, new position can't be accepted until the buffers is empty. These buffers are cleared when the motor is stopped.

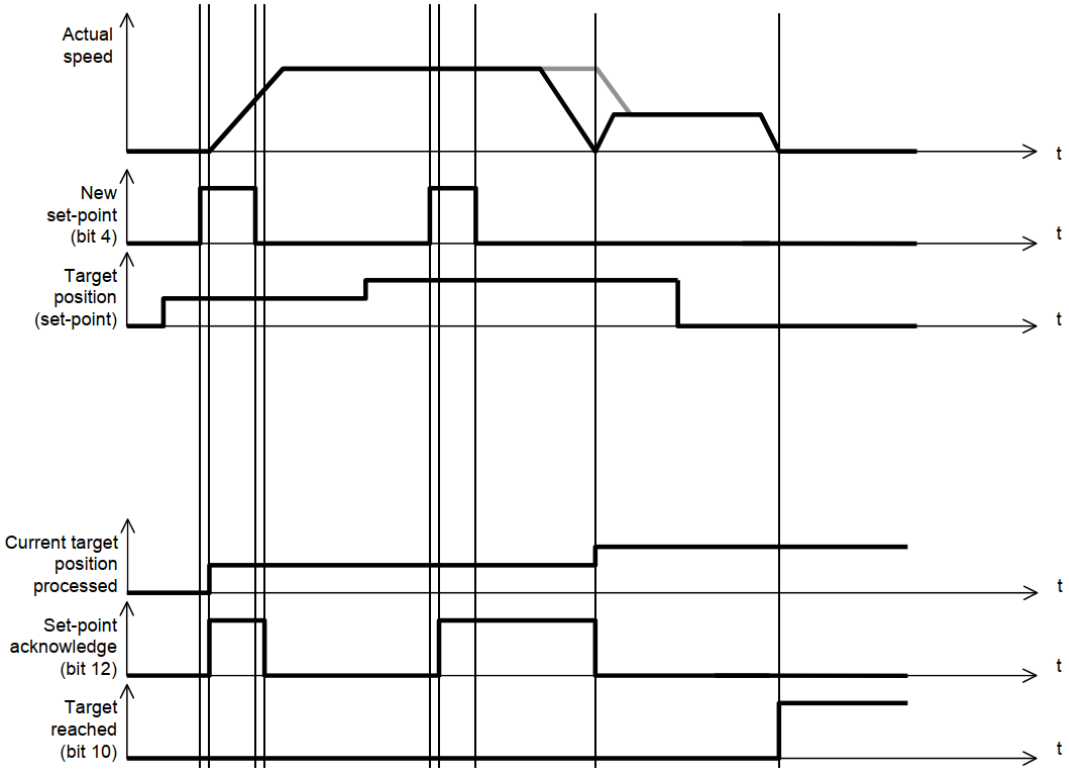


Figure 6-13

6.3.5 Impact Limited and Impact Unlimited Mode

In principle, the mode can be divided into "impact Limited" and "impact unlimited" according to impact type.

- Impact Limited

  The impact-limited positioning is performed if the object 6086_h is set to "3". The items in object $60a4_h$: 1_h-4_hrelated take effect concurrently.

- Impact Unlimited Mode

  In this mode, there is no impact limit at each position of the characteristic curve. Step to run the "impact unlimited" ramp: set the items in object 6086_h to "0".

6.3.6 Configuration Examples

- Configuration mode

  2002_h:01_h=0, operating mode 6060h=0x01, to make the device work in contour position mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 01 00 00 00 (set to profile position mode)
  Note: When switching modes, the motor should be in a deactivated state

- Parameter configuration

  Write the target position 607Ah (user units).
  Write the current segment displacement command uniform running speed 6081h (user unit/s).
  Setting the acceleration 6083h (user units/s2) and deceleration 6084h (user units/s2) of the displacement.
  Send: 601 23 7A 60 00 10 27 00 00 (set the target position 607Ah to 10000)
  Send: 601 23 81 60 00 10 27 00 00 (set target speed 6081h to 10000)
  Send: 601 23 83 60 00 40 9C 00 00 (Set acceleration 6083h to 40000)
  Send: 601 23 84 60 00 40 9C 00 00 (Set deceleration 6084_h to 40000)

- Set the control word to enable the motor

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0xF to control word 6040h in sequence; to enable it afterwards, write 0x7; to enable it, write 0xF.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- Trigger generator operation (after motor enable)

  6040_h = 0x(n)F → 0x(n+1)F, motor operation.
  The different instruction types are shown in the following table.

Table 6-11

6040_h-bit6	**6040_h-bit5**	**6040_hChange**	**Description**
0	0	0x0F → 0x1F	Absolute position, not immediate update
0	1	0x2F → 0x3F	Absolute position, updated immediately
1	0	0x4F → 0x5F	Relative position, not updated immediately
1	1	0x6F → 0x7F	Relative position, update now

Send: 601 2B 40 60 00 5F 00 00 00 (Set 6040_h to 0x5F, relative position movement, not immediate update)
Send: 601 2B 40 60 00 1F 00 00 00 (Set 6040_h to 0x1F, absolute position movement,

not updated immediately)

Send: 601 2B 40 60 00 6F 00 00 00 (set 6040_h to 0x6F, relative position movement, immediate update)

Send: 601 2B 40 60 00 3F 00 00 00 (set 6040_h to 0x3F, absolute position movement, update immediately)

The motor receives new position commands on the rising edge of bit 4 of control word 6040_h, so this bit needs to be cleared after each execution of a run.

- Monitoring parameters

  Actual position feedback: 6063_h (encoder unit), 6064_h (user unit).

6.4 Velocity Mode

6.4.1 Structural Chart

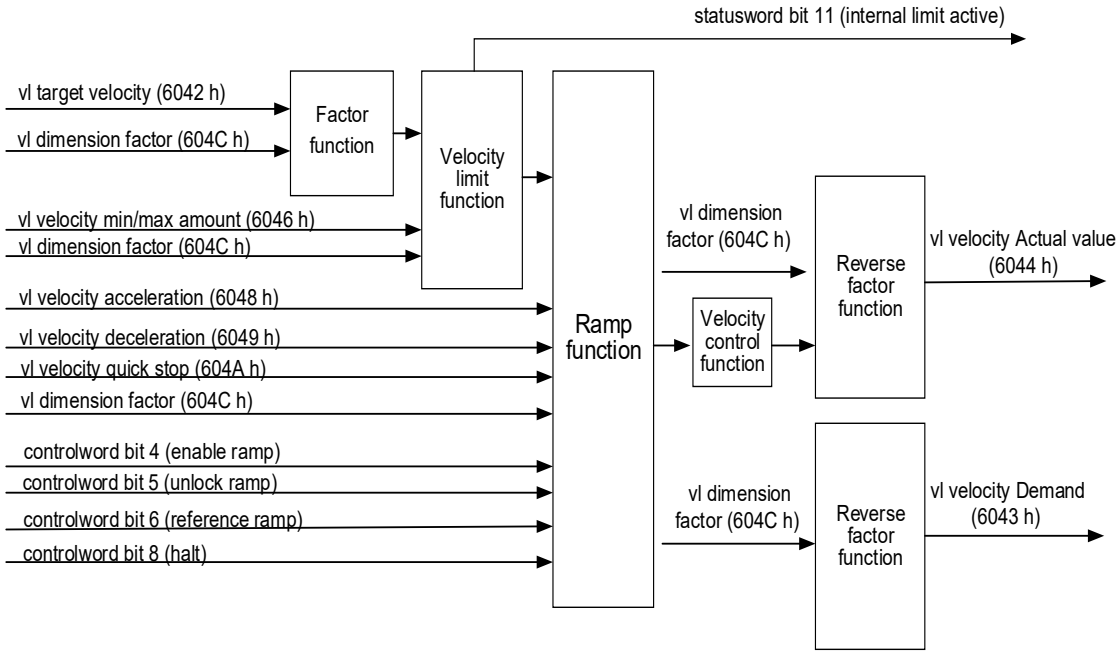


Figure 6-14

6.4.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-12

Object Index	Description
6042_h (Vl target velocity)	Target velocity in VM mode,(default unit rpm,related to $604C_h$)
6043_h (Vl velocity demand)	Target velocity as the VM mode is activate,(default unit rpm,same as 6042_h)
6044_h (Vl velocity actual value)	Current feedback velocity actual value（rpm）

Object Index	Description
604C$_h$ (VL dimension factor)	Scaling factor for speed units in VM mode(default unit: rpm) Sub-index 1 contains the numerator (multiplier) used to calculate speed, and sub index 2 contains the denominator (divisor). Velocity unit: $\text{rpm} * \frac{\text{Sub - index 1}}{\text{Sub - index 2}}$ If any of the sub indexes 1 and 2 is 0, the speed unit is rpm.
6048$_h$ (Velocity acceleration)	VI velocity acceleration =Delta speed (6048$_h$:01$_h$) / Delta time (6048$_h$:02$_h$) Acceleration in VM mode. Sub-index 1 contains the speed change （unit:rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: VI velocity acceleration =Delta speed (6048$_h$:01$_h$) / Delta time (6048$_h$:02$_h$)
6049$_h$ (Velocity deceleration)	Deceleration in VM mode. Sub-index 1 contains the speed change （unit:rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: VI velocity deceleration = Delta speed (6049$_h$:01$_h$) / Delta time (6049$_h$:02$_h$)
604A$_h$ (Quick stop deceleration)	Deceleration during Quick stop in VM mode.Sub-index 1 contains the speed change（unit:rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: VI Quick stop deceleration = Delta speed (604A$_h$:01$_h$) / Delta time (604A$_h$:02$_h$)
6046$_h$ (Velocity min max amount)	Speed limit in VM mode (default unit: rpm, related to 604C$_h$). The details are as follows: 6046$_h$: 01$_h$ set the minimum speed. If the target speed (6042$_h$) is lower than the minimum speed, its value will be set to the minimum speed of 6046$_h$: 01$_h$. 6046$_h$: 02$_h$ sets the maximum speed. If the target speed (6042$_h$) is higher than the maximum speed, its value will be set to the maximum speed of 6046$_h$: 02$_h$.

The following figure shows the effect of acceleration and deceleration on the operation process:

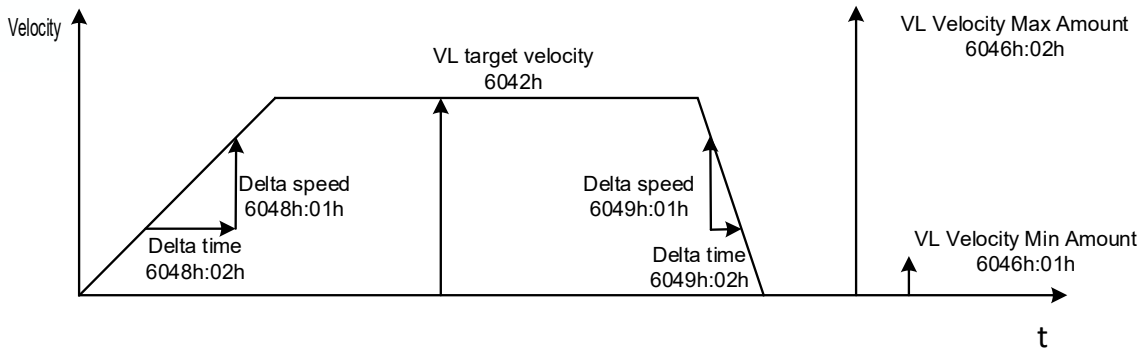


Figure 6-15

6.4.3 Control Instruction and Status Information

Enable

To enable this mode,the value must be set to "0" in object 2002_h:01_h and "2" in object 6060_h (Modes of Operation).

Control Word

15 9	8	7	6	5	4	3 0
(see 4.1.2)	Halt	(see 4.1.2)	Reference ramp	Unlock ramp	Enable ramp	(see 4.1.2)

MSB LSB

In Velocity mode, the following bits in the object 6040_h (control word) have special functions:

Table 6-13

Bit	Value	Description
2	0	When Quick stop is triggered, the motor will perform quick braking according to the Quick stop deceleration set in the object $604A_h$. The controller then switches to the "Switch on disabled" state
	1	No Operation
4	0	The output changes to the given immediately without acceleration and deceleration
	1	Adjust the velocity according to the acceleration and deceleration setting
5	0	No longer adjust velocity from the planner output, the velocity output shall be locked to the current value
	1	Adjust the speed according to the speed output of the planner
6	0	velocity curve planner input is 0
	1	velocity curve planner input is the given target velocity

Bit	Value	Description
8	0	No Operation
	1	Motor suspension

Status Word

15 14	13 12	11	10	9 0
(see 4.1.3)	reserved	(see4.1.3)	reserved	(see 4.1.3)
MSB				LSB

In Velocity mode, the bit in the object **6041_h** (control word) has no special function. For the definition of the general bit, see 4.1.3 status sub-word.

6.4.4 Function Description

In this mode, the motor operates according to the preset target velocity, similar to the converter.

6.4.5 Configuration Examples

- Configuration mode

  Write 2002_h:01_h=0, 6060_h=0x02, configured for velocity mode (VM).
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 02 00 00 00 (set to velocity mode)
  Note: When switching modes, the motor should be in a deactivated state

- Parameter configuration

  Write acceleration: 6048h:01h= 500 (unit rpm), 6048h:02h= 1 (unit 1s).
  Write acceleration: 6049h:01h= 500 (in rpm), 6049h:02h= 1 (unit 1s).
  Send: 601 23 48 60 01 F4 01 00 00 (set 6048h:01h to 500)
  Send: 601 2B 48 60 02 01 00 00 00 (set 6048h:02h to 1)
  Send: 601 23 49 60 01 F4 01 00 00 (Set 6049h:01h to 500)
  Send: 601 2B 49 60 02 01 00 00 00 (Set 6049_h:02_h to 1)

- Write control word

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0x7F to control word 6040h in sequence; to write 0x7 for subsequent interruptions; and to write 0x7F for enabling.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 7F 00 00 00 (Set 6040_h to 0x7F to enable motor)

- Set the target speed and run the motor.

  When the motor is enabled, it will run at the speed of 6042h.
  Send: 601 2B 42 60 00 2C 01 00 00 (Set target speed 6042_h to 300rpm, motor will run at 300rpm)

- Parameter monitoring

  Current actual speed 606C$_h$ (unit rpm).

6.5 Profile Velocity Mode (PV)

In this mode, the upper controller sends the Target velocity, acceleration and deceleration to the servo drive, and the servo adjusts velocity and torque internally.

6.5.1 Structure Chart

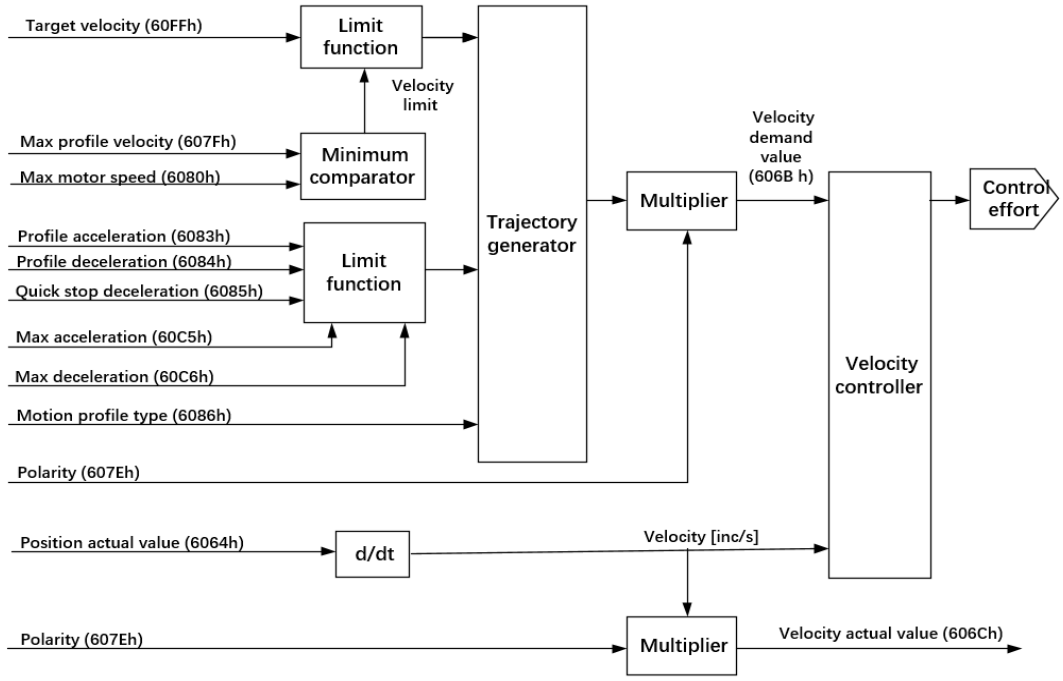


Figure 6-16

6.5.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-14

Object index	Description
6063$_h$ (Position Actual Enc Value)	Position Actual Enc Value(encoder unit)
6064$_h$（Position actual user value）	Position actual user value(user unit)
606B$_h$（Velocity demand value）	Velocity demand value
606C$_h$（Velocity actual value）	Current velocity actual value feedback(unit:rpm)
607E$_h$（Polarity）	Rotation direction(polarity),See "5.1.4 607E$_h$: Polarity" for details
607F$_h$（Max profile velocity）	Maximum profile velocity during operation (user unit / s), which is used to limit the velocity
6080$_h$（Max motor speed）	Max motor speed(rpm)

Object index	Description
6083_h（Profile acceleration）	The profile acceleration (user unit /s^2) during operation, which is limited by $60C5_h$.
6084_h（Profile deceleration）	The profile deceleration (user unit /s^2) during operation, which is limited by $60C6_h$.
6085_h（Quick stop deceleration）	Deceleration during "Quick stop" (user unit/s^2)
6086_h（Motion profile type）	Choose the type of curve, i.e. slope, If the value is "0", the impact (acceleration) will not be limited, i.e. ladder curve; If the value is "3", the value in $60A4_h$: 01_h-02_h will be used to limit the impact (acceleration), i.e. S-curve. This equipment only uses 01_h and 02_h indexes.
$60C5_h$（Max acceleration）	Max acceleration (user unit /s^2)
$60C6_h$（Max deceleration）	Max deceleration (user unit /s^2)
$60FF_h$（Target velocity）	Target velocity(user unit /s)

6.5.3 Control Instruction and Status Information

Enable

When this mode is enabled, the value must be set to "0" in object 2002_h:01_h and "3" in object 6060_h (Modes of Operation).

Control Word

15 9	8	7	6 4	3 0
(see 4.1.2)	Halt	(see 4.1.2)	reserved	(see 4.1.2)
MSB				LSB

In Profile velocity mode, the following bits in the object 6040_h (control word) have special functions

Table 6-15

Bit	Value	Meaning
8	0	Motor keep operating
	1	Suspend

Status Word

15	14	13	12	11	10	9 0
(see 4.1.3)	-		Speed	(see 4.1.3)	Target reached	(see 4.1.3)
MSB						LSB

In Position velocity mode, the following bits in the object 6041_h (status word) have special function

Table 6-16

Bit	Value	Definition
10	0	Halt(Bit8 in control word 6040_h)=0:Target position not reached Halt (Bit8 in control word 6040_h) = 1: Axis decelerates
	1	Halt (Bit8 in control word 6040_h) = 0: Target position reached Halt (Bit8 in control word 6040_h) = 1: Velocity of axis is 0
12	0	Velocity of axis is not 0
	1	Velocity of axis is 0

6.5.4 Function Description

In this mode, the master controller sends target speed, acceleration and deceleration to the servo drive, and the driver adjust the speed and torque internally.

6.5.5 Configuration Examples

- Configuration mode

  Write 2002_h:01_h=0, run mode 6060_h=0x03 to operate in profile velocity mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 03 00 00 00 (set to profile velocity mode)
  Note: When switching modes, the motor should be in a deactivated state

- Configuration parameters.

  Write contour acceleration: 6083_h = 40000 (user units/s2).
  Write contour deceleration: 6084_h= 40000 (user units/s2).
  Send: 601 23 83 60 00 40 9C 00 00 (set acceleration 6083_h to 40000)
  Send: 601 23 84 60 00 40 9C 00 00 (set deceleration 6084_h to 40000)

- Write control word.

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0xF to control word 6040h in sequence; to write 0x7 for subsequent interruptions; to write 0xF for enabling.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- Set the target speed and run the motor.

  When the motor is enabled, it will run at the speed of $60FF_h$.
  Send: 601 23 FF 60 00 50 C3 00 00 (Set target speed $60FF_h$ to 50000, motor will run at 300rpm)
  Send: 601 23 FF 60 00 B0 3C FF FF (Set target speed $60FF_h$ to -50000, motor will run at -300rpm)

- Parameter monitoring: Current actual speed $606C_h$(unit rpm).

6.6 Profile Torque Mode(PT)

In this mode, the master controller sends the target torque 6071_h and the torque ramp constant 6087_h to the driver, then the driver regulates torque internally. When the velocity reaches the limit , it will switches to the velocity regulation stage.

6.6.1 Structural Chart

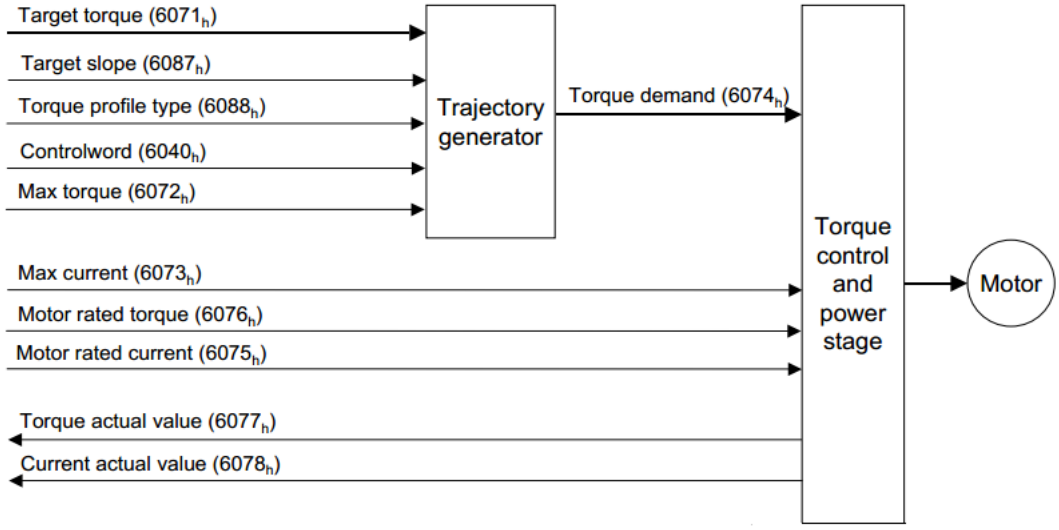


Figure 6-17

6.6.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-17

Object Index	Description
$605D_h$ (Halt option code)	Halt option
$605A_h$ (Quick stop option code)	Quick stop option
6071_h (Target torque)	Target torque(unit: 0.1%)
6072_h (Max torque)	Maximum torque on the entire slope (acceleration, holding torque, braking)(unit: 0.1%)
2007_h:10_h(TorqCtrlFwrSpeedLimit)	Forward velocity limit during torque control (rpm)
2007_h:11_h(TorqCtrlBwrSpeedLimit)	Backward velocity limit during torque control (rpm)
6087_h (Torque slope)	Maximum torque change per second(unit: 0.1%)
6088_h (Torque profile type)	Torque ramp type (0-ramp, 2-none)
6077_h (Torque actual value)	Torque actual value(unit: 0.001N • m)

6.6.3 Control Instruction and Status Information

Enable

When this mode is enabled, the value in object 2002_h:01_h and object in 6060_h (Modes of Operation)must be set to "0" and "4" respectively.

Control Word

15 9	8	7	6 4	3 0
(see 4.1.2)	Halt	(see 4.1.2)	reserved	(see 4.1.2)

MSB LSB

In the Profile torque mode, the following bits in object 6040_h (control word) have special functions.

Table 6-18

Bit	Value	Description
8	0	Motor keep operating
	1	Suspend

Status Word

15 14	13 12	11	10	9 0
(see 4.1.3)	reserved	(see 4.1.3)	Target reached	(see 4.1.3)

MSB
LSB

In Position torque mode, the following bits in the object 6041_h (status word) have special function

Table 6-19

Bit	Value	Definition
10	0	Halt(Bit8 in control word 6040_h)=0:Target position not reached Halt (Bit8 in control word 6040_h) = 1: Axis decelerates
	1	Halt (Bit8 in control word 6040_h) = 0: Target position reached Halt (Bit8 in control word 6040_h) = 1: Velocity of axis is 0

6.6.4 Function Description

Velocity Limit Function

The Integrated low-voltage servo motor operating in Position torque mode, when the motor velocity exceeds the velocity limit (2007_h:10_h and 2007_h:11_h), it will automatically switch to the Velocity mode and the current velocity will be controlled within the velocity limit.When the servo detects that the torque commend is less than the torque average at current velocity, the servo motor will exit Velocity mode and return to Position torque mode.

6.6.5 Configuration Examples

- Configuration mode

  Write 2002h:01h=0, operating mode 6060_h=0x04 to operate in profile torque mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 04 00 00 00 (set to profile torque mode)
  Note: When switching modes, the motor should be in a deactivated state

- Configuration parameters

  Write torque ramp type 6088_h = 0 (0 - ramp, 2 - no ramp).
  Write torque ramp 6087_h = 10 (unit 0.1%).
  Send: 601 2B 88 60 00 00 00 00 00 (set ramp type 6088h to ramp)
  Send: 601 23 87 60 00 10 00 00 00 (set torque ramp 6087_h to 10)

- Write control word

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0xF to the control word 6040_h in sequence; to write 0x7 for subsequent interruptions; and to write 0xF for enabling.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- Set the target speed and run the motor.

  When the motor is enabled, it will run at the torque of 6071_h.
  Send: 601 2B 71 60 00 F4 01 00 00 (Set target torque of 6071_h to 50000, motor will run at 50% of rated torque)

- Parameter monitoring

  Actual torque 6077_h (unit 0.001N • m).

6.7 Homing Mode

The Homing mode is used to operate from the current position to the home position of the device. In the operation ,the maximum acceleration, minimum deceleration, maximum velocity and minimum velocity are all taken into account.

6.7.1 Structure Chart

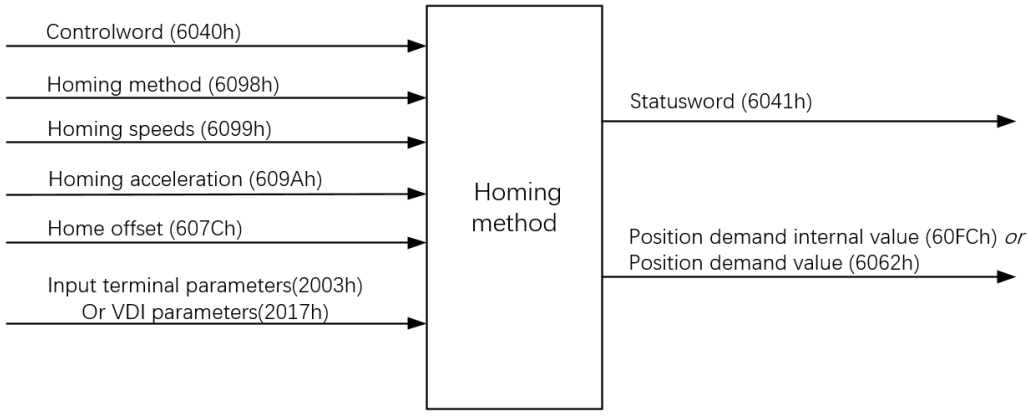


Figure 6-18

6.7.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-20

Object Index	Description
6062h（Position demand value）	Target position instruction value in effect currently(user unit)
$607C_h$ (Home Offset)	Set the physical deviation between the motor origin and the machine reference point; When the Homing operation is completed, Bit15= 1 in the status word 6041_h, it takes effect
6098_h (Homing Method)	Homing Method(See"6.7.4 Function description")
6099_h (Speed)	Velocity(user unit/s) Sub-index：6099_h:01_h ： the velocity to find the switch 6099h:02h ： the velocity to find the origin
$609A_h$ (Homing Acceleration)	Acceleration and deceleration in homing process(user unit/s^2)
$60FC_h$（Position demand internal value）	Output of track generator, i.e.real-time position instruction internal planned(encoder unit)
6063h (Position actual enc value)	Position actual enc value(encoder unit)
2003_h (Input terminal parameters)	Function configuration of entity input terminals (see "7.1.1 Digital input") Note: multiple input terminals cannot be configured as the same switch at the same time, otherwise terminal setting failure will be reported
2017_h (VDI/VDO parameters)	The parameters of virtual terminal input function and logic function

Example of Velocity Meaning

The following figure shows the velocity change during the homing method 18:

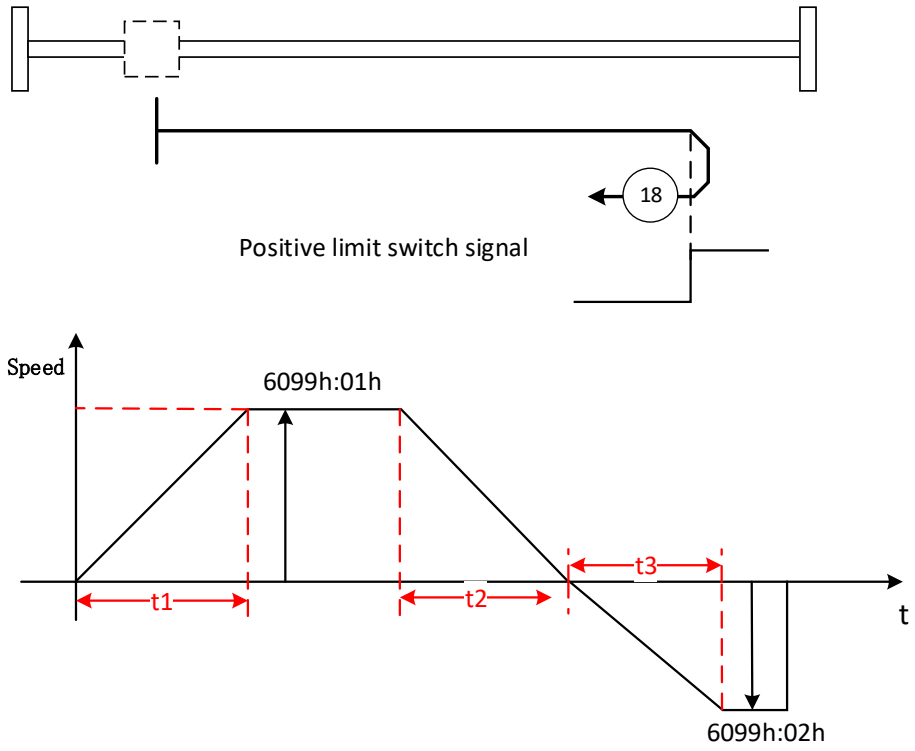


Figure 6-19

$t1=\frac{6099_h:01_h}{609A_h}$; $t2=\frac{6099_h:01_h}{609A_h}$; $t3=\frac{6099_h:02_h}{609A_h}$

6.7.3 Control Instruction and Status Information

Enable

To enable this mode, the value in object $2002_h:01_h$ must be set to "0",the value in object 6060_h (Modes of Operation) must be set to "6", and the multi-function terminals are configured to origin switch or limit switch function. See Section7.1 for multi-function terminals configuration.

Control Word

15 9	8	7	6 5	4	3 0
(see 4.1.2)	Halt	(see 4.1.2)	reserved (0)	Homing operation start	(see 4.1.2)
MSB					LSB

In Homing mode, the following bits in the object **6040_h** (control word) have special functions

Table 6-21

Bit	Value	Definition
4	0	Homing mode is not enable
	1	Enable or continue Homing mode
8	0	Whether to enable the homing or not according to bit4
	1	Suspend according to the setting of $605D_h$

Status Word

15 14	13	12	11	10	9 0
(see 4.1.3)	Homing error	Homing attained	(see 4.1.3)	Target reached	(see 4.1.3)
MSB					LSB

In Homing mode, the following bits in the object 6041_h (Status-word) have special function:

Table 6-22

Bit	Value	Definition
10	0	Target position not reached
	1	Target position reached
12	0	Homing not completed
	1	Homing completed

6.7.4 Function Description

6.7.4.1 Switch to find the origin

The way of using the switch to return to the origin, currently supports 17 ~ 30 in CiA402. For the return to zero, we often need to start working from zero. Generally speaking, we need to run to the zero position command after returning to zero. For the convenience of users, this series of integrated motors will automatically return to the zero position after homing.

Function number configuration of input terminal DI which related to HM mode:

Table 6-23

Function number	Function definition
14	Positive limit switch
15	Negative limit switch
31	Origin switch

The Homing method supports 17-30 in CiA402,Generally speaking,it is necessary to start from home position after the homing operation.For the convenience to users, PMM series integrated servo automatically returns to the home position after the homing operation.

The direction and switch status requiring homing are as follows:

Table 6-24

Homing Method	Switch Related
17	Negative limit switch

Homing Method	Switch Related
18	Positive limit switch
19	Origin switch
20	Origin switch
21	Origin switch
22	Origin switch
23	Origin switch,Positive limit switch
24	Origin switch,Positive limit switch
25	Origin switch,Positive limit switch
26	Origin switch,Positive limit switch
27	Origin switch,Negative limit switch
28	Origin switch,Negative limit switch
29	Origin switch,Negative limit switch
30	Origin switch,Negative limit switch

- Homing Method 17:

  Mechanical origin: negative limit switch

  When the negative limit switch is invalid, servo starts reversing to home position at high speed,once the negative limit switch is valid (edge-jump), slows down, reverses, moves forward at low speed, and halts until the negative limit switch is invalid (edge-jump).

  When the negative limit switch is valid, servo starts reversing to home position at low speed directly, and halts until the negative limit switch is invalid (edge-jump).

  The movement track is as follows:

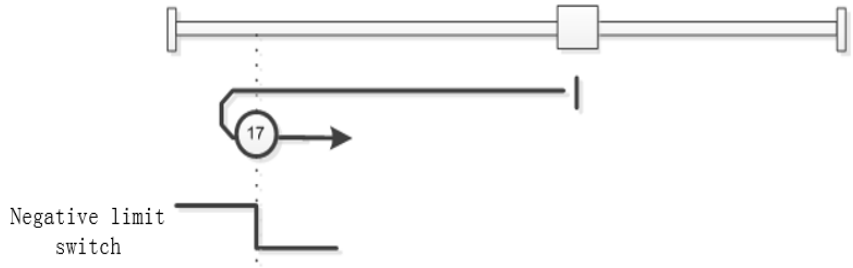


Figure 6-20

- Homing Method 18:

  Mechanical origin: positive limit switch

  When the positive limit switch is invalid, servo starts moving to home position at high speed. once the positive limit switch is valid (edge-jump), slows down, reverses, moves backward at low speed , and halts until the positive limit switch is invalid (edge-jump).

  When the positive limit switch is valid, servo starts reversing to home position at low speed directly , and halts until the positive limit switch is invalid (edge-jump).

  The movement track is as follows:

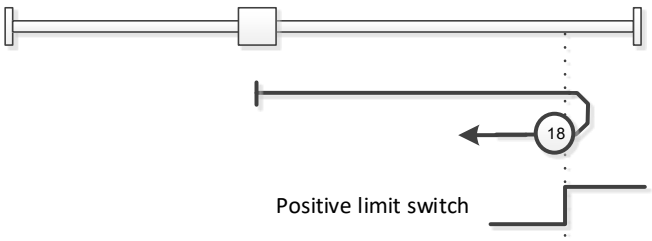


Figure 6-21

- Homing Method 19 and 20:

  Mechanical origin: origin switch

  pattern 19: When the origin switch is invalid, servo starts moving to home position at high speed. once the origin switch is valid (edge-jump), slows down, reverses, moves backward at low speed ,and halts until the origin switch is invalid (edge-jump);When the origin switch is valid, servo starts reversing to home position at low speed directly,and halts until the origin switch is invalid (edge-jump).

  pattern 20: When the origin switch is invalid, servo starts moving to home position at low speed. and halts until the origin switch is valid (edge-jump).When the origin switch is valid, servo starts reversing to home position at low speed directly,once the origin switch is invalid (edge-jump), slows down,reverses,moves backward at low speed,and halts until the origin switch is valid (edge-jump).

  The movement track is as follows:

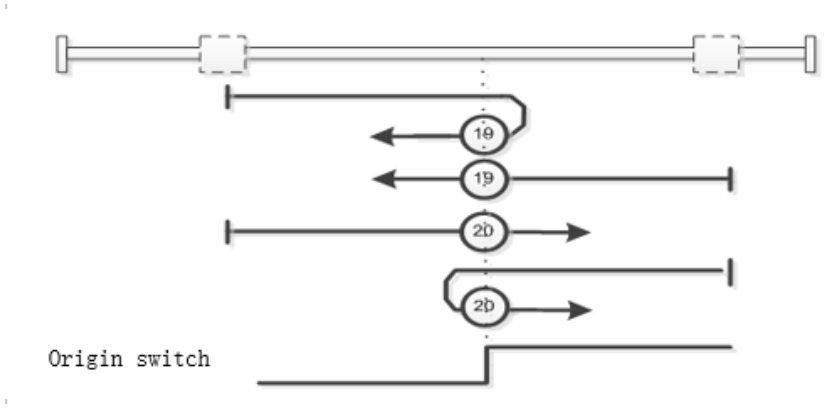


Figure 6-22

- Homing Method 21 and 22:

  Mechanical origin: origin switch

  pattern 21: When the origin switch is invalid, servo starts reversing to home position at high speed. once the origin switch is valid (edge-jump), slows down, reverses, moves backward at low speed ,and halts until the origin switch is invalid (edge-jump); When the origin switch is valid, servo starts moving to home position at low speed directly , and halts until the origin switch is valid (edge-jump).

  pattern 22: When the origin switch is invalid, servo starts moving to home position at low

speed. and halts until the origin switch is valid (edge-jump).When the origin switch is valid, servo starts reversing to home position at low speed directly,once the origin switch is invalid (edge-jump), slows down, reverses, moves backward at low speed ,and halts until the origin switch is valid (edge-jump).

The movement track is as follows:

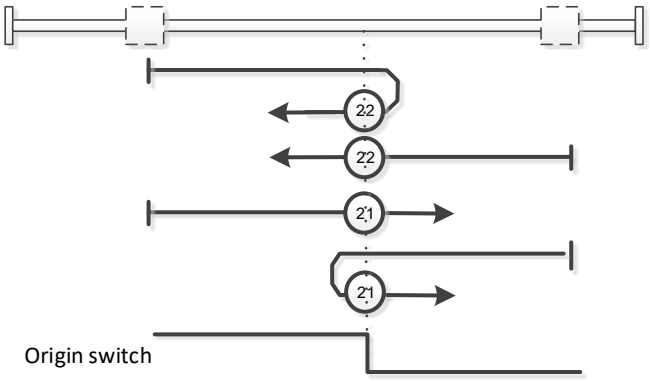


Figure 6-23

- Homing Method 23 ~ 26:

  Mechanical origin: origin switch

  Actually,in these methods the motor moves in one direction to scan the origin switch.the relatively shorter track is find only when the origin switch is valid.

  Homing Method 23:

  A. when the origin switch is invalid, the servo start reversing to home position at the high speed in the positive direction when the origin switch is not encountered , when the origin switch is valid (edge-jump), moves backward at low speed ,and halts until the origin switch is invalid (edge-jump);
  B. when the origin switch is invalid, the servo starts moving to home position at high speed, when the positive switch is valid (edge-jump) reverses,moves at high speed, when the origin switch is valid (edge-jump) slows down , continue to moves in the reverse direction at low speed, and halts until the origin switch is invalid (edge-jump);
  C. when the origin switch is valid, the servo start reversing to home position at the low speed,and halts until the origin switch is invalid (edge-jump).

  Homing Method 24:

  A. when the origin switch is invalid, when the origin switch is not encountered,the motor start reversing to home position at the high speed, when the origin switch is valid (edge-jump), slows down, reverses, moves backward at low speed,when the origin switch is invalid (edge-jump), reverses, moves forward at low speed , and halts until the origin switch is valid (edge-jump);
  B. When the origin switch is invalid, the motor start moving to home position at the high speed, when the positive switch is encountered, reverses,moves backward at high

speed.When the origin switch is valid (edge-jump),slows down,, moves backward at low speed ,When the origin switch is invalid(edge-jump),reverses,moves forward at low speed,and halts until the origin switch is valid (edge-jump).

C. when the origin switch is invalid, the motor start reversing to home position at the low speed directly, when the origin switch is invalid (edge-jump),reverses, moves forward at low speed,and halts until the origin switch is valid (edge-jump).

Homing Method 25:

A. when the origin switch is invalid, the motor start moving to home position at the high speed in the positive direction when the origin switch is not encountered , when the origin switch is valid (edge-jump),slows down,moves forward at low speed,when the origin switch is invalid (edge-jump),reverses , moves backword at low speed,and halts until the origin switch is valid (edge-jump);
B. when the origin switch is invalid, the motor starts moving to home position at high speed, when the positive switch is valid (edge-jump) reverses , moves backward at high speed, when the origin switch is valid (edge-jump) slows down , moves forward at low speed, when the origin switch is invalid (edge-jump)reverses , moves backward at low speed, and halts until the origin switch is valid (edge-jump);
C. when the origin switch is invalid, the motor start moving to home position at the low speed directly, when the origin switch is invalid (edge-jump), , reverses, moves backward at low speed,and halts until the origin switch is valid (edge-jump).

Homing Method 26:

A. when the origin switch is invalid, the motor start moving to home position at the high speed in the positive direction if the positive limit switch is not encountered, when the origin switch is valid (edge-jump) ,slow down, run at a positive low speed , and halts when the origin switch is invalid (edge-jump);
B. when the origin switch is invalid, the motor start moving to home position at the high speed in the positive direction ,when the positive limit switch is encountered reverses , moves backward at high speed,when the origin switch is valid (edge-jump) slows down, moves at a positive low speed, halts when the origin switch is invalid (edge-jump);
C. when the origin switch is valid, starts moving to home position at low speed, and halts when the origin switch is invalid (edge-jump).

The motion track is as follows:

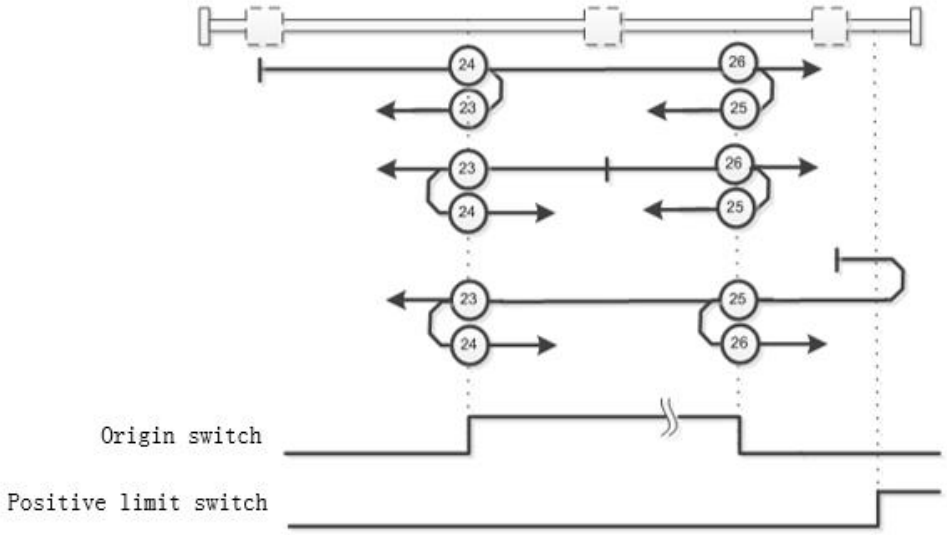


Figure 6-24

- Homing Method 27~30

  Mechanical origin: origin switch

  Actually,in these methods the motor moves in one direction to scan the origin switch.the relatively shorter track is find only when the origin switch is valid.

  Homing Method 27:

  A. If the origin switch is invalid, when the negative limit switch is not encountered, start to return to home position at reverse high speed, when the origin switch is valid (edge-jump), slow down, reverse, forward at low speed, and halts when the origin switch is invalid (edge-jump);
  B. If the origin switch is invalid, the motor start moving to home position at the high speed in the negative direction, When the negative limit switch is encountered, moves backward at high speed. When the origin switch is valid (edge-jump), slow down ,slows down , moves forward at low speed. and halts when the origin switch is invalid (edge-jump);
  C. when the origin switch is valid, the motor start moving to home position at the low speed directly, and halts until the origin switch is invalid (edge-jump).

  Homing Method 28:

  A. when the origin switch is invalid and the negative limit switch is not encountered, the motor start moving to home position at the high speed in the positive direction , when the origin switch is valid (edge-jump), slow down, reverse, moves forward at low speed, when the origin switch is invalid (edge-jump), reverse, moves backward at low speed,and halts until the origin switch is valid (edge-jump);
  B. when the origin switch is invalid, the motor starts moving to home position at high speed, when the positive limit is valid, reverse,moves forward at high speed, when the origin switch is valid (edge-jump)slow down , moves forward at low speed, when

the origin switch is invalid (edge-jump),reverses , moves backward at low speed, and halts until the origin switch is valid (edge-jump);

C. when the origin switch is valid, the motor start moving to home position at the low speed directly, when the origin switch is invalid (edge-jump), reverses, moves backward at low speed,and halts until the origin switch is valid (edge-jump).

Homing Method 29:

A. when the origin switch is invalid, the motor start moving to home position at the high speed in the positive direction when the negative switch is not encountered , when the origin switch is valid (edge-jump), slows down, moves backward at low speed ,when the origin switch is invalid (edge-jump),reverses , moves backword at low speed,and halts until the origin switch is valid (edge-jump);
B. when the origin switch is invalid, the motor starts moving to home position at high speed, when the positive limit is valid, reverse,moves forward at high speed, when the origin switch is valid (edge-jump)slow down , moves backward at low speed, when the origin switch is invalid (edge-jump),reverses , moves forward at low speed, and halts until the origin switch is valid (edge-jump);
C. when the origin switch is valid, the motor start moving to home position at the low speed directly, when the origin switch is invalid (edge-jump), reverses, moves forward at low speed,and halts when the origin switch is valid (edge-jump).

Homing Method 30:

A. when the origin switch is invalid, the motor start moving to home position at the high speed in the negative direction when the origin switch is not encountered , when the origin switch is valid (edge-jump), slows down, moves backward at low speed ,and halts until the origin switch is valid (edge-jump);
B. when the origin switch is invalid, the motor starts moving to home position at high speed, when the negative limit is valid, reverse,moves forward at high speed, when the origin switch is valid (edge-jump)slow down , moves backward at low speed, and halts when the origin switch is invalid (edge-jump);
C. when the origin switch is valid, the motor start moving to home position at the low speed in the negative direction directly,and halts when the origin switch is invalid (edge-jump).

The motion track is as follows:

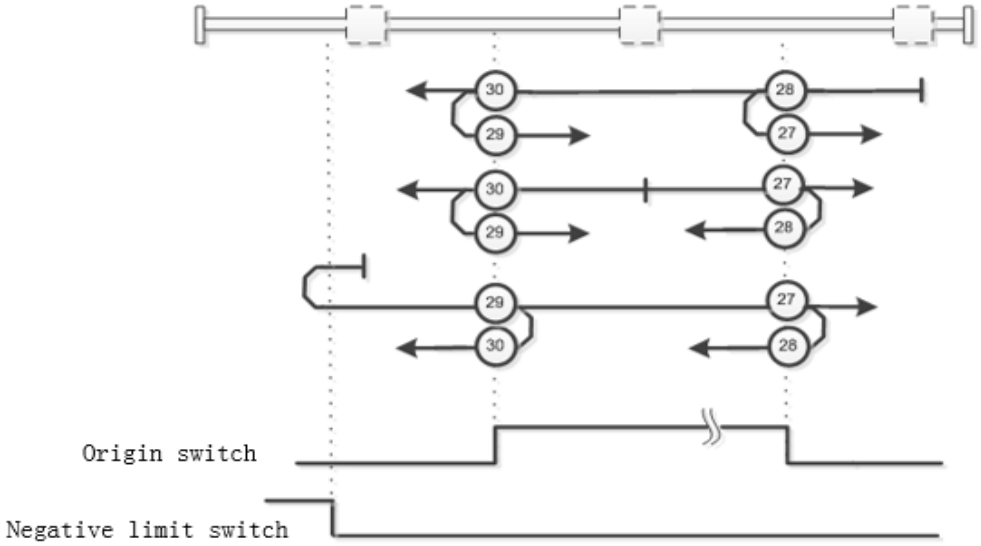


Figure 6-25

6.7.4.2 Finding the origin by blocking rotation

The return-to-origin method currently only supports the closed-loop control mode. The method of returning to the origin of locked rotor is similar to the method of using the switch to find the origin. The difference is that the torque detection value of the locked rotor replaces the limit switch.

The method and running direction of the origin return are shown in the following table:

Table 6-25

Origin return mode 6098h	Running direction
37	Press 6099_h: 02_h to run forward
38	Reverse running at 6099_h: 02_h speed

In addition, the method needs to set the following two parameters:

1. The detected torque when the locked rotor is looking for the origin (2007_h: 13_h)

   The stall torque threshold should be greater than the actual running torque in the process of finding the origin, and should generally be less than the maximum torque to avoid triggering the stall protection.

2. Detection time when finding the origin by blocking rotation (2007_h: 15_h)

   This time is the time during which the motor is locked to find the origin.

   The specific schematic diagram of the process of the locked-rotation origin return method is as follows:

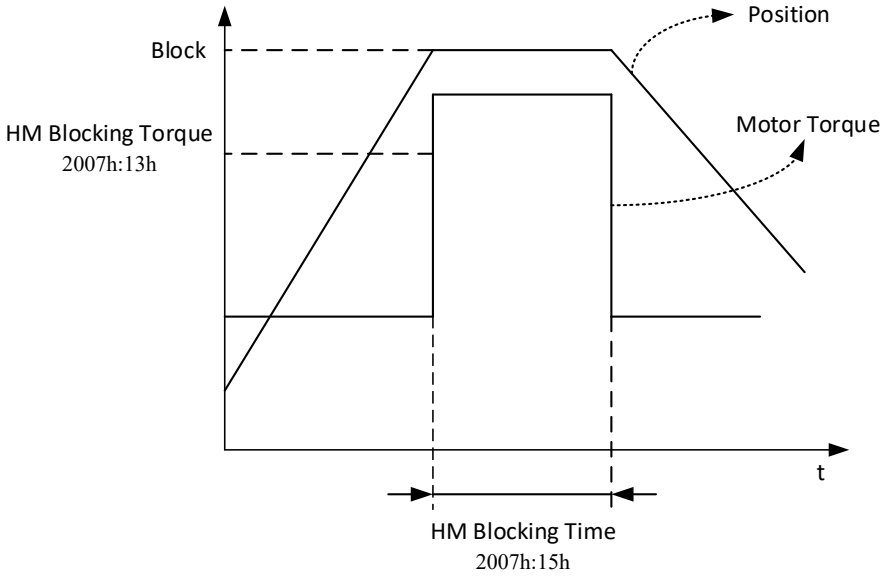


Figure 6-26

6.7.5 Configuration examples

- Setting mode

  Write 2002_h:01_h=0, run mode 6060_h=0x06 to operate in homing mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 06 00 00 00 (set to homing mode)
  Note: When switching modes, the motor should be in the deactivated state

- Setting terminal

  2003_h:03_h=15 (negative limit switch), 2003h:04h=0 (active low).
  Send: 601 2B 03 20 03 0F 00 00 00 (Set DI1 function as negative limit switch)
  Send: 601 2B 03 20 04 00 00 00 00 (set DI1 logic to active low)

- Write go home method

  6098_h =17；
  Send: 601 2F 98 60 00 11 00 00 00 (Set homing mode to 17)

- Set find limit switch speed and find home signal speed

  6099h:01h= 10000 (user unit/s).
  6099h:02h= 1000 (user units/s).
  Send: 601 23 99 60 01 10 27 00 00 (Set the speed of finding limit switch to 10000)
  Send: 601 23 99 60 02 10 27 00 00 (Set the speed of finding the home signal to 10000)

- Set go home acceleration

  609Ah = 200000 (user units/s2).
  Send: 601 23 9A 60 00 40 0D 03 00 (Set go home acceleration to 200000)

- Write control word

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0xF to the control word 6040h in sequence; to write 0x7 for a subsequent break; to write 0xF for an enable.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- Trigger generator operation (after motor enable)

  6040h = 0x0F → 0x1F, motor operating
  Send: 601 2B 40 60 00 1F 00 00 00 (Set 6040_h to 0x1F, motor operating)

6.8 Interpolated Position Mode

Interpolated position mode is used to synchronize multiple axes.To do this, the advanced controller calculates slope and path, transmits corresponding axis position demanded at a specific to the controller .The controller interpolates between these intermediate positions.

6.8.1 Structure Chart

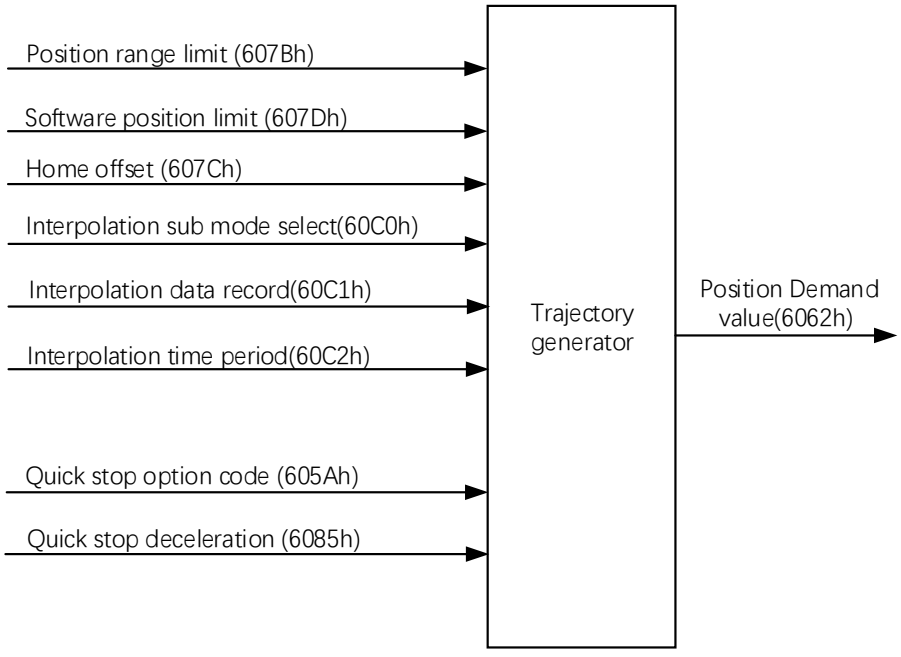


Figure 6-27

6.8.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-26

Object index	Description
607Bh (Position Range Limit)	Position range limit
607Dh (Software Position Limit)	Software position limit(user unit)
$60C0_h$(Interpolation sub mode select)	0 Choose the interpolation type. This device only supports linear interpolation. The default value is 0
$60C1_h$(Interpolation data record)	Set interpolation target location(user unit)
$60C2_h$ (Interpolation time period)	Interpolation period(recommended 1-20ms), defined by sub indexes 01_h and 02_h, interpolation period = t * 10^n seconds sub indexes: $60C2_h$:01_h(Interpolation time period value): Interpolation time constant t. $60C2_h$:02_h(Interpolation time index): Interpolation time constant n
$605A_h$ (Quick stop option code)	Quick stop option
$605D_h$ (Halt option code)	Halt option
6085_h (Quick stop deceleration)	Deceleration during "Quick stop" (user unit/s^2)
6062_h (Position Demand Value)	Current position demand value inside the drive (user unit)
6063_h (Position actual enc Value)	Current position actual value feedback (encoder unit)

6.8.3 Control Instruction and Status Information

Enable

As this mode is enabled, the value in object 2002_h:01_h and object 6060_h (Modes of Operation)must be set to "0" and "7" respectively.

Control Word

15 9	8 7	6 5	4	3 0
(see 4.1.2)	Halt	(see 4.1.2)	reserved	Enable interpolation	(see 4.1.2)
MSB				LSB

In Interpolated position mode, the following bits in object 6040_h (control word) have special functions.

Table 6-27

Bit	Value	Definition
4	0	Disable interpolation
	1	Enable interpolation
8	0	Execute instruction of bit4
	1	Halt according to the configuration of $605D_h$

Status Word

15 14	13	12	11	10	9 0
(see 4.1.3)	following error	ip mode active	(see 4.1.3)	Target reached	(see 4.1.3)
MSB					LSB

In Interpolated mode, the following bits in the object 6041_h (status word) have special function

Table 6-28

Bit	Value	Definition
10	0	The target position not reached
	1	The target position not reached
12	0	Interpolation not enable
	1	Interpolation enable
13	0	No following error
	1	Following error

6.8.4 Function Description

Interpolation Diagram

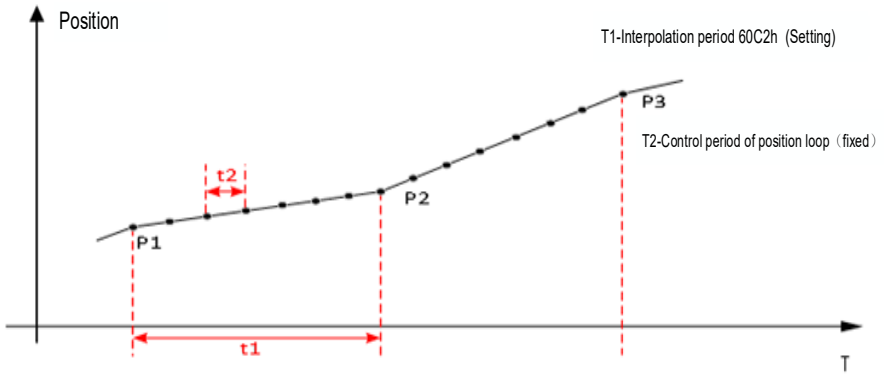


Figure 6-28

Notes:

t1: Interpolation period, which can be set in object dictionary $60C2_h$.

t2: Control period of position loop , determined by the servo drive internally.

P1 / P2 / P3: Absolute position. The absolute position instructions is sent by the object dictionary $60C1_h$:01_h. The interpolation mode only supports the absolute position instructions.

6.8.5 Configuration Examples

- Setting mode

  Write 2002_h:01_h=0, run mode 6060h=0x07 to make it work in interpolation mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 07 00 00 00 (set to interpolation mode)
  Note: When switching modes, the motor should be in the deactivated state

- Set interpolation time

  Set the time constant 60C2h: 01h = 20 and the time index $60C2_h$: 02_h = -3, the time is 20*10-3 (unit s);
  Send: 601 2F C2 60 01 14 00 00 00 (Set the time constant $60C2_h$:01h to 20)
  Send: 601 2F C2 60 02 FD 00 00 00 (Set the time index $60C2_h$:02_h to -3)

- Write control word

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0x1F to the control word 6040h in sequence; to write 0x7 for subsequent disconnection; to write 0x1F for enabling.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 1F 00 00 00 (Set 6040_h to 0x1F to enable motor)

- Trigger generator operation (after motor enable)

  The host computer writes the interpolation position $60C1_h$:01_h according to the interpolation cycle (only absolute position commands are supported, user units)

6.9 Cyclic Sync Position Mode (CSP)

In this mode, Preset absolute positions are sent to the controller through the field bus at fixed intervals (hereinafter call as "cycles").In this mode, the controller no longer calculates the slope,only follows the preset value. The target position is transmitted by PDO, and the controller will react immediately. Bit4 in control word does not need to be set (different from Profile position mode). The preset target value is absolute and therefore independent to the transmitting times per cycle.

6.9.1 Structure Chart

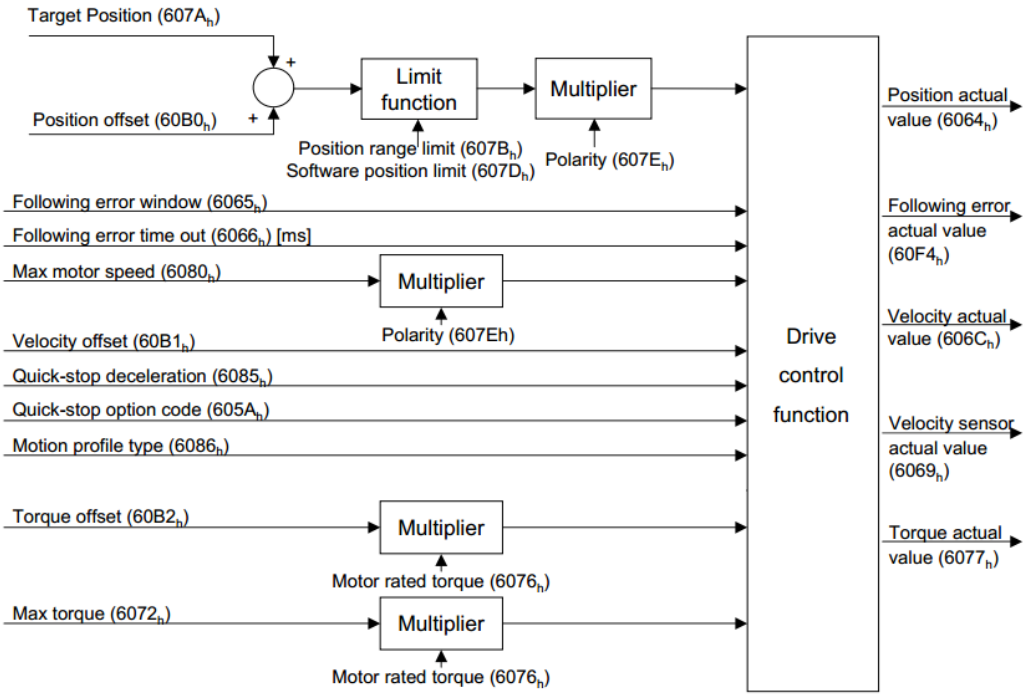


Figure 6-29

6.9.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-29

Object Index	Description
607A$_h$ (Target position)	Target position(user unit)
607B$_h$ (Position Range Limit)	Position Range Limit(user unit)
607D$_h$ (Software Position Limit)	Target position limit (user unit)
607E$_h$ (Polarity)	Rotation direction(polarity),See "5.1.4 607E$_h$: Polarity" for details
6065$_h$ (Following Error Window)	this object specifies the allowed tolerance from the rated position. If the actual position exceeds this tolerance for longer than the specified time (6066$_h$), tracking error will be reported, (user unit)
6066$_h$ (Following Error Time Out):	Time range of following error , unit: millisecond. If the actual position exceeds the tolerance (6065$_h$) for longer than this time range, tracking error will be reported, Bit13 of the control word will be set to 1.
6080$_h$ (Max motor speed)	Max motor speed(rpm)
6085$_h$ (Quick-Stop Deceleration)	Deceleration during "Quick stop" (user unit/s^2)
605A$_h$ (Quick-Stop Option Code)	Quick-Stop Option
605D$_h$ (Halt Option Code)	Halt Option
6086$_h$ (Motion Profile Type)	The slope type to be operated,if the value is "0", the impact will not be limited. If the value is "3", the impact will

Object Index	Description
	be limited with the value of $60a4_h$: 01_h - 04_h. Only 01_h-02_h is used by This equipment.
$60B0_h$ (Position offset)	Position offset(user unit)
$60B1_h$ (Velocity offset)	Velocity offset(user unit)
6064_h (Position Actual Value)	Current absolute position feedback from motor(user unit)
$606C_h$ (Velocity Actual Value)	Current velocity actual value feedback(unit:rpm)
$60F4_h$ (Following Error Actual Value)	Following Error Actual Value(user unit)

6.9.3 Control Instruction and Status Information

Enable

As this mode is enabled, the value in object 2002_h:01_h and object 6060_h (Modes of Operation)must be set to "0" and "8" respectively.

Control Word

In the Cyclic sync position mode, the following bits in object 6040_h (control word) have special functions. For the definition of the general bits, please refer to section 4.1.2 control words.

Note:The CSP mode only supports the absolute position instructions.

Status Word

15 14	13	12	11	10	9 0
(see 4.1.2)	Following error	Target position ignored	(see 4.1.2)	reserved	(see 4.1.2)
MSB					LSB

the Cyclic sync position mode, the following bits in object **6041_h** (status word) have special functions:

Table 6-30

Bit	Value	Description
12	0	The controller doesn't follow the target preset value,**$607A_h$** (Target position) will be ignored.
	1	The controller follows the preset target value, and the object **$607A_h$** (target position) is used as the position control input.
13	0	No following error
	1	Following error

6.9.4 Function Description

In this mode, absolute preset position values will be sent to the controller by the field bus at fixed time intervals (hereinafter call as "cycles")

6.9.5 Configuration Example

- Configuration mode
- Write 2002_h:01_h=0, run mode 6060_h=0x08 to make it work in cyclic synchronous position mode.

  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 08 00 00 00 (set to cyclic synchronous position mode)

- Set Control word

  6040_h=0x06→0x07→0x0F,motor operates;
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- the master controller sends the target position $607A_h$ periodically and synchronously (only absolute position command is supported).

6.10 Cyclic Sync Velocity Mode(CSV)

In the Cyclic sync velocity mode, the master controller sends the calculated target speed $60FF_h$ to the servo drive periodically and synchronously,the speed and torque regulation are performed by the servo internally.

6.10.1 Structural Chart

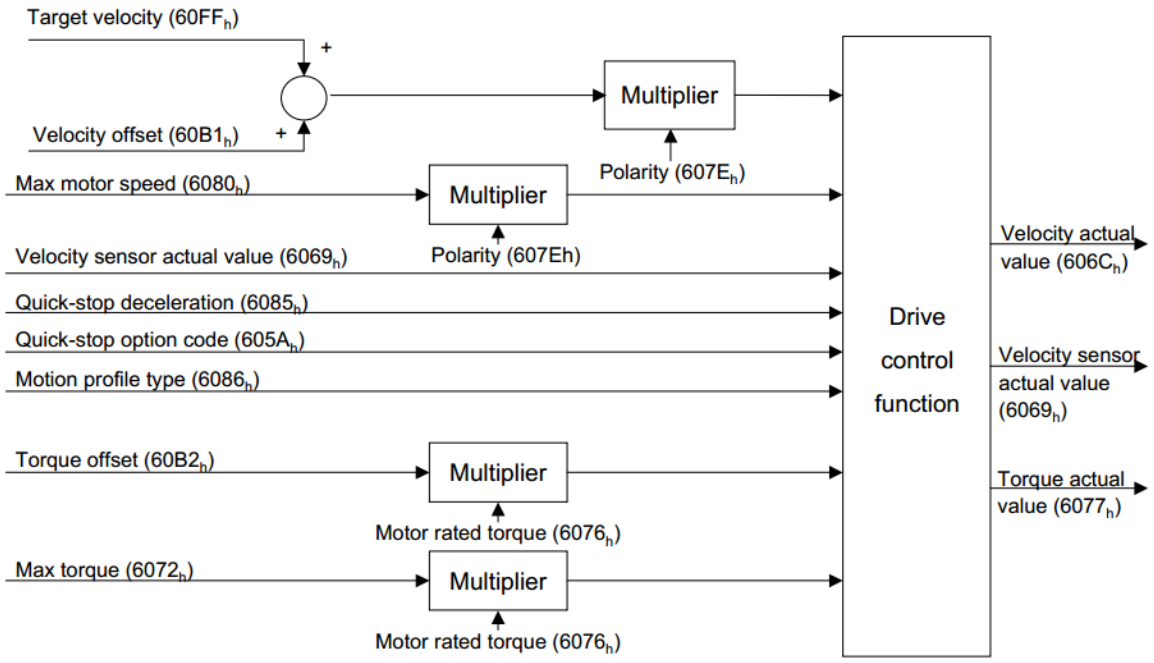


Figure 6-30

6.10.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-31

Object Index	Description
$605A_h$ (Quick-Stop Option Code)	Quick stop option
$605D_h$ (Halt Option Code)	Halt option
$606C_h$Velocity actual value	Current velocity actual value feedback (unit:rpm)
$607E_h$Polarity	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
$60FF_h$Target velocity	Target velocity (user unit:/s)
6080_hMax motor speed	Max motor speed
6085_hQuick-stop deceleration	Deceleration during "quick stop" (user unit/s^2)
6086_hMotion profile type	Choose the type of curve, i.e. slope, If the value is "0", the impact (acceleration) will not be limited, i.e. ladder curve; If the value is "3", the value in $60A4_h$: 01_h-02_h will be used to limit the impact (acceleration), i.e. S-curve. This equipment only uses 01_h and 02_h indexes.
$60B1_h$Velocity offset	Velocity offset (user unit)
$60B2_h$Torque offset	Torque offset (0.1%)
6072_hMax torque	Maximum torque on the entire slope (acceleration, holding torque, braking)(unit: 0.1%)

6.10.3 Control Instruction and Status Information

Enable

When this mode is enabled, the value must be set to "0" in object 2002_h:01_h and "8" in object 6060_h (Modes of Operation).

Control Word

In the Cyclic sync velocity mode, the bits in the object 6040_h (control word) have no special function. For the definition of the general bits, please refer to section 4.1.2 Control words.

Status Word

15 14	13	12	11	10	9 0
(see 4.1.3)	reserved	Target velocity ignored	(see 4.1.3)	reserved	(see 4.1.3)
MSB					LSB

In Cyclic sync velocity mode, the following bits in the object 6041_h (status word) have special function

Table 6-32

Bit	Value	Definition
12	0	Target speed be discarded
	1	Target speed be planner input

6.10.4 Function Description

In Cyclic sync velocity mode, the master controller sends the calculated target speed $60FF_h$ to the motor drive periodically and synchronously, and the speed is adjusted by the motor internally.

6.10.5 Configuration Examples

- Setting mode

  Write 2002_h:01_h=0, run mode 6060_h=0x09 to make it work in cyclic synchronous velocity mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 09 00 00 00 (set to cyclic synchronous velocity mode)
  Note: When switching mode, the motor should be in the deactivated state

- Write control word

  6040_h=0x06→0x07→0x0F, the motor runs, the upper computer needs to write the speed object $60FF_h$ in accordance with the synchronous cycle.
  Send: 601 2B 40 60 00 06 00 00 00 (Set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (Set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- The host computer sends the target speed $60FF_h$ (user unit/s) in accordance with the synchronization cycle

6.11 Cyclic Sync Torque Mode (CST)

In this mode, the upper controller sends the calculated target torque 6071_h periodically and synchronously to the servo drive, and the torque is adjusted by the servo internally. When the velocity reaches its limit, the servo will switch the velocity regulation stage.

6.11.1 Structure chart

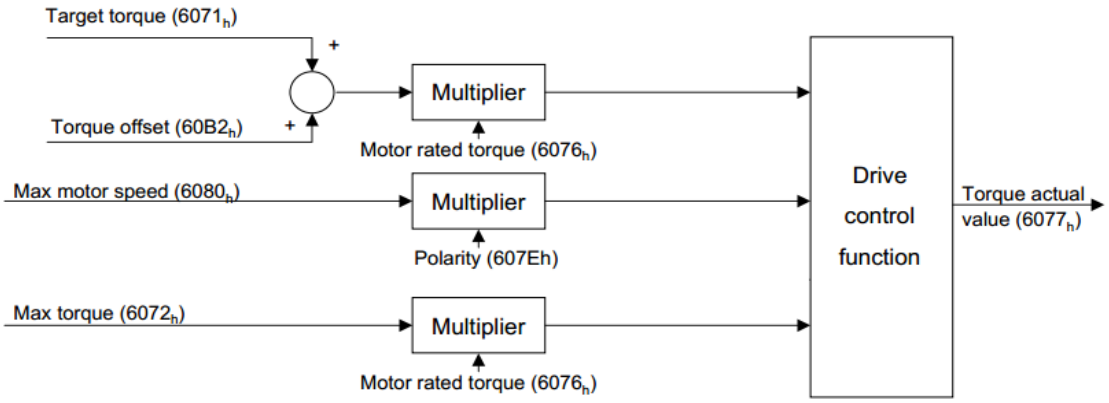


Figure 6-31

6.11.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-33

Object index	Description
$605A_h$ (Quick stop option code)	Quick stop option
$605D_h$ (Halt option code)	Halt option
6071_h (Target torque)	Target torque(unit: 0.1%)
6072_h (Max torque)	Maximum torque on the entire slope (acceleration, holding torque, braking)(unit: 0.1%)
6073_h (Max current)	Max current allowable (0.01A)
6080_h (Max motor speed)	Max motor speed (rpm)
$60B2_h$ (Torque offset)	Torque offset (0.1%)
6088_h (Torque profile type)	Torque ramp type (0-ramp, 2-none)
6077_h (Torque actual value)	Torque actual value(unit: 0.001N • m)

6.11.3 Control Instruction and Status Information

Enable

To enable this mode,the value of object 2002_h: 01_h and the value of object 6060_h (Modes of Operation) must be set to "0" and "8" respectively.

Control Word

In the Cyclic sync torque mode, the bits in the object 6040_h (control word) have no special function. For the definition of the general bits, please refer to section 4.1.2 Control words.

Status Word

15 14	13	12	11	10	9 0
(see 4.1.3)	reserved	Target torque ignored	(see 4.1.3)	reserved	(see 4.1.3)
MSB					LSB

In he Cyclic sync torque mode, the following bits in the object 6041_h (Status-word) have special function

Table 6-34

Bit	Value	Definition
12	0	Target torque be discarded
	1	Target torque speed be planner input

6.11.4 Function Description

In this mode, the upper controller sends the calculated target torque 6071_h periodically and synchronously to the servo drive.

6.11.5 Configuration Examples

- Setting mode

  Write 2002_h:01_h=0, run mode 6060_h=0x0A to make it work in cyclic synchronous torque

mode.
Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
Send: 601 2F 60 60 00 0A 00 00 00 (set to cyclic synchronous torque mode)
Note: When switching the mode, the motor should be in deactivated state

- Write control word

  Write control word 6040_h=0x06→0x07→0x0F, motor operation.
  Send: 601 2B 40 60 00 06 00 00 00 (Set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (Set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 0F 00 00 00 (Set 6040_h to 0xF to enable motor)

- The upper computer sends the target torque 6071_h according to the synchronous cycle (unit 0.1%)

6.12 NiMotion Mode

6.12.1 Summary

The NiMotion mode is different from CIA402 mode, without state machine management and controlled by multi-functional terminals.Manufacturer modes include Manufacturer position mode, Manufacturer velocity mode, Manufacturer torque mode. 2002_h:01_h (Control mode selection) is used to determine which mode the servo be in. See 6.1 Control mode overview for specific correspondence.

6.13 NiMotion Position Mode

6.13.1 Structural Chart

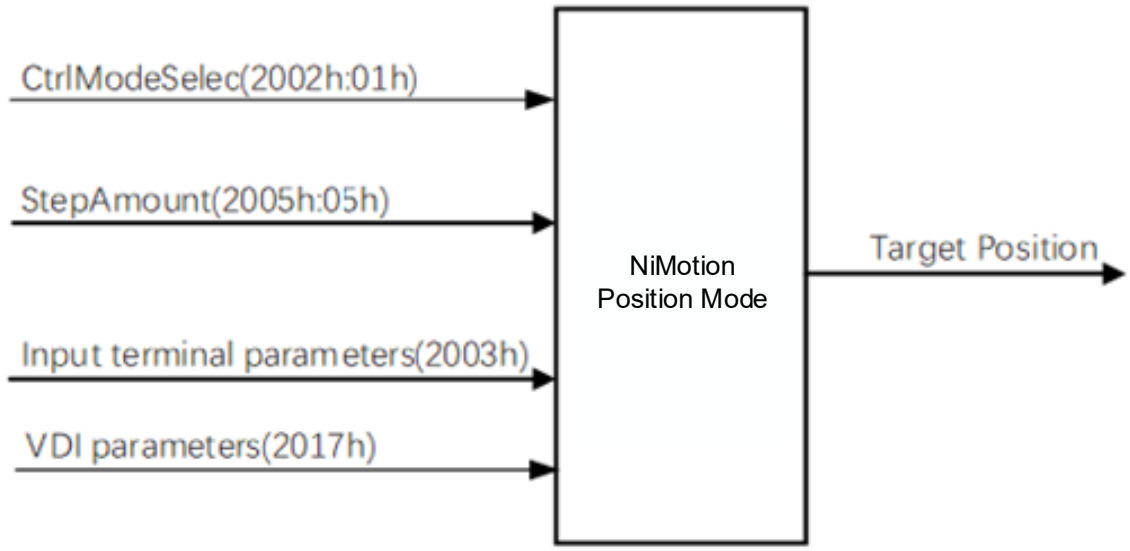


Figure 6-32

6.13.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-35

Object Index	Description
2002_h:01_h (CtrlModeSelec)	Operation mode selection
2005_h:05_h (StepAmount)	Step amounts setting (encoder unit)
2017_h (VDI parameters)	7.1.1 Virtual terminal input function configuration (see "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	7.1.1 Entity input terminal function configuration (see "7.1.1 Digital input" for details)

6.13.3 Control Instruction and Status Information

In this mode, there is no state machine, and the motor is enabled by the terminals.

Enable

The premise of this mode: the integrated low-voltage servo motor is switched to the Manufacturer position mode (2002_h: 01_h to 1).

1 Position source option:

Object (2005_h: 01_h) is the source of position instruction (2006_h: 01_h), meaning: 0-pulses, 1-step amounts;

2 set multi-function terminals (both physical terminals and virtual terminals are applicable. See 7.1 Digital input and output for specific configuration)

If position instructions is set to from 1, configure one input terminal as motor enable source, and set its valid logic level according to the actual situation(by default low-level-valid); and configure the other input terminal as step enabling source, set its valid logic level according to the actual situation, (by default low-level-valid);

If position instructions is set to from 0, (given by external pulses), it is necessary to configure the input terminals DI1 and DI2 as the corresponding pulse inputs ,and set the enable terminal of motor in addition.

Position configure

If the source of the position instruction is selected as 1, the step amount setting value needs to be set and written into the object 2005_h: 05_h;

If the position command source is selected as 0, pulse signals need to be input from terminals DI1 and DI2.

Operation

If position instructions is set to from 1, enable the motor by the terminal, and enable the step amounts through the terminal, then the motor will operates the distance (unit: Inc) preset in (2005_h: 05_h) for each stepping enabled;

If the position command is from 0, enable the motor by terminal, then the motor operates according to the pulses from the input terminal.

6.13.4 Function Description

In this mode, the motor operates a controllable distance according to the signal from the input terminals.

6.13.5 Configuration examples

Stepping amount given position control

- Parameter setting
  Setting to NiMotion position mode: 2002_h:01_h=1.
  Setting the position command source to step amount: 2005_h:01_h = 1.
  Setting the step amount: 2005_h:05_h = 50 (user units).
  Setting servo enable: 2003_h:03_h = 1, logic active low: 2003_h:04_h = 0.
  Set step amount enable: 2003_h:05_h = 20, logic is active on falling edge: 2003_h:06_h = 3.
  Send: 601 2B 02 20 01 01 00 00 00 (set to NiMotion position mode)
  Send: 601 2B 05 20 01 01 00 00 00 (Set position command source to step)
  Send: 601 2B 05 20 05 32 00 00 00 (Set step amount to 50)
  Send: 601 2B 03 20 03 01 00 00 00 (Set DI1 function to servo enable)
  Send: 601 2B 03 20 04 00 00 00 00 (Set DI1 logic to active low)
  Send: 601 2B 03 20 05 14 00 00 00 (Set DI2 function to step enable)
  Send: 601 2B 03 20 06 03 00 00 00 (Set DI2 logic to active low)
- Operation
  Motor enable: DI1 input low.
  Step amount enable: DI2 input a falling edge, step a distance, each input a falling edge will step a distance.

Quadrature pulse given position control

- Parameter Setting
  Setting to NiMotion position mode: 2002_h:01_h=1.
  Setting the source of position command as pulse: 2005_h:01_h=0.
  Setting input terminals.
    DI1 quadrature pulse input A: 2003_h:03_h=40.
    DI2 quadrature pulse input B: 2003_h:05_h=41.
    DI3 function number set to servo enable.
    Function selection for motor enable: 2003_h:07_h = 1.
    Logic selection active low: 2003_h:08_h = 0.
  Send: 601 2B 02 20 01 01 00 00 00 (set to NiMotion position mode)
  Send: 601 2B 05 20 01 00 00 00 00 (Set position command source to pulse)
  Send: 601 2B 03 20 03 28 00 00 00 (Set DI1 function to quadrature pulse input A)
  Send: 601 2B 03 20 05 28 00 00 00 (Set DI2 function to quadrature pulse input B)
  Send: 601 2B 03 20 07 01 00 00 00 (Set DI3 function to servo enable)
  Send: 601 2B 03 20 08 00 00 00 00 (Set DI3 logic to active low)
- Operation
  Motor enable: DI3 input low level.
  Pulse input control: terminals DI1 and DI2 input quadrature pulses, the motor is position controlled according to the input pulses.

6.14 NiMotion Velocity Mode

6.14.1 Structure chart

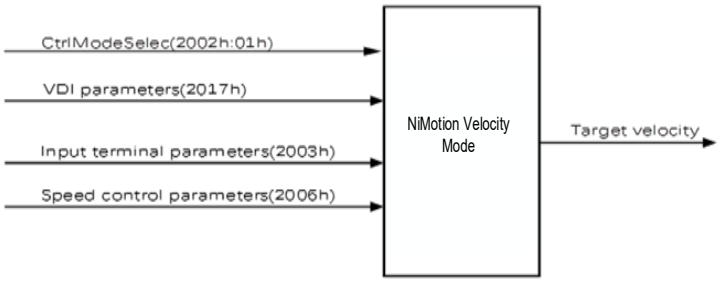


Figure 6-33

6.14.2 Objects Related

In this mode,the following objects need to be pay attention to:

Table 6-36

Object index	Description
2000_h:$0F_h$	Max motor velocity (rpm)
2002_h:01_h (CtrlModeSelec)	Operation mode selection
2006_h (Velocity control parameters)	2006_h:01_h: NiMotion velocity mode main velocity reference source, 0-digital reference, 3-duty cycle reference (20Hz-20kHz, 1kHz recommended), 4-50% duty cycle velocity regulation reference (20Hz-20kHz , 1kHz is recommended) 2006_h:02_h: NiMotion velocity mode auxiliary velocity reference source, 0-digital reference, 3-multi-velocity 2006_h:03_h: NiMotion velocity mode velocity command selection, 0-main velocity reference, 1- auxiliary velocity reference, 2-main velocity and auxiliary velocity reference (default 0-main velocity reference) 2006_h:04_h: velocity command digital given value, unit rpm 2006_h:07_h: Velocity command acceleration ramp time constant, unit ms, representing the time it takes to accelerate from 0rpm to 1000rpm. 2006_h:08_h: Velocity command deceleration ramp time constant, unit ms
2017_h (VDI parameters)	Virtual terminal input function configuration (see "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Function configuration of entity input terminals (see "7.1.1 Digital input") Note: multiple input terminals cannot be configured as the same switch at the same time, otherwise terminal setting failure will be reported

6.14.3 Control Instruction and Status Information

In this mode, there is no state machine, and the motor is enabled by the terminals.

Enable

The prerequisite for the operation of this mode: the integrated low-voltage servo motor is switched to NiMotion velocity mode (2002 $_h$:01$_h$ is set to 2).

1 Selection of velocity source:

The object is the main velocity command source (2006$_h$:01$_h$), meaning: 0-digital reference, 3-duty cycle reference, 4-50% duty cycle velocity control reference;

The object is the source of auxiliary velocity command (2006$_h$:02$_h$), meaning: 0-digital reference, 3-multi-velocity;

Through the velocity command selection (2006$_h$:03$_h$), the actual effective velocity is the main velocity command (effective by default), the auxiliary velocity command or the main and auxiliary velocity command.

2 Set up multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "7.1 Digital Input and Output" for terminal settings):

Configure one input terminal as motor enable, and configure its valid logic level according to the actual situation, the default low level is valid.

If the main velocity command source set in the previous step is 3 (duty cycle given), and the velocity command selection (2006$_h$:03$_h$) is 0, you need to set the corresponding input terminals DI1 and DI2 as duty cycle input and duty cycle direction Function;

If the main velocity command source set in the previous step is 4 (50% duty cycle velocity control setting), and the velocity command selection (2006$_h$:03$_h$) is 0, you need to set the corresponding input terminal DI1 as the duty cycle input function;

3 Velocity, acceleration and maximum velocity setting

If the main velocity command source is selected as 0-digital reference, the velocity reference value needs to be set and written to the object 2006$_h$:04$_h$;

If the main velocity command source is selected as 3-duty ratio setting, input pulse signal from DI1 (duty ratio) and DI2 (direction) input direction signal, 100% duty ratio corresponds to the maximum velocity (2000$_h$:0F$_h$), 0% duty cycle corresponds to velocity 0;

If the main velocity command source is selected as 4-50% duty cycle velocity control setting, you need to input pulse signal from DI1 (duty cycle), 0~50% corresponds to-limited velocity (2006$_h$:0B$_h$)~0, 50% ~100% corresponds to 0~+limit velocity (2006$_h$:0A$_h$);

Configure acceleration and deceleration 2006$_h$:07$_h$ and 2006$_h$:08$_h$

4 Run

According to the set motor enable terminal, enable the motor, and the motor will run at the set velocity.

6.14.4 Function Description

In this mode, the motor will run to the corresponding velocity according to the input of the terminal or the given digital quantity.

6.14.5 Configuration examples

Implementation of digital feeder speed regulation.

- Parameter setting

Setting to NiMotion velocity mode: 2002_h:01_h=2.
Setting the main speed command source to 0 (digital give): 2006_h:01_h=0 and setting the speed command selection 2006_h:03_h=0.
Setting the acceleration ramp time constant to 100ms: 2006_h:07_h=100 (unit ms).
Setting the deceleration ramp time constant to 100ms: 2006_h:08_h=100 (unit ms).
Setting the digital give to 60rpm: 2006_h:04_h=60(unit rpm).
Set DI1 terminal.
Select function as servo enable: 2003_h:03_h = 1.
Logic selection active low: 2003_h:04_h = 0.

Send: 601 2B 02 20 01 02 00 00 00 (set to NiMotion velocity mode)
Send: 601 2B 06 20 01 00 00 00 00 (Set master speed command source to 0)
Send: 601 2B 06 20 03 00 00 00 00 (Set speed command to master speed give)
Send: 601 2B 06 20 07 64 00 00 00 (Set acceleration ramp time constant to 100ms)
Send: 601 2B 06 20 08 64 00 00 00 (Set deceleration ramp time constant to 100ms)
Send: 601 2B 06 20 04 3C 00 00 00 (Set the digital give to 60rpm)
Send: 601 2B 03 20 03 01 00 00 00 (Set DI1 function to servo enable)
Send: 601 2B 03 20 04 00 00 00 00 (Set DI1 logic to active low)

- Operation

DI1 input low, motor speed regulation according to the digital given (2006h:04h)

To achieve duty cycle input velocity regulation.

- Parameter setting.

Setting the velocity mode to 2 (NiMotion speed): 2002_h:01_h = 2.
setting the main speed command source to 3 (duty cycle): 2006_h:01_h=3 and setting the speed command selection 2006_h:03_h=0.
Setting the acceleration ramp time constant to 100ms: 2006_h:03_h=100 (unit ms).
Setting the deceleration ramp time constant to 100ms: 2006_h:04_h=100 (unit ms).
Setting the maximum speed to 3000rpm: 2000_h:$0F_h$=3000 (unit rpm).
DI1 duty cycle input: 2003_h:03_h=44.

DI2 duty cycle direction input: 2003_h:05_h=45.
DI3 function number is set to servo enable.
Function selection for servo enable 2003_h:07_h = 1.
Logic select low active 2003_h:08_h = 0.
Send: 601 2B 02 20 01 02 00 00 00 (set to NiMotion velocity mode)
Send: 601 2B 06 20 01 03 00 00 00 (Set master speed command source to 3)
Send: 601 2B 06 20 03 00 00 00 (Set speed command to master speed give)
Send: 601 2B 06 20 07 64 00 00 00 (Set acceleration ramp time constant to 100ms)
Send: 601 2B 06 20 08 64 00 00 00 (Set deceleration ramp time constant to 100ms)
Send: 601 2B 00 20 0F B8 0B 00 00 (Set the maximum speed to 3000rpm)
Send: 601 2B 03 20 03 28 00 00 00 (Set DI1 function as duty cycle input)
Send: 601 2B 03 20 05 28 00 00 00 (Set DI2 function as duty cycle direction input)
Send: 601 2B 03 20 07 01 00 00 00 (Set DI3 function to servo enable)
Send: 601 2B 03 20 08 00 00 00 00 (Set DI3 logic to active low)

- Operation

  DI1 input a certain duty cycle square wave, DI3 input low level to enable the motor, the motor speed according to the input duty cycle.

6.15 NiMotion Torque Mode

6.15.1 Structural Chart

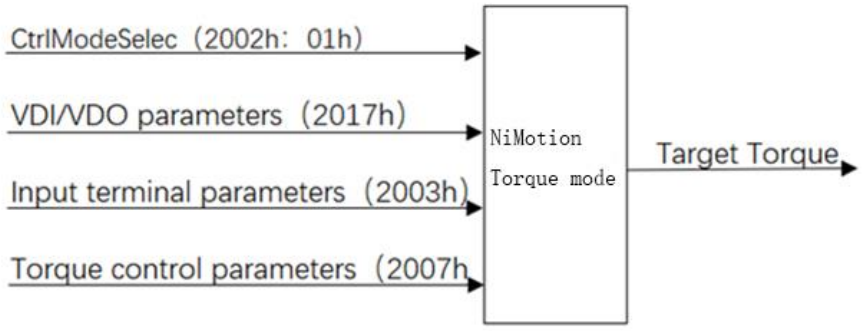


Figure 6-34

6.15.2 Objects Related

In this mode, the following objects need to be pay attention to:

Table 6-37

Object Index	Description
2002_h:01_h(CtrlModeSelec)	Operation mode selection
2007_h (Torque control parameters)	Sub-index: 2007_h:01_h:Torque source in Manufacturer torque mode, 0-digital given 2007_h:04_h : Torque command digital given (0.1%) 2007_h:$0A_h$: Positive limit of Internal torque (0.1%) 2007_h:$0B_h$: Negative limit of Internal torque (0.1%)

Object Index	Description
	2007_h:10_h : Positive velocity limit of torque controlrpm 2007_h:11_h : Negative velocity limit of torque controlrpm
2017_h (VDI parameters)	Virtual terminal input function configuration (see "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Function configuration of entity input terminals (see "7.1.1 Digital input") Note: multiple input terminals cannot be configured as the same switch at the same time, otherwise terminal setting failure will be reported

6.15.3 Control Instruction and Status Information

In this mode, there is no state machine, and the motor is enabled by the terminals.

Enable

The prerequisite for the operation of this mode: the integrated low-voltage servo motor is switched to NiMotion torque mode (2002_h:01_h is set to 3).

1 Torque source selection:：

The object is the main torque reference source (2007_h:01_h), meaning: 0-digital reference;

The object is the source of auxiliary torque reference (2007_h:02_h), meaning: 0-digital reference;

Through the torque command selection (2007_h:03_h), the actual effective is the main torque command (effective by default), auxiliary torque command or main and auxiliary torque command.

2 Set up multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "7.1 Digital Input and Output" for terminal settings):

Configure one input terminal as motor enable, and configure its valid logic level according to the actual situation, the default low level is valid

3 Torque setting and torque and speed limit setting

If you select the speed command source as the digital reference, you need to set the torque reference value and write it to the object 2007_h:04_h;

Configure positive and negative internal torque limit 2007_h:$0A_h$ and 2007_h:$0B_h$

Configure forward and reverse speed limit values 2007_h: 10_h and 2007_h: 11_h

4 Run

According to the set motor enable terminal, enable the motor, and the motor will run according to the set torque.

6.15.4 Function Description

In this mode, the motor will run to the corresponding torque through the enable control of the physical or virtual terminal and the torque digital quantity setting. When the running velocity of the motor exceeds the velocity limit (forward velocity limit value 2007_h:10_h and reverse velocity limit value 2007_h:11_h), it will automatically switch to velocity mode control and control the current velocity within the velocity limit . When it is detected that the given target torque is less than the average torque of the current velocity, the servo motor exits velocity control and returns to torque mode control.

6.15.5 Configuration Examples

- Parameter setting

  Setting to NiMotion torque mode: 2002_h:01_h=3.
  Setting the source of the main torque command to 0 (digital given): 2007h:01h=0.
  Setting the speed command selection to the main torque command: 2007_h:03_h=0.
  Setting the forward and reverse internal torque limit to 100%: 2007_h:$0A_h$=1000 and 2007_h:$0B_h$=1000 (unit 0.1%).
  Setting the forward and reverse speed limit to 3000 rpm: 2007_h:10_h=3000 and 2007_h:11_h=3000 (unit rpm).
  Setting the digital given value to 100%: 2007_h:04_h=1000 (unit 0.1%).
  Set DI1 terminal.
    Function selection for servo enable 2003_h:03_h = 1.
    Logic selection active low 2003_h:04_h = 0.
  Send: 601 2B 02 20 01 03 00 00 00 (set to NiMotion torque mode)
  Send: 601 2B 07 20 01 00 00 00 00 (set main torque command source to 0)
  Send: 601 2B 07 20 03 00 00 00 00 (Set torque command to main torque give)
  Send: 601 2B 07 20 0A E8 03 00 00 (Set positive internal torque limit to 100%)
  Send: 601 2B 07 20 0B E8 03 00 00 (Set reverse internal torque limit to 100%)
  Send: 601 2B 07 20 10 B8 0B 00 00 (Set forward speed limit value to 3000 rpm)
  Send: 601 2B 07 20 11 B8 0B 00 00 (Set reverse speed limit to 3000 rpm)
  Send: 601 2B 07 20 04 E8 03 00 00 (Set the digital limit to 100%)
  Send: 601 2B 03 20 03 01 00 00 00 (Set DI1 function to servo enable)
  Send: 601 2B 03 20 04 00 00 00 00 (Set DI1 logic to active low)

- Operation

  When DI1 is input low, the motor runs according to the digital given (2007_h:04_h) torque.

6.16 Multi-Segment Position Mode

6.16.1 Structure diagram

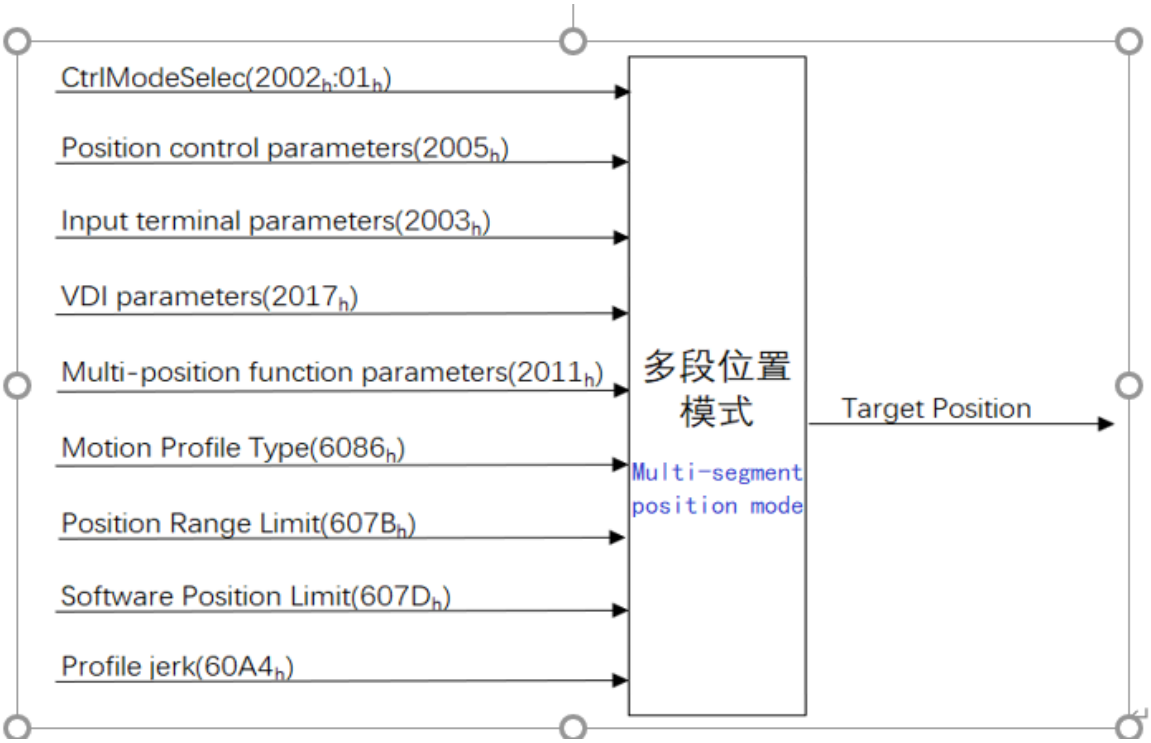


Figure 6-35

6.16.2 Related Objects

The following objects need to be noted in this mode:

Table 6-38

Object index	Description
2002_h:01_h (CtrlModeSelec)	Selection of operating mode
2005_h (Position control parameters)	2005h: 01h: Source of position command, 0-pulse, 1-step amount, 2-multi-segment position 2005h: 05h: Step amount setting (encoder unit)
2017_h (VDI parameters)	Virtual input terminal function configuration (see: "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Physical input terminal function configuration (see: "7.1.1 Digital input" for details) It should be noted that multiple input terminals cannot be configured as the same switch at the same time, otherwise the terminal setting failure will be reported.
2011_h (Multi-position function parameters)	Multi-segment position function parameter group

Object index	Description
607Bh (Position Range Limit)	Location range limitation (user unit)
607Dh (Software Position Limit)	Limitation of target location (user unit)
6086h (Motion Profile Type)	Select the type of planner curve that is the slope, If the value is "0", the impact (acceleration) will not be limited, that is, the trapezoid curve; If the value is "3", the value in 60A4h: 01h-02h will be used to limit the shock (acceleration), that is, S-shaped curve, this device only uses the 01h and 02h indexes.
60A4h (Profile jerk)	The profile is accelerated, see 6.3.2 for the specific implementation process

6.16.3 Control instructions and status information

There is no state machine management in this mode, the motor is enabled through the terminal.

Enable

The operating premise of this mode: the integrated low voltage motor is switched to the NiMotion position mode (2002_h: 01_h write 1).

1 Location source selection:

The object (2005_h: 01_h) is the source of the position command (2005_h: 01_h), meaning: 0-pulse, 1-step amount, 2-multi-segment position;

2 Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "7.1 Digital Input and Output" for terminal settings)

Set the position command source to 2, configure one input terminal to enable the motor, and configure its effective logic level according to the actual situation, the default low level is effective; set the other input terminal to multi-stage enable (function number 28) Configure its effective logic level, the default low level is effective;

3 Position setting

Set the corresponding operating mode and operating parameters of each segment through the multi-segment position function parameter group (2011_h).

4 run

First enable the motor through one terminal, and then enable the multi-segment position command through the other terminal. The motor will run according to the operating mode set by the multi-segment position function parameter group (2011_h)

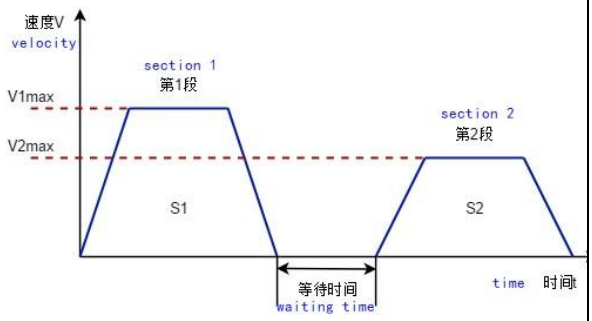
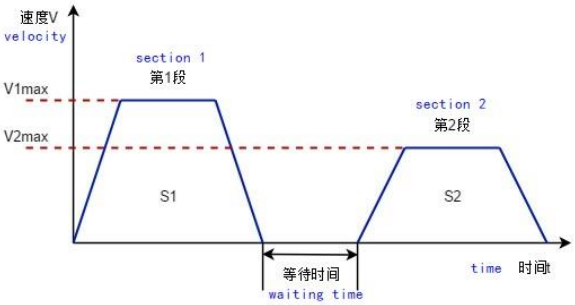
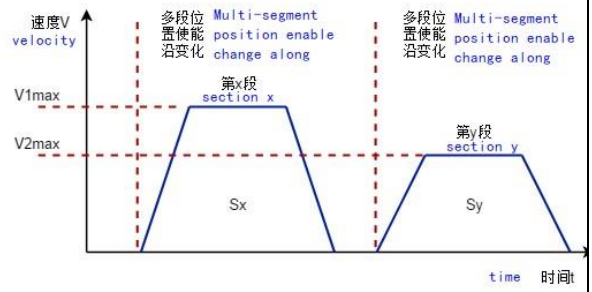
and the operating parameters of each segment.

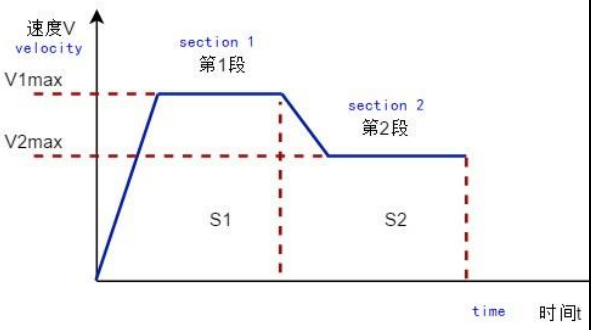
6.16.4 Function description

There are three operation modes for the multi-stage position mode, which are single operation, cyclic operation and DI switching operation. When using the multi-segment position function, the DI port must be set to servo enable (function number 1) and multi-segment enable (function number 28); during each segment operation, the servo enable must be valid; under non-DI switching operation mode, If the servo enable is valid, it must be ensured that the multi-segment enable is valid. If the multi-segment enable is turned off, the servo will give up the displacement command not sent by this segment and stop.

The detailed description of each operation mode is shown in the following table:

Table 6-39

Operation mode	Description	Running curve
Single operation $2011_h:01_h = 0$	Paragraphs 1 to 16 run once; Automatic segment number switching Waiting time can be set between segments; Multi-segment position enable is active level.	
Cyclic operation $2011_h:01_h = 1$	The number of times the 2011h: 06h set the starting segment to 2011h: 02h set the 2011h: 04h set to run; Automatic segment number switching Waiting time can be set between segments; Multi-level position enable is active level; Note: If the setting value of the end segment is smaller than the start segment, the first segment to the 16th segment will cycle once.	
DI switching operation $2011_h:01_h = 2$ **Note:** The physical terminal and the virtual terminal can be used in combination, and the combination	The position segment number switch is determined by the decimal 0 ~ 15 corresponding to the 4-digit binary number composed of multifunctional numbers 6, 7, 8, and 9;; The waiting time set between segments is invalid; Multi-segment position enable is effective along the change; The relationship between segment numbers and multifunction numbers is as	

<table>
<tr><th>Operation mode</th><th>Description</th><th>Running curve</th></tr>
<tr><td>must be 4 bits;
Physical terminal selection
DI1 = 6, DI2 = 7,
Virtual terminal
VI1 = 8, VI2 = 9</td><td>follows:
<table>
<tr><th>Fun9</th><th>Fun8</th><th>Fun7</th><th>Fun6</th><th>Section number</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td colspan="5">…..</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>16</td></tr>
</table></td><td></td></tr>
<tr><td>Run sequentially</td><td>The number of times the 2011_h: 06_h set the starting segment to 2011_h: 02_h set the 2011_h: 04_h set to run;
Automatic segment number switching
The waiting time set between segments is invalid;
Multi-level position enable is active level;
Note: If the setting value of the end segment is smaller than the start segment, the first segment to the 16th segment will cycle once.</td><td> 
</td></tr>
</table>

6.16.5 Configuration Examples

Multi-segment position-single run control

- Parameter setting

Setting to NiMotion position mode: 2002_h:01_h=1.
Setting the position command source to multi-segment position: 2005_h:01_h=2.
Set the multi-segment position function parameter group.
Multi-segment position operation mode: 2011_h:01_h = 0 (single operation).
Position command type selection: 2011_h:05_h = 0 (relative displacement).
Operating parameters of the 1st segment displacement.
moving displacement 2011_h:07_h = 10000 (user units), maximum permissible speed 2011_h:08_h = 500 (unit rpm), acceleration and deceleration time 2011_h:09_h = 500 (unit ms), waiting time after displacement completion 2011_h:$0A_h$ = 1000 (unit ms)
Setting of operating parameters for the 2nd to 16th segment with reference to the 1st segment displacement.
Setting motor enable: 2003_h:03_h = 1, logic is active low: 2003_h:04_h = 0.
Setting multi-segment enable: 2003_h:05_h=28, logic active low: 2003_h:06_h=0.
Send: 601 2B 02 20 01 01 00 00 00 (set to NiMotion position mode)
Send: 601 2B 05 20 01 02 00 00 00 (set position command source to multi-segment position)
Send: 601 2B 11 20 01 00 00 00 00 (Set multi-segment position operation mode to single operation)
Send: 601 2B 11 20 05 00 00 00 00 (Set position command type to relative

displacement)
Send: 601 23 11 20 07 10 27 00 00 (Set the 1st shift to 10000)
Send: 601 2B 11 20 08 F4 01 00 00 (Set the maximum speed of the 1st segment to 500 rpm)
Send: 601 2B 11 20 09 F4 01 00 00 (Set the acceleration and deceleration time of the 1st stage to 500ms)
Send: 601 2B 11 20 0A E8 03 00 00 (Set the waiting time for the completion of the first displacement to 1000ms)
Send: 601 2B 03 20 03 01 00 00 00 (Set DI1 function to servo enable)
Send: 601 2B 03 20 04 00 00 00 00 (Set DI1 logic to active low)
Send: 601 2B 03 20 05 1C 00 00 00 (Set DI2 function to multi-segment enable)
Send: 601 2B 03 20 06 00 00 00 00 (Set DI2 logic to active low)

- Operation

  Motor enable: DI1 input low.
  Multi-segment enable, DI2 input low, motor runs according to the operation mode and operation parameters set in the 2011_h multi-segment position function parameter group.

6.17 Multi-velocity mode

6.17.1 Structure

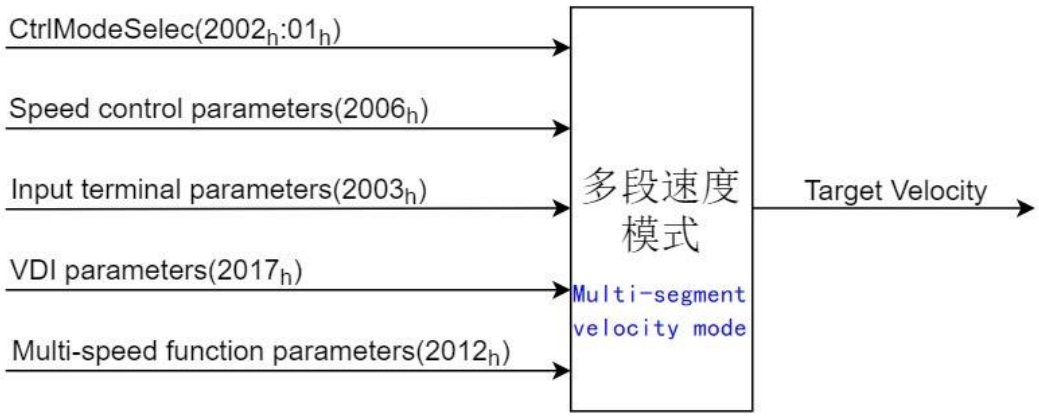


Figure 6-36

6.17.2 Related Objects

The following objects need to be noted in this mode:

Table 6-40

Object index	Description
2002_h:01_h (CtrlModeSelec)	Selection of operating mode
2006_h (Speed control parameters)	2006_h: 01_h: NiMotion velocity mode main speed reference source, 0-digital reference, 3-duty cycle reference (20Hz-20kHz, 1kHz recommended) 2006_h: 02_h: NiMotion velocity mode auxiliary speed reference source, 0-digital reference, 3-multi-step speed 2006_h: 03_h: NiMotion velocity mode speed command selection, 0-main speed reference, 1-secondary speed reference, 2-main speed and auxiliary speed reference (default 0) 2006_h: 04_h: digital reference value of speed command, unit rpm 2006_h: 07_h: Speed command acceleration ramp time constant, the unit is ms, which means the time it takes to accelerate from 0rpm to 1000rpm. 2006_h: 08_h: Speed command deceleration ramp time constant, unit ms
2017_h (VDI parameters)	Virtual input terminal function configuration (see: "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Physical input terminal function configuration (see: "7.1.1 Digital input" for details) It should be noted that multiple input terminals cannot be configured as the same switch at the same time, otherwise the terminal setting failure will be reported.
2012_h (Multi-speed function parameters)	Multi-stage speed function parameter group

6.17.3 Control instructions and status information

There is no state machine management in this mode, the motor is enabled through the terminal.

Enable

The operating premise of this mode: the integrated low-voltage servo motor is switched to the NiMotion velocity mode (2002_h: 01_h is set to 2).

1 Speed source selection:

The object is the source of auxiliary speed command (2006_h: 02_h), meaning: 0-number given, 3-multi-speed,

The speed command selection (2006_h: 03_h) actually takes effect is the main speed

command, auxiliary speed command, or main speed command.

2 Set up multi-function terminals (both physical and virtual terminals are applicable, see "7.1 Digital Input and Output" for terminal settings):

Configure one input terminal as motor enable, and configure its effective logic level according to the actual situation, the default low level is effective; configure another input terminal as multi-stage enable (function number 28), according to the actual configuration of its effective logic level , The default low level is effective.

3 Speed setting

Set multi-stage speed function parameter group (2012_h) to set the corresponding operating mode and operating parameters of each stage

4 Run

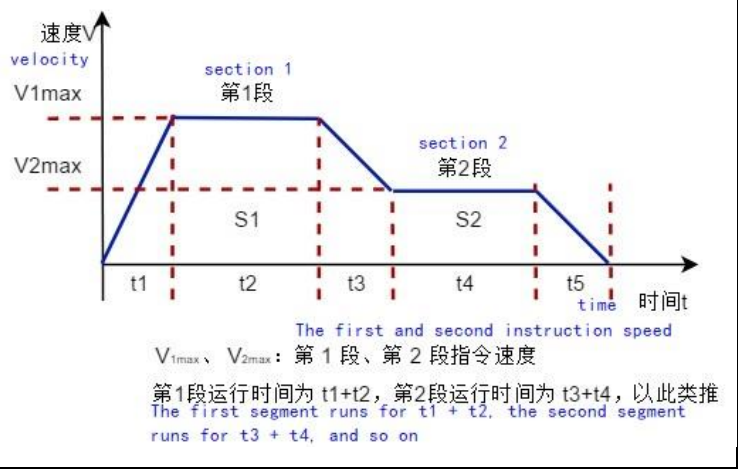
The motor is enabled through one terminal, and the motor runs according to the operating mode set by the multi-stage speed function parameter group (2012_h) and the operating parameters of each stage.

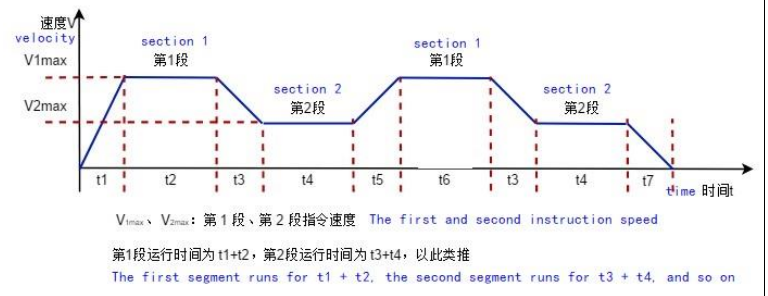
6.17.4 Function description

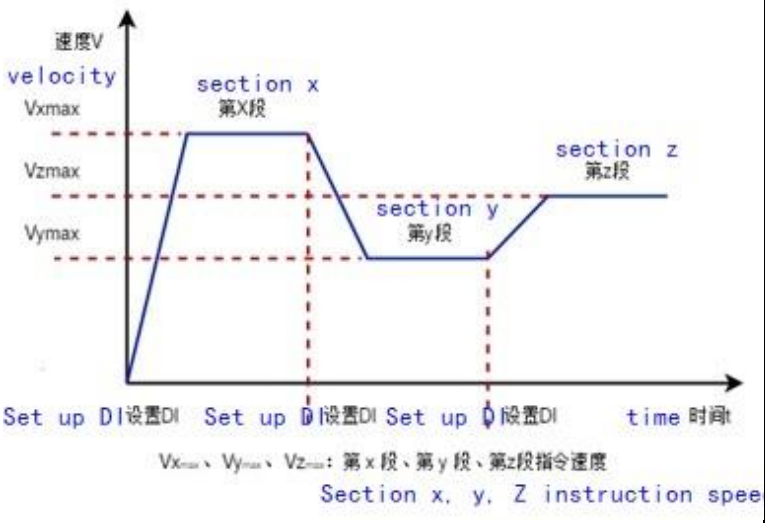
The multi-velocity mode has 4 operation modes, which are single operation, cycle operation and DI switching operation. When using the multi-stage speed function, a DI port must be set as the servo enable (function number 1). During each speed command operation, the servo enable must be effective. If the servo enable is turned off, the servo motor will stop according to the disable mode.

The detailed description of each operation mode is shown in the following table:

Table 6-41

Operation mode	Description	Running curve
Single operation 2012_h:01_h=0	Run once from paragraph 1 to paragraph 16; Segment number is automatically incremented and switched; Multi-segment enable is level effective.	

<table>
<tr><th>Operation mode</th><th>Description</th><th>Running curve</th></tr>
<tr><td>Cyclic operation
2012_h:01_h=1</td><td>The number of runs set in the first segment to 2012h: 02h and the end segment set in 2012h: 03h; the segment number is automatically incremented and switched;
Multi-segment enable is level effective.</td><td> 
</td></tr>
<tr><td rowspan="2">DI switch operation
2012_h: 01_h = 2

Note: The physical terminal and the virtual terminal can be used in combination, and the combination must be 4 bits;
Physical terminal selection
DI1 = 6, DI2 = 7
Virtual terminal
VI1 = 8, VI2 = 9</td><td>The speed segment number switch is determined by the decimal 0 ~ 15 corresponding to the 4-digit binary number composed of multifunctional numbers 6, 7, 8, and 9;
Servo enable is effective, that is, continuous operation;
The running time of each speed command is only determined by the speed interval switching interval.</td><td> 
</td></tr>
<tr><td>Correspondence table between segment number and multifunction number</td><td>
<table>
<tr><th>Fun9</th><th>Fun8</th><th>Fun7</th><th>Fun6</th><th>Section number</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td colspan="5">…..</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>16</td></tr>
</table>
</td></tr>
</table>

6.17.5Configuration Examples

Multi-segment velocity--single operation control

- Parameter setting.

  Setting to NiMotion velocity mode 2002_h:01_h=2.
  Setting the speed command source to multi-segment speed: 2006_h:02_h=3, 2006_h:03_h=1.
  Set the multi-segment speed function parameter group.
  Multi-segment speed operation mode 2012_h:01_h = 0 (single operation).
  Operating parameters for the 1st speed segment.
  speed command 2012_h:$0C_h$ = 500 (unit rpm), command run time 2012_h:$0D_h$ = 2000 (unit ms), the acceleration and deceleration time 2012_h:$0E_h$ = 500 (unit ms).
  The operating parameters of paragraphs 2 to 16 are set with reference to the operating parameters of paragraph 1 speed.
  Setting motor enable: 2003_h:03_h = 1, logic active low: 2003_h:04_h = 0.
  Set multi-segment enable: 2003_h:05_h=28, logic active low: 2003_h:06_h=0.
  Send: 601 2B 02 20 01 02 00 00 00 (set to NiMotion velocity mode)
  Send: 601 2B 06 20 02 03 00 00 00 (Set auxiliary speed command source to multi-segment speed)
  Send: 601 2B 06 20 03 01 00 00 00 (Set primary speed command as secondary speed command given)
  Send: 601 2B 12 20 01 00 00 00 00 (Set multi-segment position operation to single operation)
  Send: 601 2B 12 20 0C F4 01 00 00 (Set the 1st speed segment to 500 rpm)
  Send: 601 2B 12 20 0D D0 07 00 00 (Set the 1st segment running time to 2000ms)
  Send: 601 2B 12 20 0E F4 01 00 00 (Set the acceleration and deceleration time of the 1st stage to 500ms)
  Send: 601 2B 03 20 03 01 00 00 00 (Set DI1 function to servo enable)
  Send: 601 2B 03 20 04 00 00 00 00 (Set DI1 logic to active low)
  Send: 601 2B 03 20 05 1C 00 00 00 (Set DI2 function to multi-segment enable)
  Send: 601 2B 03 20 06 00 00 00 00 (Set DI2 logic to active low)

- Operation

  Motor enable: DI1 input low.
  Multi-segment enable, DI2 input low, the motor runs according to the operation mode set by the 2012_h multi-segment speed function parameter group and the operation parameters of each segment.

7 Special Functions

7.1 Digital Inputs and Output

The integrated low-voltage servo motor has input and output interfaces. The integrated digital inputs is used to connect auxiliary sensors of motion control, such as fretting limit switch, photoelectric switch, Hall sensor, etc. Digital output function can output 8mA current to drive solid-state relay or indicator light, etc.

7.1.1Digital Inputs

Table 7-1

Object Index	Description
2003_hInput terminal parameters	Function configuration of entity input terminals
2017_hVDI parameters	Virtual terminal input function configuration
2004_hOutput terminal parameters	DX1 terminal function configuration
200C:07_h(CommunicationVDI)	Virtual digital input enable, 0-disable, 1-enable
2031_h:01h(VDI_VirtualLevelCommSet)	VDI virtual voltage level given by communication
200Bh:05hMonitoredDiStates	Actual state of voltage level corresponding to entity terminalsbit0-DI1~bit3-DI4

Control instructions

The maximum number of digital entity input terminals is 4:

Fixed input DI1 ~ 3; when DX0 is configured as input and DoFunSelec in object 2004_h sub-index is 0, DX0 port is configured as input.

The communication virtual input terminal is set in parameter group 2017_h.The corresponding level state of communication virtual terminal(bit0-VDI1~bit15-VDI16) is set in 2031_h: 01_h .

- **Physical terminal**

Special function selection

The specific values are taken from the following table.

Table 7-2

Function Number	Function Definition
0	No definition
1	Servo Enable
2	Alarm Reset
3	Gain switching

Function Number	Function Definition
6	Multi-segment operation command switching 1
7	Multi-segment operation command switching 2
8	Multi-segment operation command switching 3
9	Multi-segment operation command switching4
12	Pause
14	Forward overtravel switch
15	Reverse overtravel switch
20	Step amount enable
28	Multi-segment enable
31	Home switch
33	Set home position
36	Analog input position/speed/torque control direction
38	Clear fault history
39	Clear power-up time
40	Quadrature pulse input A (fixed DI1)
41	Quadrature pulse input B (fixed DI2)
42	Pulse input (fixed DI1)
43	Pulse input direction (fixed DI2)
44	Duty cycle input (fixed DI1)
45	Duty cycle input direction (fixed DI2)
46	Incremental encoder input A (fixed DI1)
47	Incremental encoder input B (fixed DI2)
48	Incremental encoder input Z (fixed DI4)

Terminal Logic Selection

The logic selection of multi-function terminal is set in parameter group **2003_h** (entity input terminal) . Its specific values are as follows:

Table 7-3

Preset Value	Logic Definition
0	Low-level-valid
1	High-level-valid
2	Rising-edge-valid
3	Falling-edge-valid
4	Both rising-edge and falling-edge valid

Note:

1 "1-servo enable" is only valid in NiMotion mode, equivalent to writing enable control word to 6040h in 402 mode.

2 Different terminals can not set the same function number, if set incorrectly, it will report a parameter error fault.

3 If the working mode is set to HM mode of CiA402, no function number of 14, 15 and 31 will be reported as parameter failure.

4 The terminal function and logic selection is set to stop effective.

5 The digital input is only detected once every millisecond. Can not handle signal changes of less than 1ms at the input

6 If the detection entity terminal function number corresponding to the function number set to 46, 47 and 48, the servo motor initial angle 2000h:1Eh will automatically reset to 0, the first enable will make the servo motor automatically enter the state of finding the initial angle of the motor (402 mode state machine will return to Switch on disable state, you need to re-use the control word to make the state word into Operation enabled); when using external incremental encoder, the servo motor needs to find the initial angle of the motor when it is powered on for the first time, otherwise the motor cannot operate normally.

- **Virtual Terminal**

Virtual VDI terminal special function definition

The specific values are taken from the following table.

Table 7-4

Function Number	Function Definition
0	No definition
1	Servo Enable
2	Alarm reset
6	Multi-segment operation command switching 1
7	Multi-segment operation command switching 2
8	Multi-segment operation command switching 3
9	Multi-segment operation command switching 4
12	Pause
14	Forward overtravel switch
15	Reverse overtravel switch
20	Step amount enable
31	Home switch
33	Set zero point
36	Analog input position/speed/torque control direction
38	Clear fault history
39	Clear power-up time

Logic selection of virtual VDI terminal

The logic selection of multi-functional terminals is by the parameter group 2017_h (virtual input terminal). The specific values are as follows:

Table 7-5

Preset Value	Logic Definition
0	High-level-valid
1	Rising-edge-valid

Enable

The virtual VDI terminal of the integrated low-voltage servo motor is enabled when the index $200C_h$: 07_h is 1, and the virtual DI terminal function is closed when the value of $200C_h$: 07_h is 0, by terminal function number setting and logic selection in sub index 2017_h. When the logic selection is enabled, the function of corresponding function number is valid. The index 2031_h:01_h is the corresponding state of virtual input terminal (bit0-VDI1 ~ bit15-VDI16), which can only be given by communication. The specific definition is as follows:

Table 7-6 2031_h:01_h

Bit	15-7	6	5	4	3	2	1	0
Description	VDI16-VDI8	VDI7	VDI6	VDI5	VDI4	VDI3	VDI2	VDI1

Example

Setting input terminal DI1 to positive limit switch is as follow

- Set DI1 as positive limit switch, 2003_h: 03_h = 14
- Logic selection is low-level-valid, 2003_h: 04_h = 0

7.1.2 Digital output

Table 7-7 Related object settings

2004_hOutput terminal parameters	Output terminal parameters
2031_h:02h(DoStateCommSet)	Communication setting D0 output state (bit0~Do)

Control instruction

The digital output of this device has 1 DX1 that can be configured as a digital output, and the object 2004_h needs to be configured. When one group of DoFunSelec in the 2004_h sub-index is set to 1, the DX1 port is configured as an output, and the output state is given by the object 2031_h: 02_h communication.

Table 7-8 Output terminal function selection:

Function number	Function definition
0	Configure DX1 as input
1	Configure DX1 as output
2	Motor running stopped
3	Goal reached
4	Alarm Output

Function number	Function definition
5	Motor brake status output

Table 7-9 Output terminal logic selection:

Set value	Logical definition
0	Active low
1	Active high

For example

Example 1: Configure DX0 as user control and output high level:

- Configure DX1 as output, 2004_h:01_h =1;
- Configure DX1 logic to be active at high level, 2004_h:02_h =1;
- The communication setting DX1 output level, 2031_h:02_h = 0x01, at this time DX1 output high level

7.2 Analog Input

7.2.1 Analog Position Control

Table 7-10 Related objects

Object Index	Description
2002_h:01_h (CtrlModeSelec)	Control Mode Selection
2005_h(Position control parameters)	2005_h:01_h: Position command source, 3-Analog control mode 1, 4-Analog control mode 2 2005_h:$2A_h$: AI1 Deadband judgment time window, unit 1ms 2005_h:$2C_h$: Analog 10V corresponding position value (encoder unit)
2017_h (VDI parameters)	Virtual input terminal function configuration (see: "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	2003_h:17_h : AI1 Bias, unit 1mV 2003h:18h : AI1 Input filter time constant in 1ms 2003h:19h : AI1 Deadband, unit 0.01mV 2003h:1Ah : AI1 Multiplier, unit 0.001

Enable

There is no state machine management in this mode, and the motor is enabled to run via the terminals.

Prerequisite for this mode of operation: Integrated low-voltage servo motor switched to NiMotion position mode (2002_h:01_h write 1).

1. Position source selection “

   Set the position command source (2005_h:01_h), meaning: 3 - Analog control mode 1,

4 - Analog control mode 2.

2. Set multi-function terminals (both physical and virtual terminals are applicable, see "7.1 Digital Inputs and Outputs" for details of terminal settings)

   Configure one input terminal as motor enable, and configure its valid logic level according to the actual situation, the default low level is valid.

   If the set position command source is 3, you need to set the corresponding input terminal DI1 or DI2 for the analog input position control direction function (multi-function number 36).

3. Position Setting

   Configure the analog 10V corresponding position value (2005_h:$2C_h$), which is the absolute position of the motor.

4. Operation

   Enable the motor through one terminal, and the motor will run according to the analog input for the corresponding position.

   Note: Analog position control is absolute position control, you need to clear the current position to 0 before use.

Analog Position Control Method

There are two ways of analog input position control: analog control mode 1 and analog control mode 2.

1. Analog control method 1.

   The given position before deadband = symbol (corresponding to the level of multi-function number 36) * analog 10V corresponding position value * (input voltage * AI1 multiplier + AI1 bias)/10000, multi-function number 36 is valid for negative direction, invalid for positive direction.

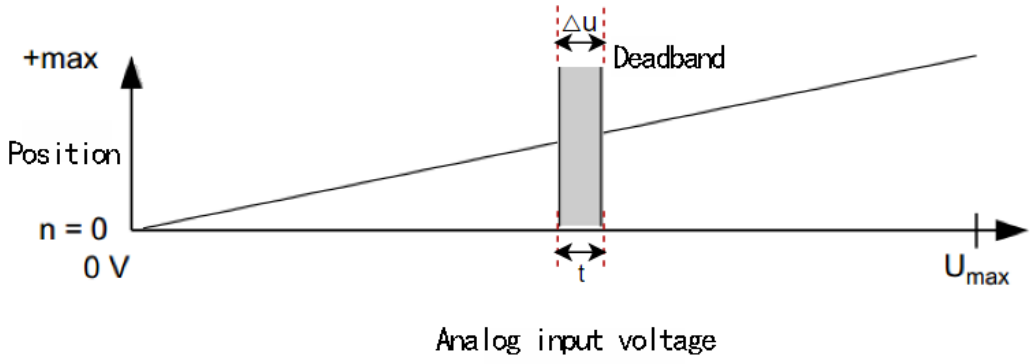


Figure 7-1

2. Analog control mode 2：

Input voltage 5~10V: given position before deadband = analog 10V corresponding position value * (input voltage * AI1 multiplier + AI1 offset)/10000.

Input voltage 0~5V: given position before deadband = - analog 10V corresponding speed value * (input voltage * AI1 multiplier + AI1 bias)/10000.

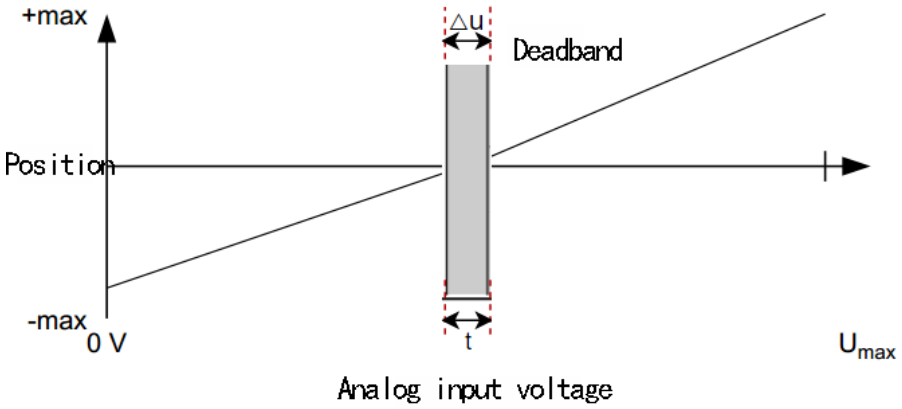


Figure 7-2

Note: 1. The given position before the dead zone is the analog voltage (△u/t) after input filtering and multiplier bias in the AI1 dead zone judgment time window (2005_h:$2A_h$) changes continuously greater than the AI1 dead zone (2003_h:19_h) setting, otherwise the given position maintains the value of the previous moment, through this setting can eliminate the position jitter caused by analog voltage jitter.

2.In analog control mode 2, the voltage needs to be set first, and then the enable command is sent for operation.

Configuration Examples

- Parameter Setting

  Setting to NiMotion position mode: 2002_h:01_h=1.
  Setting the position command source to 3: 2005_h:01_h=3：
  Setting motor enable: 2003h:03h=1, logic is active low: 2003h:04h=0.
  Set analog input position control direction: 2003_h:05_h=36, logic is valid for falling edge: 2003_h:06_h=3.

- Operation

  DI1 input low, the motor runs according to the position given by the analog quantity, DI2 controls the motor running direction.

7.2.2 Analog Velocity Control

Table 7-11 Related Objects

Object Index	Description
2002_h:01_h (CtrlModeSelec)	Control mode selection
2006_h (Speed control parameters)	2006_h:01_h : NiMotion velocity mode primary speed giving source, 1-analog control mode 1 2006_h:02_h : NiMotion velocity mode auxiliary speed giving source, 1-Analog control mode 2 2006_h:03_h : NiMotion velocity mode speed command selection, 0-Main speed given, 1-Auxiliary speed given 2006_h:04_h : Speed command digital given value, unit rpm 2006_h:07_h : Speed command acceleration ramp time constant unit ms, indicating the time used to accelerate from 0rpm to 1000rpm 2006_h:08_h : Speed command deceleration ramp time constant, Unit: ms 2006_h:$0A_h$: Forward speed threshold, unit rpm 2006_h:$0B_h$: Forward speed threshold, unit rpm
2017_h (VDI parameters)	Virtual input terminal function configuration (For details, see: "7.1.1 Digital input")
2003_h (Input terminal parameters)	2003_h:17_h : AI1 bias, unit 1mV 2003_h:18_h : AI1 input filter time constant in 1ms 2003_h:19_h : AI1 deadband in 1mV 2003_h:$1A_h$: AI1 multiplier, unit 0.001 2003_h:$1F_h$: Analog 10V corresponding speed value in rpm

Enable

There is no state machine management in this mode, and the motor is enabled to run via the terminals.

Prerequisite for operation: The integrated low-voltage servo motor is switched to NiMotion velocity mode (2002_h:01_h set to 2).

1. Speed source selection.

   Set the primary speed command source (2006_h:01_h) and the secondary speed command source (2006_h:02_h) to 1. Meaning: 1 - analog control mode.
   Select by speed command (2006_h:03_h) the actual effective is the main speed command (analog control mode 1) or auxiliary speed command (analog control mode 2), meaning: 0 - main speed given, 1 - auxiliary speed given.

2. Set multi-function terminals (both physical and virtual terminals are applicable, see "7.1 Digital Inputs and Outputs" for details of terminal settings).

   Configure one input terminal as motor enable, and configure its valid logic level according to the actual configuration, the default low level is valid.

If the speed command selection (2006_h:03_h) set in the previous step is 0, you need to set the corresponding input terminal DI1 or DI2 for the analog input speed control direction function (multi-function number 36); 3.

3. Speed, acceleration and maximum speed setting

   If the speed command selection (2006_h:03_h) is 0, the analog 10V corresponding speed value (2003_h:$1F_h$) needs to be configured.
   If the speed command selection (2006_h:03_h) is 1, the forward and reverse speed thresholds (2006_h:$0A_h$ and 2006_h:$0B_h$) need to be configured.
   Configuration of plus and minus speeds (2006_h:07_h and 2006_h:08_h).

4. Operation

   Enable the motor via one terminal, and the motor runs at the corresponding speed according to the analog input.

Analog speed control methods

There are two mode for analog input speed control: analog control mode 1 and analog control mode 2.

1. Analog control mode 1.

   The given speed before deadband = sign (corresponding to the level of multi-function number 36) * analog 10V corresponding speed value * (input voltage * AI1 multiplier + AI1 bias)/10000, multi-function number 36 is valid for negative direction, invalid for positive direction.

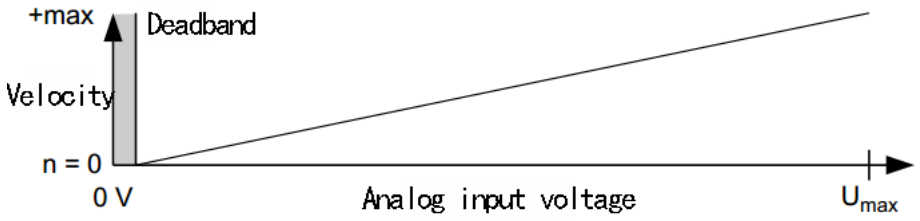


Figure 7-3

2. Analog control mode 2.

   Input voltage 5~10V: given speed before deadband = speed value corresponding to analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000.
   Input voltage 0~5V: the given speed before deadband = - the speed value corresponding to analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000.

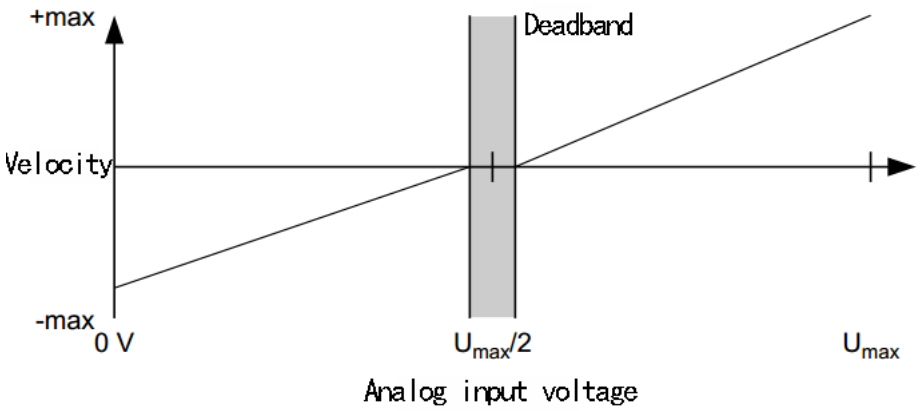


Figure 7-4

Note:In analog control mode 2, the voltage needs to be set first, and then the enable command is sent for operation.

Configuration example

- Parameter setting

  Setting to NiMotion velocity mode: 2002_h:01_h=2.
  setting the main speed command source to 1: 2006_h:01_h=1 and setting the speed command selection 2006_h:03_h=0.
  Setting the acceleration ramp time constant to 100ms: 2006_h:07_h=100 (unit ms).
  Setting the deceleration ramp time constant to 100ms: 2006_h:08_h=100 (unit ms).
  Setting the analog 10V corresponding speed value, 2003_h:$1F_h$=3000 (unit rpm).
  Setting DI1 and DI2 terminals.
  DI1 function selected as servo enable: 2003_h:03_h = 1.
  DI1 logic selection active low: 2003_h:04_h = 0.
  DI2 function selected for analog input speed control direction: 2003_h:05_h = 36.
  DI2 logic selection active low: 2003_h:06_h = 0.

- Operation:

  DI1 input low, the motor runs according to the speed given by analog (2006_h:06_h), and DI2 controls the motor running direction.

7.2.3 Analog torque control

Table 7-12 Related objects

Object Index	Description
2002_h:01_h(CtrlModeSelec)	Control mode selection
2007_h (Torque control parameters)	2007_h:01_h : NiMotion Torque Mode Primary Torque Feed Source, 1-Analog Control Mode 1 2007_h:02_h : NiMotion Torque Mode Auxiliary Torque Feed Source, 1-Analog Control Mode 2

Object Index	Description
	2007_h:03_h : NiMotion Torque mode torque command selection, 0-Main torque setting, 1-Auxiliary torque setting 2007_h:04_h : Torque command digital setting value (0.1%) 2007_h:$0A_h$: Positive internal torque limit (0.1%) 2007_h:$0B_h$: Reverse internal torque limit (0.1%) 2007_h:10_h : Torque control positive speed limit value (rpm) 2007_h:11_h : Torque control reverse speed limit value (rpm)
2017_h (VDI parameters)	Virtual input terminal function configuration (For details, see: "7.1.1 Digital input")
2003_h (Input terminal parameters)	2003_h:17_h : AI1 bias, unit 1mV 2003_h:18_h : AI1 input filter time constant, unit 1ms 2003_h:19_h : AI1 deadband, unit 1mV 2003_h:$1A_h$: AI1 multiplier, unit 0.001 2003_h:20_h : Analog 10V corresponding torque value, unit 0.1%

Enable

There is no state machine management in this mode, and the motor is enabled to run via the terminals.

Prerequisite for operation: The integrated low-voltage servo motor is switched to NiMotion torque mode (2002_h:01h set to 3).

1. Torque source selection.

   Set the primary torque command source (2007_h:01_h) and the secondary torque command source (2007_h:02_h) to 1. Meaning: 1 - analog control mode.
   The actual effective by torque command selection (2007_h:03_h) is either the main torque command (analog control mode 1) or the auxiliary torque command (analog control mode 2), meaning: 0 - main torque given, 1 - auxiliary torque given.

2. Set multi-function terminals (both physical and virtual terminals are applicable, see "7.1 Digital Inputs and Outputs" for details of terminal settings).

   Configure one input terminal as motor enable, and configure its valid logic level according to the actual situation, the default low level is valid.
   If the torque command selection (2007_h:03_h) set in the previous step is 0, the corresponding input terminal DI1 or DI2 should be set as the analog input torque control direction function (multi-function number 36).

3. Torque giving and torque and speed limit setting

   If the torque command selection (2007_h:03_h) is 0, it is necessary to configure the analog 10V corresponding torque value (2003_h:20_h).
   If the torque command selection (2007_h:03_h) is 1, the forward and reverse internal torque limit values (2007_h:$0A_h$ and 2007_h:$0B_h$) need to be configured.
   Configure forward and reverse speed limit values (2007_h:10_h and 2007_h:11_h).

4. Operation

   Enable the motor via one terminal, and the motor will be controlled with the corresponding torque according to the analog input.

Analog torque control mode

There are two modes of analog input torque control: analog control mode 1 and analog control mode 2.

1. Analog control mode 1:

   The given torque before deadband = sign (corresponding to the level of multi-function number 36) * analog 10V corresponding torque value * (input voltage * AI1 multiplier + AI1 bias)/10000, Note: Multi-function number 36 is valid for negative direction and invalid for positive direction.

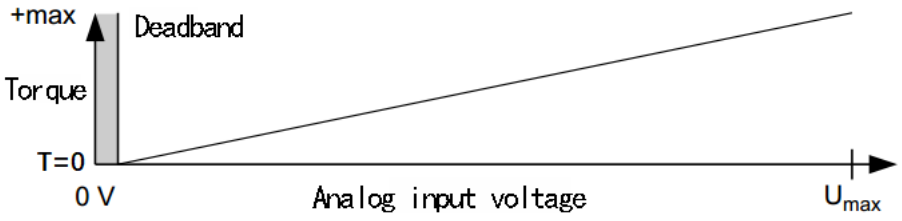


Figure 7-5

2. Analog control mode 2:

   Input voltage 5~10V: given torque before deadband = analog 10V corresponding torque value * (input voltage * AI1 multiplier + AI1 offset)/10000.
   Input voltage 0~5V: given torque before deadband = - analog 10V corresponding torque value * (input voltage * AI1 multiplier + AI1 bias)/10000.

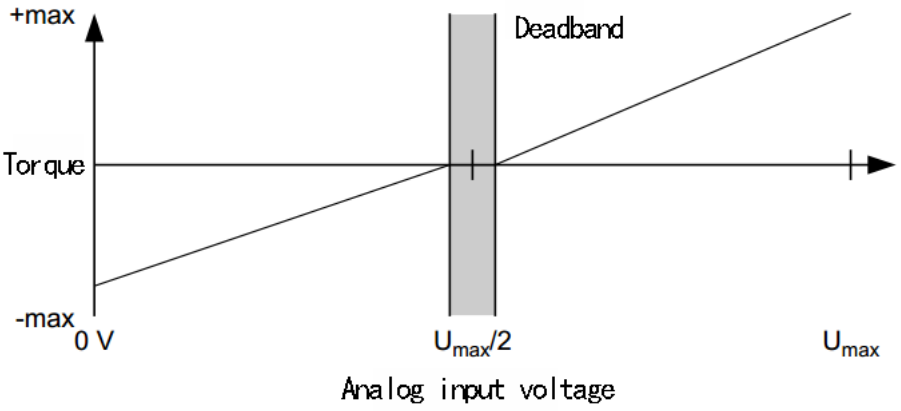


Figure 7-6

Note:In analog control mode 2, the voltage needs to be set first, and then the enable command is sent for operation.

Configuration example

- Parameter setting

  Setting to NiMotion torque mode: 2002_h:01_h=3.
  Setting the source of the main torque command to 1: 2007_h:01_h=1.
  Setting the torque command selection to the main torque command: 2007_h:03_h=0.
  Setting the forward and reverse internal torque limit to 100%: 2007_h:$0A_h$=1000 and 2007_h:$0B_h$=1000 (unit 0.1%).
  Setting forward and reverse speed limit to 3000rpm: 2007_h:10_h=3000 and 2007_h:11_h=3000 (unit rpm).
  Setting the analog 10V corresponding torque value: 2003_h:20_h=1000 (unit 0.1%).
  Setting DI1 and DI2 terminals.
  DI1 function selected as servo enable: 2003_h:03_h = 1.
  DI1 logic selection active low: 2003_h:04_h = 0.
  DI2 function selected for analog input torque control direction: 2003_h:05_h = 36.
  DI2 logic selection active low: 2003_h:06_h = 0.

- Operation

  DI1 input low, the motor runs according to the torque given by the analog quantity (2007_h:04_h), DI2 controls the motor running direction.

7.2.4 Analog quantity switching function

When the control bus communication is abnormal, the motor cannot respond normally to the control instructions issued by the main control system. At the same time, the main control system cannot identify the operating status of the controlled motor. The controlled motor reports a communication fault and stops running. The indicator light shows the status as a fault alarm. If the application scenario requires that the output speed or position of the motor can still be controlled in the event of communication interruption, the system control mode needs to be switched from bus communication control to analog control at physical terminals, that is, by adjusting the size of the analog voltage, the output size of the controlled quantity can be controlled. Therefore, the analog quantity switching function is suitable for this application scenario.

- Switch the PV mode to the analog speed control function

  1.Select the analog speed mode (Mode 1 or Mode 2) that needs to be configured and set it through the object dictionary parameter 2006_h:03_h.
  2.Set the corresponding speed value of the analog quantity 10V, that is, 0 to 10V corresponds to the set speed range.
  3.Set the analog speed switching function number 50, that is, set 2003h:03h to 50, which is effective at low levels.
  4.Connect the analog input voltage of the AI port.
  5.In the shutdown state, when the DI1 input level is valid, the motor automatically switches to the analog speed mode. By adjusting the analog voltage value, the motor speed changes accordingly.

The specific settings of the relevant parameters for analog speed control mentioned above can be referred to in Section 7.2.2.

- Switch the PP mode to the analog position control function

  1.Select the position mode of the analog quantity to be configured (Mode 1 or Mode 2), and set it through the object dictionary parameter 2005_h:01_h.

  2.Set the position value corresponding to the analog quantity 10V.

  3.Set the analog quantity position switching function number 51, that is, set 2003_h:03_h to 51, which is effective at low levels.

  4.Connect the analog input voltage of the AI port.

  5.In the shutdown state, when the DI1 input level is valid, the motor automatically switches to the analog position control mode. By adjusting the analog voltage value, the motor position changes accordingly. (Note: The analog quantity position is given as the absolute position. Please confirm the current position value before switching.)

Note:The analog quantity switching function takes effect when the motor is off! If the switch occurs under a communication failure, the communication failure needs to be configured to handle the fault shutdown. The virtual terminal function number cannot be configured as the enable function number; otherwise, the switching function will be invalid!

7.3 Brake Setting

The brake is to prevent the servo motor shaft from moving when the servo motor is in the disabled state, thereby locking the position of the motor. The integrated low-voltage servo motor automatically switches the state of the brake output according to the CiA402 state machine, as shown below:

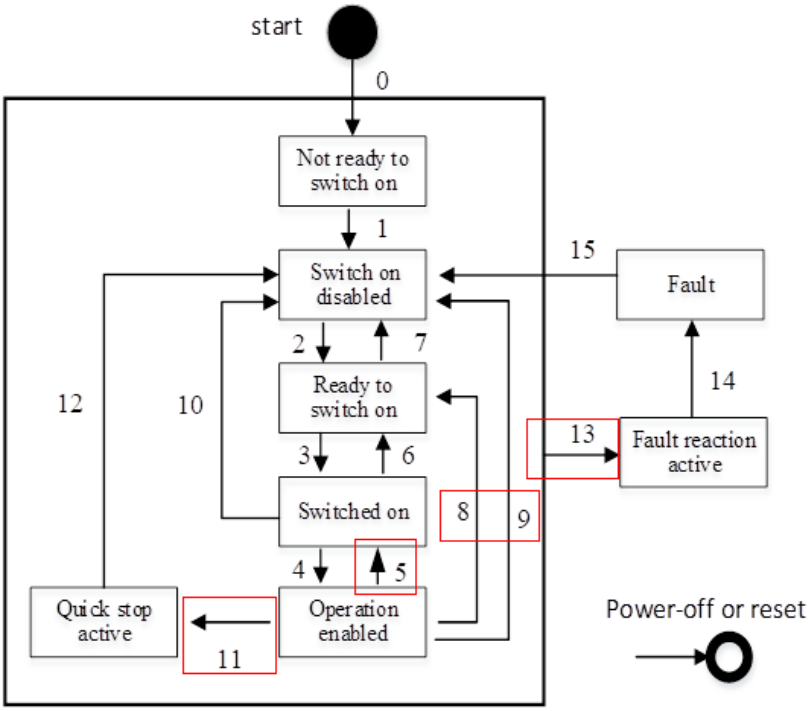


Figure 7-7

Note:When the integrated servo state machine leaves the running state (Operation enable), the brake output is closed.

Table 7-13

Bit	Name	Description

0	Servo ready	When the status word Bit0~bit3 are all 1, the brake output terminal is turned on.
1	Connect the main circuit power	
2	Fast shutdown	
3	Servo operation	

Related objects

Table 7-14

Object index	Description
2002_h:$0A_h$ (BrkOffToStopMotorOffDelay)	The delay from when the brake output is closed to when the motor is not energized, unit ms
$200A_h$:$1F_h$ (BrakeDetectionEn)	Brake detection enable, 0-zero-gravity preset, 1-enable gravity preset;
$200A_h$:20_h (BrakeLoadGravity)	Brake load gravity (default value of gravity, torque output by the motor when the brake is opened), 0.1% of rated torque
$200D_h$:$0B_h$ (BrakeLoadTun)	Brake load identification, 0-disable gravity identification, 1-enable online gravity identification

Note:

1. **When the servo motor is used for a vertical axis, the gravity preset and online gravity recognition functions need to be turned on, and the gravity preset value is used to prevent the brake loosening and the load drop when the motor starts due to the load's own weight or external force.**
2. **When the servo motor is used for a vertical axis, it is necessary to properly set the brake output to the motor's non-energized delay to prevent the load from falling due to its own weight or external force when the motor is stopped from operation to stop or failure.**
3. **Enable online real-time identification of load gravity. When the servo motor speed is below 300rpm, the default value of gravity ($200A_h$:20_h) will be automatically updated.**

Enable and disable

1 The brake output control can be enabled through the object dictionary 2004h:05h. When the function number is set to 30, the brake function will be automatically enabled;

When the terminal function of DX1 (2004h:01h) is configured as multi-function number 5 (motor brake status output) and the logic is active high, DX1 outputs high level when the brake is opened.

2 The brake output control can be closed through the object dictionary 2004h:05h. When the function number is set to 0, the brake function will be automatically closed.

When the terminal function of DX1 (2004h:01h) is configured as multifunction number 5 (motor brake status output) and the logic is active high, DX1 outputs low level when the

brake is closed.

Duty cycle

The output of the brake triggers a PWM signal with an adjustable duty cycle. The duty cycle (the ratio of pulse time to cycle time) is set in 2002_h:04_h, and its value is a percentage value, which can be selected between 1 and 100 (it is recommended to configure above 90). When the value is 100, the brake output will be continuously connected.

7.4 I^2t Overload Protection of Motor

After the integrated low-voltage servo motor is powered on, heat will be generated continuously due to the thermal effect of current, and be released to the surrounding environment at the same time. The temperature of the motor will rise when the heat generated exceeds the heat released, which will cause the motor to burn out. The purpose of i^2t overload protection of motor is to ensure the motor operating normally under the temperature limit and to prevent the motor from damage.

Objects Related

Table 7-15

Object Index	Description
2000_h:09_hRatedCurrent	Rated Current (unit0.01A)
2000_h:$0A_h$MaximumCurrent	MaximumCurrent (unit0.01A)
2000_h:$0B_h$MaxCurrentDurTime	MaxCurrentDurTime (unitms)

Enable

To enable this protection, the three object items above must be described correctly. When the motor operates above the rated current, the servo drive detects and calculates the time integral of I^2.When the integral value is greater than the product of the set maximum current (2000_h: $0A_h$) and the maximum current duration (2000_h: $0B_h$), the motor will give an overload warning and limit the torque output to the rated value. If these conditions are not met, the i^2t function will be disabled.

7.5 Brake Configuration

When the directions of torque and the motor velocity are opposite, the energy is fed back to the drive from the motor,which causing the bus voltage increasing. If it reaches the braking point, the energy will be consumed by the braking resistance, otherwise the drive will be damaged. The There are two types of braking resistance: internal and external (setting in 2002_h: 13_h: 0 ~ internal, 1 ~ external),which can not be used at the same time. The specification of internal brake resistance is 15 , 25W, and the power toleration is 12.5W.

7.4.1Without External Load Torque

When the motor reciprocates, the braking energy will be converted to electrical energy and fed back into the bus capacitance. If the bus voltage exceeds the braking voltage (2001_h: $0F_h$), the braking resistance will consume the redundant feedback energy. The following table shows the braking energy E generated by the motor from 3000rpm to 0 rpm:

Table 7-16

Capacity	Servo Motor Model	Rotor Inertia ($10^{-4}kgm^2$)	Braking Energy (J)
26W	BLM4203	0.14	5.4

The selection of braking resistance power P is related to the cycle T (unit s) of motor reciprocating ,the rotation velocity,the motor inertia and the ratio N of load inertia to the motor inertia. The specific calculation formula is as follows:

$$P = \frac{2\times(N+1)\times J*n*n}{182*T}$$

In case of the servo motor operating from 3000rpm to 0rpm, the specific speed curve is as follows:

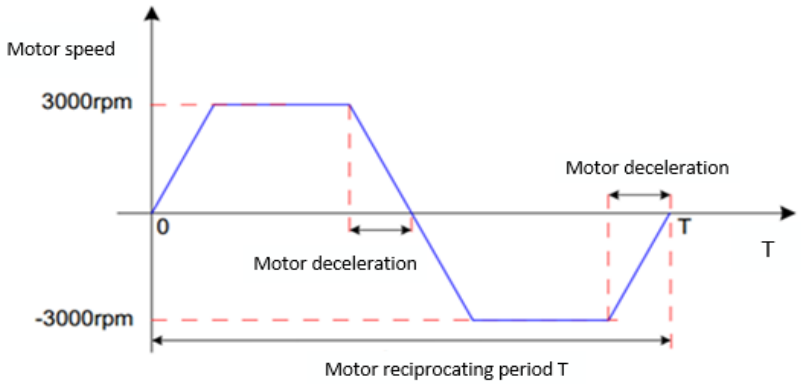


Figure 7-8

Taking 26W motor as an example, assuming that the period of reciprocating motion T = 2s, the maximum velocity is 3000rpm, and the load inertia is 5 times of the motor inertia, the braking resistance power is required:

$$Pa = \frac{2\times(N+1)\times J*n*n}{182*T} = \frac{2\times(5+1)\times 0.14*0.0001*3000*3000}{182*2} = 4.15W$$

Pa is less than 5.4W,which is the power limit of the internal braking resistance, so the internal resistance meets this requirements.

If the load inertia in the above conditions is changed from 5 times to 10 times, and other conditions remained, the braking resistance power Pb is required:

$$Pb = \frac{2\times(N+1)\times J*n*n}{182*T} = \frac{2\times(10+1)\times 0.14*0.0001*3000*3000}{182*2} = 7.62W$$

Pb is greater than the power limit of the internal braking resistance. Therefore, an external braking resistance which is reduced by 70% is required. i.e. Pb / (1-70%) = 25.4w.

7.4.2With External Load Torque

In some special cases, the rotation direction of the motor is opposite to the torque,the motor will output energy, and feed back to the servo drive.

Taking 26W (rated torque 0.08Nm) capacity as an example, when the external load torque is 60% of the rated torque and the velocity reaches 1000rpm, the power fed back to the servo drive is calculated as follows:

$$P=\frac{2\pi\times n\times M}{60}\times 60\% = \frac{2\pi\times 1000\times 0.08}{2}\times 60\%=5.024W$$

Considering that the braking resistance needs to be derated by 70%, the external braking resistance power P is actually 5.024 / (1-70%) = 16.7w,.

Note: n is rotation velocity (rpm), M is torque (N.m)

7.6 Position Recovery

The absolute position can be restored after the motor is powered off and on again.

When the position recovery mode (2005_h:18_h) value is 1, it is suitable for single-turn reciprocating motion. When the motor is in the negative direction relative to the zero point, the return to zero point needs to run in the positive direction; when the motor is in the positive direction relative to the zero point, the return The zero point needs to run in the negative direction.

For single-turn position recovery, you can set the maximum forward rotation angle (2005h:1Dh) to change the direction of returning to zero, as shown in the following figure:

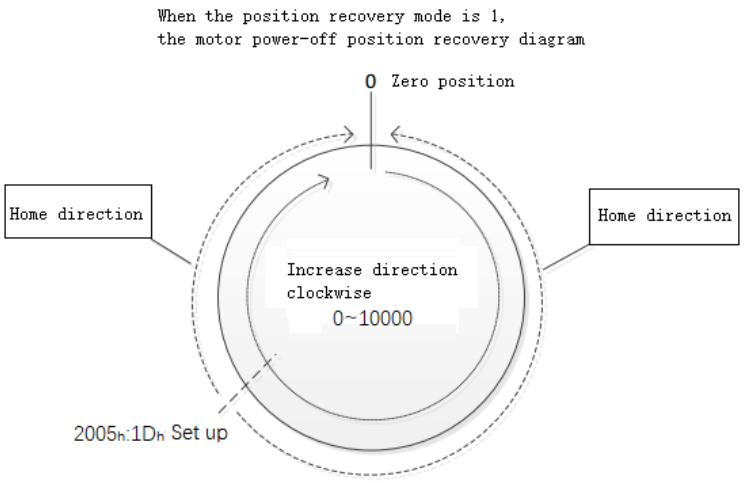


Figure 7-9

When the position recovery mode (2005_h:18_h) value is 0, it is suitable for multi-turn motion trajectory, and the motor can recover the current absolute position.

7.7 Parameter Saving and Restore

Parameter management is written according to the CiA301 standard, 1010_h is the function object for saving parameters, and 1011_h is the function object for reading parameters.

7.7.1 Parameter Saving

In the sub-index of 1010h, a value of 1 indicates that the corresponding save operation is supported. 0x65766173 and 0x9A899E87 are the save parameter commands. Write the save parameter command word to the corresponding sub-index of 1010_h. Return 1 to save the parameters successfully, and return non-1 to unsuccessful.

For example: 1010_h:01_h write 0x65766173 to save the parameters to the user area,, 1010_h:$0B_h$ write 0x65766173 to save the NiMotion parameters.

7.7.2 Parameter Restore

In the sub-index of 1011_h, a value of 1 indicates that the corresponding recovery operation is supported. 0x64616F6C and 0x9A899E80 are the recovery parameter commands. Write the recovery parameter command word to the corresponding sub-index of 1011_h. Return 1 means the recovery parameter is successful, and return non-1 means unsuccessful.

For example: 1011_h:01_h write 0x64616F6C to restore the factory area parameters to the user area.

7.8 Power Limit Function

The power limit function ensures that the output power of the motor does not exceed the rated power, so that the motor can run continuously and stably.

When 2000_h: 24_h (bit0 to bit14) =0, the power limiting function is disabled. When 2000_h: 24_h (bit0~bit14) ≠0, it indicates that the power limiting function is enabled, and its value is set to the torque value that can be output by one times the rated current at low speed. The power limiting function is enabled by default. Rated power 2000_h: 08_hIndicates the maximum power that can be output after the power limit is enabled. The highest bit (bit15) in 2000_h:24_hindicates whether the power limit alarm function is enabled. 0 indicates off and 1 indicates on. After this function is enabled, the electric machine reports a power limiting fault in the power limiting state. Factory default disable power limit alarm function.

For example, if the power limit is set to 40W, set 2000_h: 08_hin the object dictionary to 4(40W). If the torque value of 1x rated current output in the measured torque-frequency curve is 0.77Nm(the torque value near 100rpm in the characteristics can be referred to), then set 2000_h: 24_h=77(indicating 0.77Nm), so that the motor has started the function of limited power of 40W. If you want to enable the power limiting alarm at this time, set 2000_h:24_h=32845 (1000 0000 0100 1101). When the electric machine is in the power limiting state, the power limiting alarm will be performed.

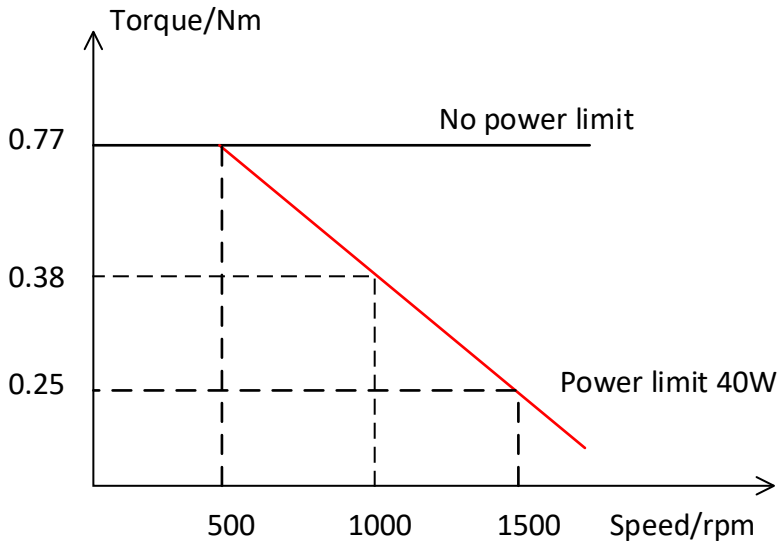


Figure 7-10

7.9 Modify CAN-ID Online

This function can modify the CAN-ID of the motor on the CAN bus by sending SDO messages.For example: Suppose the CAN-ID of all the motors on the CAN bus is 1. Set the CAN-ID of the integrated motor with serial number 64110 to 3.

Table 7-8

Step	Function	Message	Note
1	Send a message to confirm the serial number of the target	0601 23 7D 2F 00 6E FA 00 00	The 6E FA of message is the serial number 64110.

	motor		
2	Modify the CAN-ID of the target motor	0601 23 7E 2F 00 03 00 00 00	
3	Save Parameters	0601 23 10 10 01 73 61 76 65	
4	Restart takes effect		

When the CAN-ID on the CAN bus is not number 1 and the CAN-ID of the set motor is unknown, the master station needs to send a configuration message containing all the ID addresses on the CAN bus. The example is as follows.

Table 7-9

Step	Function	Message	Note
1	Send a message to confirm the serial number of the target motor	6xx 23 7D 2F 00 6E FA 00 00	xx represents all ID addresses, such as 01,02,03...
2	Modify the CAN-ID of the target motor	6xx 23 7E 2F 00 03 00 00 00	
3	Save Parameters	6xx 23 10 10 01 73 61 76 65	
4	Restart takes effect		

7.10 Absolute Value Encoder

The integrated motor detects the battery voltage once each time it is powered on and updates it to the parameter $200F_{h:}0B_{h}$. When the battery voltage of the integrated motor drops below the specified value during operation, the integrated motor will issue a warning and fault information.

Table 7-10

Battery Voltage	Alarm Information	Alarm Recovery	Battery Replacement Method
2.6V＞DC＞2.4V	0x7302	The counting of multi-turn encoders may have errors. Recommend replacing the battery.	1.The integrated motor is powered on and in a non-operating state； 2.Replace the battery； 3.The alarm is automatically lifted.
DC≤2.4V	0x7301	Please replace the battery immediately.	
DC≥2.6V	0x7309	Power off and restart	

7.11 Communication Watchdog Function

The communication watchdog function is designed to enable the servo motor to promptly alarm and stop rotating when communication is disconnected to prevent accidents.

When parameter H200C-1B = 0, this function is turned off. When H200C-1B ≠ 0, this function is activated, and the dog feeding time is the parameter value of H200C-1B. If the integrated motor does not receive the dog feeding command within the set time, the motor will report a communication failure 0x7501.

Note: The communication failure code 0x7501 must be set as an unallowable self-recovery failure, such as 0x1007501.

For example: Suppose the CAN-ID of the motor on the CAN bus is 1. Enable the watchdog function and set the watchdog time of the integrated motor to 100ms, then turn off the watchdog function.

Serial number	Function	Message	Note
1	Modify the watchdog time of the target motor by 100ms	0601 2B 0C 20 1B 64 00 00 00 00	Effective immediately
2	Repeat feeding the dog	0601 2B 0C 20 1B 64 00 00 00 00	No alarm
3	Turn off the watchdog function	0601 2B 0C 20 1B 00 00 00 00 00	
4	Save parameters	0601 23 10 10 01 73 61 76 65	
5	Restart takes effect		The watchdog function is turned off.

8 Adjust

8.1 Summary

Servo gain must be adjusted reasonably according to the characteristics of the driving load,in order to maximize the performance of the integrated low-voltage servo motor and make the equipment respond quickly and accurately to the internal instructions or instructions from the master computer. Servo gain includes velocity loop gain, position loop gain, velocity loop integration time, filtering, etc,which is affected by each other.Therefore the balance between each other must be considered when adjusting servo gain parameters.

Note: before servo gain adjustment, inching test first is recommended to perform to confirm that the motor can operate normally; during the adjustment process, the vibration of the motor may be accompany, please pay full attention to safety.

8.2 Manual Gain Adjustment

8.2.1 Objects Related

Table 8-1

Object Index	Description
2008_h:01_h (SpeedLoopGain)	Velocity loop gain (unit: 0.1Hz), Increasing this parameter without causing noise or vibration can accelerate the positioning , with better velocity stability and following; when there is noise arising, reduce the parameter value.
2008_h:02_h (SpdLoopIntergrTime)	Velocity-loop integration time constant(unit:0.01ms), reducing this value can strengthen the integral effect and accelerate positioning. but if the value is set too small,mechanical vibration will be raised.
2008_h:03_h (PosLoopGain)	Gain of position-loop,increasing it can accelerate positioning and improve the anti-disturbance capacity during the motor operating stably.But setting too large can cause system instability and oscillation.
2008_h:$0F_h$ (SpdFdFwrFilterTime)	Speed feedforward filter time constant (unit:0.01ms)
2008_h:10_h (SpeedFdFwrGain)	Speed feedforward gain
2008_h:11_h (TorqFdFwrFilterTime)	Torque feedforward filter time constant (unit:0.01ms)
2008_h:12_h (TorqFdFwrGain)	Torque feedforward gain
2007_h:06_h (TorqRefFilterTime)	Torque command filter time constant,which can eliminate high frequency noise and mechanical resonance. The vibration raised by 2008_h:01_h can be restrained by adjusting 2007_h: 06_h. If the value is set too large, the response of the current-loop will be reduced.

8.2.2 Basic Gain Adjustment Instructions

The integrated low-voltage servo motor is composed of three control loops, which are current

loop, speed loop and position loop. The current loop has the highest response frequency. The factory-default current loop gain of servo have ensured sufficient responsiveness, and generally do not need to be adjusted, so only the position loop gain, speed loop gain and speed loop integration time constant need to be adjusted. In order to ensure the system stability,the position loop gain and the speed loop gain should be increased simultaneously.

The basic gain adjustment is as follows:

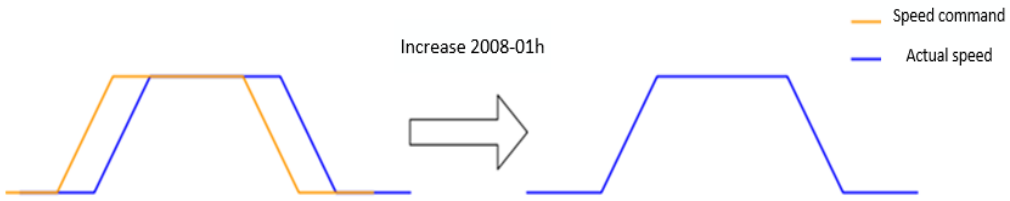


Figure 8-1 Speed loop gain

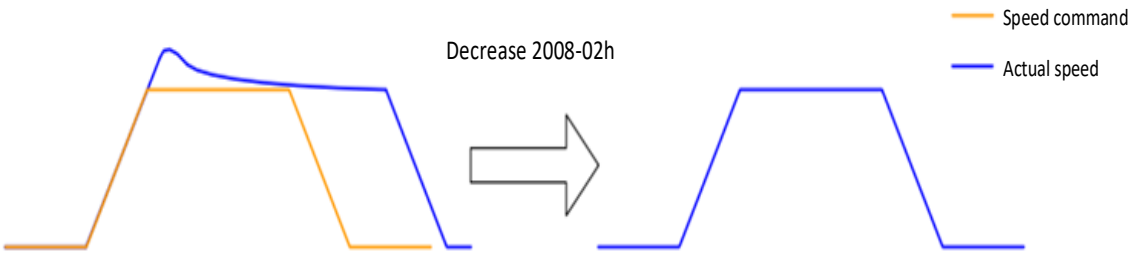


Figure 8-2 Integration time constant of speed loop

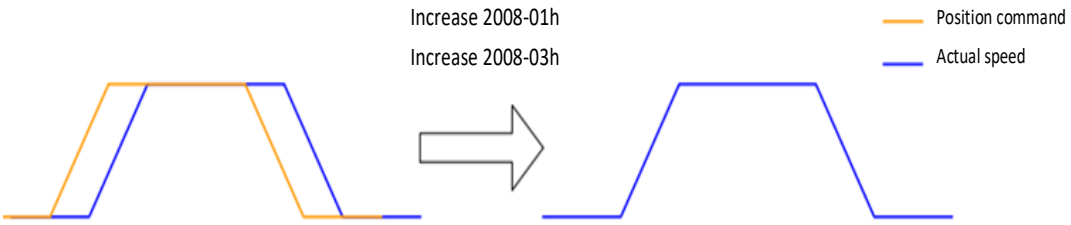


Figure 8-3 Position loop gain

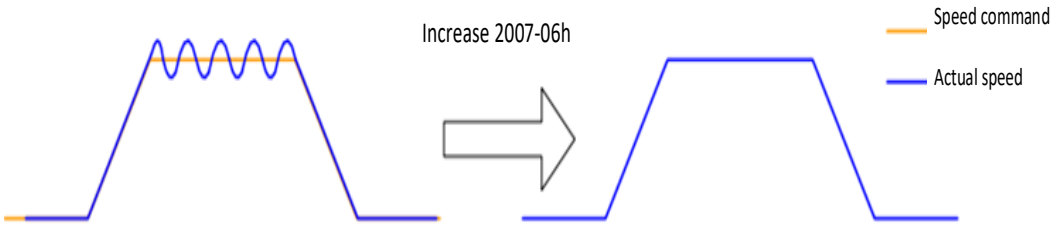


Figure 8-4 Torque instruction filter time constant

8.2.3 Parameter Adjustment Based on Rigid Table

The parameter adjustment based on the rigidity table refers to setting the PID parameters

corresponding to different rigidity grades through the rigidity table provided by the manufacturer (as shown in the table below), in order to achieve the purpose of quickly adjusting the stability of the integrated motor.

Table 8-2The parameters updated in the rigid table parameter adjustment mode

Rigidity	2008-03$_h$ Position gain(Hz)	2008-01$_h$SpeedLoop Gain(0.1Hz)	2008-02$_h$Velocity Integral Time Constant(0.01ms)
1	2	15	37000
2	3	20	28000
3	3	25	22000
4	4	30	19000
5	5	35	16000
6	6	45	12000
7	8	60	9000
8	10	75	7000
9	12	90	6000
10	14	110	5000
11	18	140	4000
12	32	180	3100
13	39	220	2500
14	48	270	2100
15	63	350	1600
16	72	400	1400
17	90	500	1200
18	108	600	1100
19	135	750	900
20	162	900	800
21	206	1150	700
22	251	1400	600
23	305	1700	500
24	377	2100	400
25	449	2500	400
26	500	2800	350
27	560	3100	300
28	610	3400	300
29	660	3700	250
30	720	4000	250
31	810	4500	200
32	900	5000	200

Under no-load conditions, it is recommended that the PID parameters be set to the parameters corresponding to rigidity level 12 or 13.

When parameter adjustment is required, it is recommended to prioritize PID parameters with a low rigidity grade and then gradually increase the rigidity grade until the usage requirements are met.

When the expected effect is not achieved by setting the parameters in the rigid table, the PID parameters can be fine-tuned. Optimize the effect through more meticulous adjustments.

Note: Before applying the rigid table for parameter adjustment, please confirm with the technical support personnel whether the purchased integrated motor supports this function.

8.2.4 Feedforward Gain Adjustment Instructions

Speed Feedforward

Velocity feedforward can improve the responsiveness of velocity command and reduce the position deviation when the velocity is fixed. It is mainly used in the full-closed-loop position control mode. The basic adjustment method is as follows:

Set 2008_h:13_h to a fixed value; then, adjust the value in 2008_h:14_h to increase gradually from 0,until a certain value, speed feedforward effects. During the adjustment, adjust 2008_h:13_h and 2008_h:14_h repeatedly to get good balance.

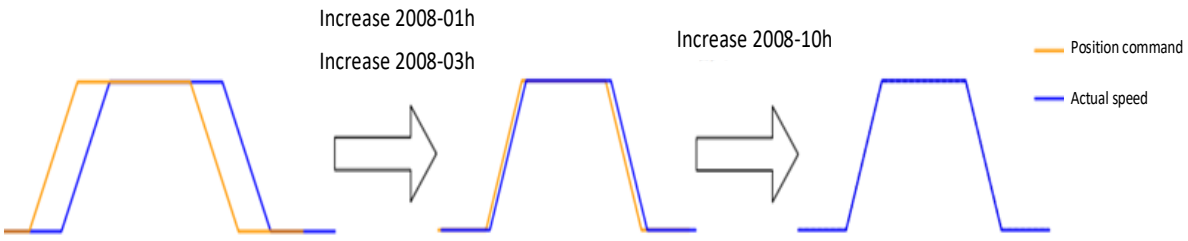


Figure 8-5 Speed feed-forward

Torque Feedforward

Position control mode, using torque feedforward, can improve the responsiveness of torque command and reduce the position deviation when fixed acceleration and deceleration; speed control mode, using torque feedforward, can improve the responsiveness of torque command and reduce the speed deviation at fixed speed. The basic adjustment method is as follows:

Set 2008_h:15_h as a fixed value; then set the value of 2008_h:16_h to increase gradually from 0, until a certain value,speed feedforward effect. During the adjustment, adjust 2008_h:15_h and 2008_h: 16_h repeatedly to find the setting with good balance.

8.2.5 Description of Gain Switching Adjustment

The gain switching function of the servo motor is triggered by the external DI or internal state. When the switching conditions are met, the servo motor switches to the first group or the second group of gains according to the effectiveness of the relevant parameter settings, to meet the actual use requirements. The specific functions are as follows:

- The state of motor switches from the running state to a higher gain to obtain better command tracking performance;
- The state of motor switches from the static state to a higher gain to shorten the positioning time;
- The state of motor switches from the static state (servo is enabled) to a lower gain to suppress vibration;
- The state of motor switches to different gain settings through external signals according to load conditions, etc;

Flow Chart of Gain Switching

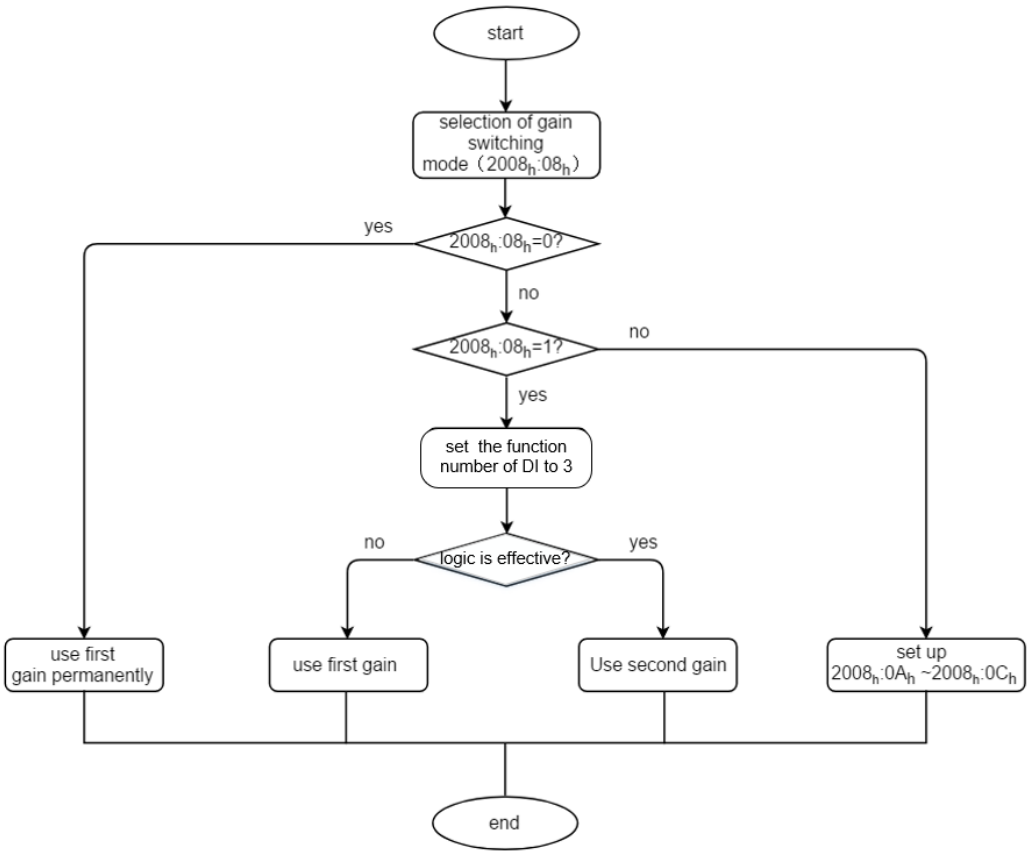


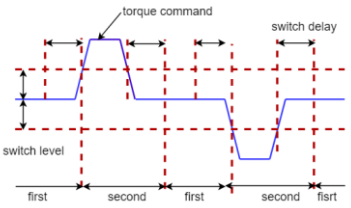
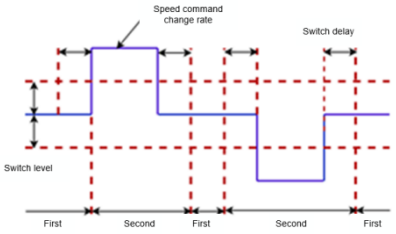
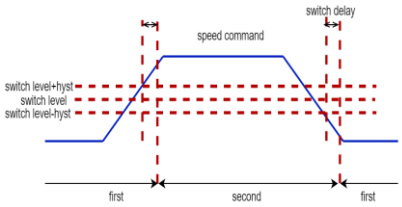
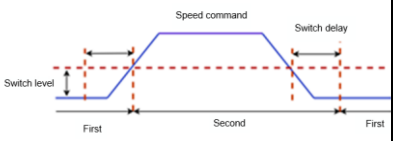
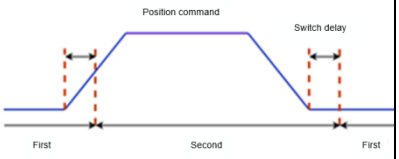
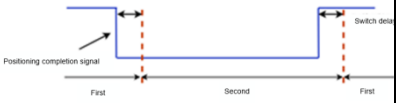
Figure 8-6

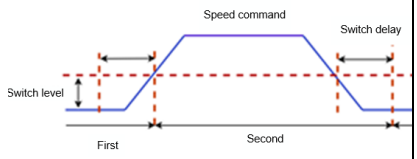
Conditions of Gain Switching

There are 10 modes for switching the first gain (2008_h: 01_h ~ 2008_h: 03_h) and the second gain (2008_h: 04_h ~ 2008_h: 06_h). The schematic diagram and associated parameters of different modes are shown in the table below:

Table 8-1

Gain Switching Mode			Correlation Parameters		
2008_h :08_h	Mode name	Sketch map	Delay time (2008_h:$0A_h$)	Switching level (2008_h:$0B_h$)	Switching delay (2008_h:$0C_h$)
0	Fixed use of the first set of	-	Ineffective	Ineffective	Ineffective

Gain Switching Mode			Correlation Parameters		
	gains				
1	Use external DI to switch	-	Ineffective	Ineffective	Ineffective
2	Torque instruction		Effective (ms)	Effective (%)	Effective (%)
3	Speed instruction	The internal effective speed instruction value ($606B_h$) is maintained for 25ms,the first set of gains is used fixedly, and the second set of gains is used when it is changed;	Effective (ms)	Ineffective	Ineffective
4	The variation ratio of speed instruction		有效 (ms)	有效 (rpm/ms)	有效 (rpm/ms)
5	Speed instruction		Effective (ms)	Effective (rpm)	Effective (rpm)
6	Position deviation		Effective (ms)	Effective (encoder unit)	Effective (encoder unit)
7	Position instruction		Effective (ms)	Ineffective	Ineffective
8	Positioning is completed		Effective (ms)	Ineffective	Ineffective
9	Actual speed		Effective	Effective	Effective

Gain Switching Mode			Correlation Parameters		
			(ms)	(rpm)	(rpm)

Note: the delay time (2008_h: $0A_h$) is effective when the first gain is switched to the second gain or the second gain is switched to the first gain.

8.3 Resonance Suppression

8.3.1 Mechanical resonance suppression

Due to the inherent resonance characteristics of the mechanical system, as the basic servo gain is increased,the servo motor may resonates at the mechanical resonance frequency, which may affect the servo performance. The notch filter can suppress resonance by reducing the gain at a specific frequency.

The integrated low-voltage servo motor has 4 groups of manual notch filters, each group of which has three parameters, namely, the frequency, band, width and depth.

Each parameter is set manually by the user, as shown in the table below:

Table 8-2

Name	Manual notch filters			
	1	2	3	4
Frequency(Hz)	2009_h:$0D_h$	2009_h:10_h	2009_h:13_h	2009_h:16_h
Notch band(Hz)	2009_h:$0E_h$	2009_h:11_h	2009_h:14_h	2009_h:17_h
Notch depth(%)	2009_h:$0F_h$	2009_h:12_h	2009_h:15_h	2009_h:18_h

Note: The notch filter is invalid when the depth level is 0.

When using the manual notch filter, it is necessary to set the frequency of the notch filter to the actual central resonance frequency, and input the width level (central resonance frequency range) and depth level of the notch filter group. If the resonance is suppressed, the notch filter will effect.

Notch Depth

When the depth level of the notch is 100, the input is completely suppressed at the center frequency; when the depth level of the notch is 0, the notch is invalid. Therefore, the larger the depth level of the notch, the deeper the notch depth, the stronger the suppression of mechanical resonance, but it may cause system instability, so it should be pay attention to when adjustment.

The principle of notch filter suppression is shown in the figure below:

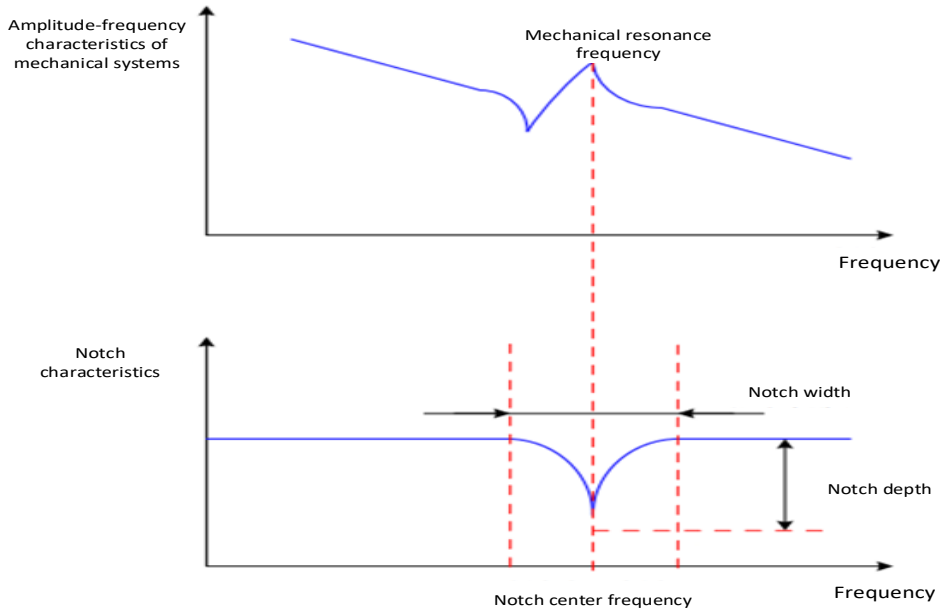


Figure 8-7

8.3.2 Frequency vibration suppression

Due to the inherent characteristics of the dual inertial system during position control, end vibrations are prone to occur during an emergency stop, which affects the positioning effect. The frequency of such vibrations is generally within 100 Hz. The low-frequency vibration suppression function can reduce the vibrations here.

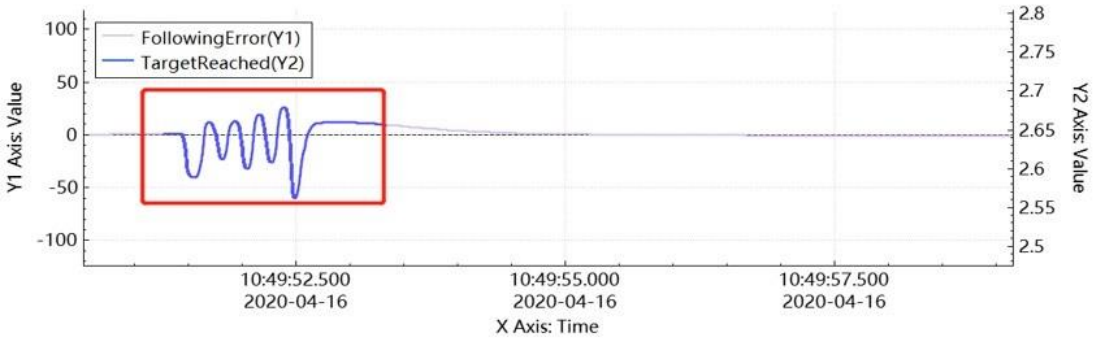
Related objects

Table 8-3

Object index	Description
2009_h:05_h (LowFreqResonantDepressMode)	Low frequency resonance suppression mode selection, 0-off, 1-on
2009_h:$1D_h$ (LowFreqResonantFreq)	Low frequency vibration frequency (unit: 0.1Hz), set the low frequency vibration suppression center frequency, the limit range is 1 ~ 1000
2009_h:$1E_h$ (LowFreqResonantFreqFilterSet)	Vibration suppression width level (unit%), set the effective frequency range of vibration suppression, 0 means only suppress the vibration at the center frequency; increase the setting value, you can increase the frequency range of low frequency vibration suppression, but it will lead to longer positioning time ; If the setting value is too small, the vibration suppression (such as belt load) cannot be completed when the load vibration frequency will change, and the setting can be debugged according to the actual situation. Limit range 0 ~ 100
2009_h:23_h (Reserve0)	Damping ratio coefficient, set the system vibration damping ratio coefficient, this value defaults to 5, generally not modified, the limit range is 0 ~ 50

Set low frequency vibration suppression parameters

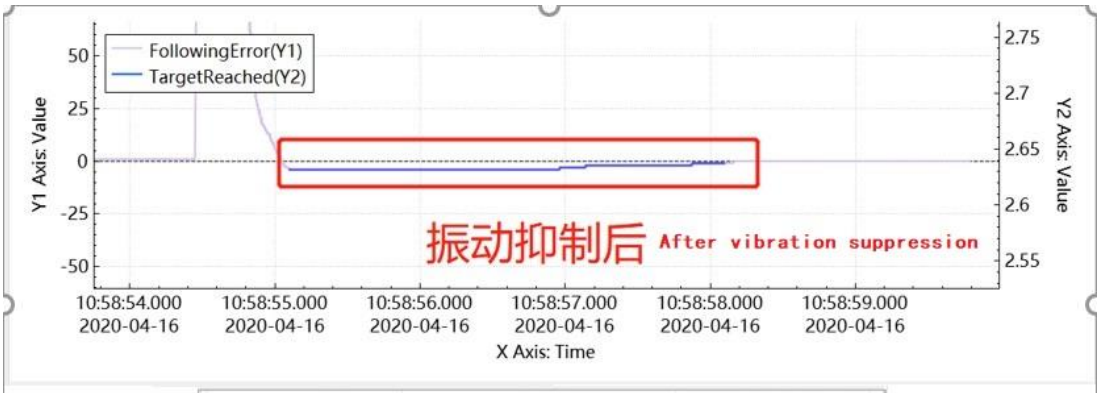
- Turn on the low-frequency vibration suppression mode, set 1 at 2009h: 05h;
- Use NiMStudio host computer software to collect the waveform of the motor in the position positioning state, and automatically calculate the frequency of low-frequency resonance after the movement stops;



Index	Parameter name	Equipment value
H2009-05	Low frequency resonance suppression mode selection	1
H2009-1D	Low frequency resonance frequency	40
H2009-1E	Vibration suppression width level	0

Figure 8-8

- Set the low-frequency vibration frequency and vibration suppression width level according to the low-frequency vibration frequency recognized by the NiMStudio host computer software;
- Observe the effect of motor positioning after using the low frequency vibration suppression function. If it is within the acceptable range, it means that the low frequency vibration suppression is effective; if it is not within the acceptable range, repeat the above two steps or manually set the low frequency vibration frequency and vibration suppression width grade.



Index	Parameter name	Equipment value
H2009-05	Low frequency resonance suppression mode selection	1
H2009-1D	Low frequency resonance frequency	40
H2009-1E	Vibration suppression width level	0

Figure 8-9

9 Error Management

9.1 Alarm Codes

When the integrated servo has a warning or fault, the servo will actively alarm (indicator light - yellow light flashes, flashing frequency being 2Hz indicates warning, 4Hz indicates fault), and execute corresponding actions. The corresponding alarm code will be stored in object 1003_h: 02_h ~ 1003_h: 10_h, and the current fault number be stored in object 1003_h: 00_h.

There are 16 alarm queues in object 1003_h, which are stored in stack mode and follow the first in, first out rule. If the historical alarm storage is occupied, a new alarm will be generated, and the earliest errors be deleted, the previous errors move downward in turn. The specific alarm codes are as follows:

Table 9-1

Alarm codes	Content of Alarms	Alarm Type	Response Code	Self Reset Allowed	(Error Code)
0x2300	Over current	Fault	0	No	0x2301
0x4012311	Over load	Warning	4	Yes	0x2311
0x3002312	Lock motor	Halt and Lock Motor	3	No	0x2312
0x2321	Phase-A upper bridge-arm short circuit *	Fault	0	No	0x2321
0x2322	Phase-A lower-bridge short circuit *	Fault	0	No	0x2322
0x2323	Phase-B upper-bridge short circuit *	Fault	0	No	0x2323
0x2324	Phase-B lower-bridge short circuit *	Fault	0	No	0x2324
0x2325	Phase-C upper-bridge short circuit *	Fault	0	No	0x2325
0x2326	Phase-C lower-bridge short circuit *	Fault	0	No	0x2326
0x13210	Power over-voltage	Fault	0	Yes	0x3210
0x1013220	Power under-voltage	Fault	1	Yes	0x3220
0x1003221	Power supply failure	Fault	1	No	0x3221
0x13211	Over pressure relief value fault *	Fault	1	Yes	0x3211
0x14210	Over temperature alarm	Fault	0	Yes	0x4210
0x14220	Under temperature alarm	Fault	0	Yes	0x4220
0x5080	Drive fault *	Fault	0	No	0x5080
0x5081	Phase-A upper-bridge Gate fault *	Fault	0	No	0x5081
0x5082	Phase-A lower-bridge	Fault	0	No	0x5082

Alarm codes	Content of Alarms	Alarm Type	Response Code	Self Reset Allowed	(Error Code)
	Gate fault *				
0x5083	Phase-B upper-bridge Gate fault *	Fault	0	No	0x5083
0x5084	Phase-B lower-bridge Gate fault *	Fault	0	No	0x5084
0x5085	Phase-C upper-bridge Gate fault *	Fault	0	No	0x5085
0x5086	Phase-C lower-bridge Gate fault*	Fault	0	No	0x5086
0x5540	Flash fault*	Fault	0	No	0x5540
0x5541	Flash initialization failure *	Fault	0	No	0x5541
0x4005542	Flash validation error warning *	Warning	4	No	0x5542
0x5543	Flash user area without parameters *	Fault	0	No	0x5543
0x4005544	Power down position storage abnormal	Warning	4	No	0x5544
0x4005545	Data saved after power failure is not stored	Warning	4	No	0x5545
0x6000	Hardware initialization fault *	Fault	0	No	0x6000
0x16320	Parameter setting fault	Fault	0	Yes	0x6320
0x6321	Registration failure*	Fault	0	No	0x6321
0x7301	Low-voltage fault of the encoder battery	Fault	0	No	0x7301
0x4017302	Low-voltage alarm of the encoder battery	Warning	4	Yes	0x7302
0x7303	Encoder communication fault	Fault	0	No	0x7303
0x1017304	Encoder multi-turn counting error	Fault	1	Yes	0x7304
0x7305	Z pulse failure *	Fault	0	No	0x7305
0x7306	Encoder failure *	Fault	0	No	0x7306
0x4017307	Encoder warning *	Warning	4	Yes	0x7307
0x1007309	Encoder battery failure fault	Fault	1	No	0x7309
0x17310	Over velocity	Fault	0	Yes	0x7310
0x17501	Communication fault	Fault	0	Yes	0x7501
0x4017502	CAN passive error	Warning	4	Yes	0x7502
0x4017503	CAN receiving overflow	Warning	4	Yes	0x7503
0x8610	Homing timeout	Fault	0	No	0x8610
0x8611	Tracking fault	Fault	0	No	0x8611

Alarm codes	Content of Alarms	Alarm Type	Response Code	Self Reset Allowed	(Error Code)
0x3018613	Software over limit fault	Halt and Lock Motor	3	No	0x8613
0x3018614	Overrun switch	Halt and Lock Motor	3	No	0x8614
0x8700	Tracking fault *	Fault	0	No	0x8700
0x3285	Phase failure	Fault	0	No	0x3285
0x4018615	Curve planning calculation fault	Warning	4	Yes	0x8615
0x18616	Target position overflow	Fault	0	No	0x8616
0x5018617	Curve planning parameter too small	Ignore	5	Yes	0x8617
0XFF01	Motor parameter identification failure *	Fault	0	No	0xFF01
0x401FF02	Unexecutable commands *	Warning	4	No	0xFF02
0xFF0C	Encoder calibration timeout	Fault	0	No	0xFF0C

Note：

- Alarm contents with * symbol cannot be configured by object fault code parameter group 200Eh. The integrated servo motor can only operate normally after the drive and hardware initialization failure requires restarting.
- $603F_h$ contains the last error of the integrated servo motor, corresponding to the fault code in the lower 16 digits of the alarm code.

9.2 Fault action settings

The alarm code is composed of a fault code (16 digits), a fault response code (8 digits), and a self-reset permission (8 digits). The fault code indicates the failure phenomenon of the servo motor under abnormal conditions. The current status of the servo motor is determined by the fault code, which generally corresponds to the object 603Fh (error code). 0 indicates that the servo motor is currently free of faults. The fault response code indicates that the servo motor is faulty. After the corresponding action mode and the treatment of status word fault bits, such as free stop-fault status, stop at a certain deceleration-fault status, suspended shaft lock-non-fault status, etc .; self-reset allows the servo motor's fault to be eliminated, Whether to allow the servo motor to automatically reset the fault bit of the current status word, 1-Allowed, 0-Disallowed. The order of the corresponding bits is as follows:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Failure response code [8]								Self-reset allowed [8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Fault code [16]															

Specific definition of fault response code

0- The state machine is switched to the fault state (refer to “4.1 CiA402 State Machine”). If 605E> = 0, the fault shutdown mode selected by 605E is adopted; otherwise: the power tube is fully closed and the machine stops freely.
1- The state machine is switched to the fault state (refer to “4.1 CiA402 State Machine”). If 605E> = 0, then the fault stop mode selected by 605E; otherwise: stop according to the slow curve to turn off the output.
2- The state machine is switched to the fault state (refer to “4.1 CiA402 State Machine”). If 605E> = 0, then the fault stop mode selected by 605E; otherwise: stop and close the output according to the fast curve.
3- Press the quick curve to suspend the locked axis without switching to the fault state and not affected by the object $605E_h$.
4- Warning and do not change the current operating state, do not switch to the fault state and are not affected by the object $605E_h$.
5- Ignore and record in 1003h. Not switching to fault state is not affected by object 605Eh.
6- Ignored but not recorded in 1003h, does not switch to fault state and is not affected by object $605E_h$.

Alarm code configuration

The fault code group $200E_h$ exemplifies the alarm codes that can be freely configured by the user, as follows:

Table 9-2

Fault code group $200E_h$	Meaning	Default fault action
0x4012311	Motor overload	No
0x3002312	Motor stalled	No
0x13210	Power overvoltage	Yes
0x1013220	Power undervoltage	Yes
0x14210	High temperature alarm	Yes
0x14220	Low temperature alarm	Yes
0x16320	Parameter setting error	Yes
0x17310	Speeding	Yes
0x17501	Communication failure	Yes
0x8610	Origin return timeout	Yes
0x8611	Out of position	Yes
0x3018613	Software limit error	No
0x3018614	Limit switch error	No
0x3285	Output phase failure	Yes
0x4018615	Curve planning calculation error	No
0x18616	Target position overflow	Yes
0x5018617	Curve planning parameters are too small	No

Note: "Yes (No)" of the fault action in the above list indicates whether the motor enters the fault state after this fault occurs.

Example of alarm code configuration

Power undervoltage setting

$200E_h$: 04_h (power undervoltage) is set to 0x1003220. When the servo motor is running, the power supply voltage is lower than the detection threshold (default 24V), the servo motor stops according to the stop mode set by 605E (refer to the corresponding bit sequence, the fault response code Correspondence is 1); when the power supply voltage returns to normal, the motor can perform normal operation through manual reset operation (refer to the corresponding bit sequence, self-reset permission corresponds to 0).

9.3 Fault Reset

After the recovery of the integrated servo fault, the fault code that is not allowed by the self-reset needs to be manually reset operation, the motor can be operated normally. There are 3 ways of manual reset operation, as follows.:

- Clear the fault status by sending control 0x80 (control word bit7 rising edge is valid)
- Through the entity input terminal (2003_h) one of the functions configured as function code No. 2 (alarm reset), and set the corresponding logic selection, according to the logic selection of the set terminal to make it valid to reset.
- One of the functions of the virtual input terminal (2017_h) is configured as function code No. 2 (alarm reset), and the corresponding logic selection is set, and the logic selection of the terminal is set to make it valid to reset.

9.4 Fault detection instructions

- Drive fault detection

  The fault is triggered by the interrupt response of the error signal from the drive chip, and the drive fault will be reported if it is triggered.

- I^2t Overload and Locked rotor error detection

  The time integration of I^2 will be calculated as the motor is operating above the rated current.When the integral value is greater than the product of the set maximum current (2000_h: $0A_h$) and the maximum current duration (2000_h:$0B_h$), it will trigger Overload or Locked rotor (the velocity is lower than 5 revolutions/min)error.If Locked rotor error occurs, the output of servo will be forbade; if Overload warning occurs, the torque output of servo will be limited to rated value, and the limit will be released after 30s.

- Over temperature and Under temperature fault detection

  When the temperature of the servo drive is higher than 90 , Over temperature error will be triggered,which is removed when the temperature is lower than 70 ; when the temperature of the servo drive is lower than - 25 , Under temperature error will be triggered,which is removed when the temperature is higher than - 20 .

- Over voltage and Under voltage error detection

When the voltage of servo power-supply is higher than the set voltage (2001_h:11_h), Over voltage error is triggered; when the voltage of servo power-supply is lower than the set voltage(2001_h:10_h),Under voltage error is triggered.

- Parameter setting error detection

  Parameter setting error is triggered when the multi-function terminal setting is wrong, including different terminals (physical terminal and virtual terminal) with the same function number or the 14th, 15th and 31th function numbers being not set in cia402 Homing mode.

- Over-velocity error detection

  When the velocity of the servo motor is higher than the velocity set in($200A_h$: 06_h) and maintaining for 10ms, the Over-velocity error will be triggered, and the error will be removed when the velocity is lower than the set speed.

- Communication error detection

  This error is identified by the slave protocol-error flag-bit in EtherCAT communication.

- Hardware initialization error detection

  When the servo is powered on, if the hardware (drive, encoder) is not initialized successfully, the hardware initialization error will be triggered.

- Flash error detection

  When the flash of servo is operated, flash error will be triggered if the parameters storing is fail or the parameters reading is failed in power-on storage.

- Homing timeout error detection

  When Homing mode in cia402 is enabled, if it fails to complete the Homing action within the limited time (2005_h:$1C_h$), it will trigger the Homing timeout error.

- Tracking error detection

  When the position deviation of servo motor exceeds 6065h and maintaining for 6066_h (ms), the Tracking error will be triggered in position mode.

- Software over limit error detection

  In the position control mode, this error is judged according to the software limit value (user unit) set in $607B_h$ and $607D_h$. If the actual position exceeds the set value, it will trigger the Software over limit error.

- Overrun switch error detection

  According to the 14th and 15th function numbers set by the multi-function terminal, check whether the positive and negative limit switches are triggered, and the overrun switch error occurs when it is triggered.

- Phase failure error detection

  When the integrated servo operates with one or more phase is lack, the phase failure error will be triggered.

- Target position overflow error detection

  In the position control mode, if the product of the given position ($607A_h$) and the gear ratio

(6091h) exceeds the data range of int32 (- 2147483648 ~ + 2147483647) , the target position overflow error will be triggered.

- Curve planning parameter too small error detection

  In the position control mode, if the S-shaped curve can't be operated in the given displacement range under the given shock or acceleration, the curve planning parameters too small error will be triggered.

- Encoder low voltage fault detection

  When the servo motor is powered on, the encoder battery voltage is detected to be ≤2.4V, and a fault 0x1007301 is reported.

- Encoder low voltage alarm detection

  When the servo motor is powered on, the encoder battery voltage is detected to be between 2.4V and 2.6V, and a alarm 0x4017302 is reported.

- Encoder communication fault detection

  The servo motor fails to communicate with the encoder and reports a 0x7303 fault.

- Encoder multi-turn counting error detection

  If the encoder is not calibrated or the calibration fails, a 0x7304 fault is reported.

- Encoder battery failure fault

  ①During power off, if no battery is connected, it will report a 0x7309 fault.
  ②The charged hot-swappable IO battery extension cable will report a 0x7309 fault.
  When this fault is reported, connect the battery, restart the power supply of the servo motor, and reset the position to zero.

- Power supply failure fault

  The motor was only connected to the logic power supply but not the power supply for operation. It reported a fault of 0x1003221.

10 Object Dictionary

Object dictionary is an important part of device specification. It is an ordered set of parameters and variables, including all parameters of device description and device network status. Each object includes index, sub-index, name, description, data type, data range, accessibility, mappability,factory-setting, unit and activate method.

Noun interpretation

Index: Specifies the position of the same type of object in the object dictionary, expressed in hexadecimal
Sub index: ,the offset of each object in this class , there are multiple objects under the same index
"Data type and range": refer to the following table for details please

Table 10-1

Data type	Data range	Data length
Int8	-128~127	1 bit
Int16	-32768~32767	2 bits
Int32	-2147483648~2147483647	4 bits
Uint8	0~255	1 bit
Uint16	0~65535	2 bits
Uint32	0~4294967295	4 bits
String	ASCII	~

"Accessibility": please refer to the following table for details

Table 10-2

Accessibility	Description
RW	Readable and writable
WO	just writable
RO	Just Readable

Mappability: please refer to the following table for details

Table 10-3

Mappability	Description
NO	Not mappable in PDO
RPDO	Can be used as RPDO
TPDO	Can be used as TPDO

"Activate method": please refer to the following table for details

Table 10-4

activate method	Description
Immediate activate	After parameter setting, the set value takes effect immediately
Stop activate	After the parameter setting, the set value takes effect until the drive is in stop state
Restart activate	After the parameter setting , the set value takes effect until the drive is restart

10.1 Communication Parameter Group 1000h Description

Table 10-5

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
1000h	00h	Device type	Device type	uint32	-	RO	NO	-	VAR	-
1001h	00h	Error Register	Error Register	Uint8	-	RO	NO	0	VAR	-
1003h	00h	Pre-defined error field	Number of alarms in the current device	uint8	-	RO	NO	-	VAR	-
	01h	s01_NumberErrors	Historical alarm cache	uint32	-	RO	NO	-	VAR	-
	02h	s02_ErrorField		uint32	-	RO	NO	-	VAR	-
	03h	s03_ErrorField		uint32	-	RO	NO	-	VAR	-
	04h	s04_ErrorField		uint32	-	RO	NO	-	VAR	-
	05h	s05_ErrorField		uint32	-	RO	NO	-	VAR	-
	06h	s06_ErrorField		uint32	-	RO	NO	-	VAR	-
	07h	s07_ErrorField		uint32	-	RO	NO	-	VAR	-
	08h	s08_ErrorField		uint32	-	RO	NO	-	VAR	-
	09h	s09_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Ah	s10_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Bh	s11_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Ch	s12_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Dh	s13_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Eh	s14_ErrorField		uint32	-	RO	NO	-	VAR	-
	0Fh	s15_ErrorField		uint32	-	RO	NO	-	VAR	-
	10h	s16_ErrorField		uint32	-	RO	NO	-	VAR	-
1008h	00h	Device name	Device name	String	-	RO	NO	-	-	-
1009h	00h	Hardware verison	Hardware verison	String	-	RO	NO	-	-	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
100A$_h$	00$_h$	Software version		String	-	RO	NO	1.1	-	-
1010$_h$	-	Store parameter	Save parameters	String	-	RO	NO	-	ARRAY	-
	0A$_h$	SaveNimFunParam	Save factory version information parameters	uint32	-	RW	NO	1	-	-
	0B$_h$	SaveNimOriParam	Save all factory parameters	uint32	-	RW	NO	1	-	-
1011$_h$	-	Restore default parameter	Restore default parameters	String	-	RO	NO	-	ARRAY	-
	01$_h$	RestAllParam	Read all parameters	uint32	-	RW	NO	1	VAR -	-
	02$_h$	RestCommParam	Read 1000 user parameters	uint32	-	RW	NO	1	-	-
	03$_h$	RestAppParam	Read 6000 user parameters	uint32	-	RW	NO	1	-	-
	0B$_h$	RestNimOriParam	Read factory parameters	uint32	-	RW	NO	1	-	-
1017h	-	Producer heartbeat time	Producer heartbeat time	Uint16	-	RW	NO	0	ms	-
1018$_h$	-	Identity Object	Marking object	-	-	RO	NO	-	REC	-
	00$_h$	Number of Entries	Number of entries	Uint8	-	RO	NO	-	VAR	-
	01$_h$	Vendor-ID	Manufacturer ID number	uint32	-	RO	NO	0x2D	-	-
	02$_h$	Product code	Product code	uint32	-	RO	NO	0x01	-	-
	03$_h$	APP Version	Software version number	uint32	-	RO	NO	-	-	-
	04$_h$	Serial number	Product serial number	uint32	-	RO	NO	-	-	-
1400$_h$	-	Receive PDO 1 Communication Parameter	RPDO1 parameters	-	-	RO	NO	-	REC	-
	00$_h$	Number of Entries	Number of entries	Uint8	-	R0	NO	-	VAR	-
	01$_h$	COB-ID	COB-ID	uint32	-	RW	NO	0x200 +Node_ID	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	02_h	Transmission Type	Transmission type	uint32	-	RW	NO	0xFF	VAR	-
1401_h	-	Receive PDO 1 Communication Parameter	RPDO2 parameters	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of entries	Uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	COB-ID	uint32	-	RW	NO	0x300 +Node_ID	VAR	-
	02_h	Transmission Type	Transmission type	uint32	-	RW	NO	0xFF	VAR	-
1402_h	-	Receive PDO 1 Communication Parameter	RPDO3 parameters	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of entries	Uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	COB-ID Transmission type	uint32	-	RW	NO	0x400 +Node_ID	VAR	-
	02_h	Transmission Type	RPDO4 parameters	uint32	-	RW	NO	0xFF	VAR	-
1403_h	-	Receive PDO 1 Communication Parameter	Number of entries	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	COB-ID	Uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	Transmission type RPDO3 parameters	uint32	-	RW	NO	0x500 +Node_ID	VAR	-
	02_h	Transmission Type	Number of entries	uint32	-	RW	NO	0xFF	VAR	-
	-	Drive Control	Drive Control	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
1600h	01h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60400010	VAR	-
	02h	2st object to be mapped	2st object to be mapped	uint32	-	RW	NO	0x60600008	VAR	-
1601h	-	Positioning Control	Positioning Control	-	-	RO	NO	-	REC	-
	00h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x607A0020	VAR	-
	02h	2st object to be mapped	2st object to be mapped	uint32	-	RW	NO	0x60C10120	VAR	-
1602h	-	Velocity Control	Velocity Control	-	-	RO	NO	-	REC	-
	00h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60420010	VAR	-
	02h	2st object to be mapped	2st object to be mapped	uint32	-	RW	NO	0x60FF0020	VAR	-
1603h	-	Torque Control	Torque Control	-	-	RO	NO	-	REC	-
	00h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60710010	VAR	-
1800h	-	Transmit PDO 1 Communication Parameter	TPDO1 parameters	-	-	RO	NO	-	REC	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	00_h	Number of Entries	Number of Entries	Uint8	-	RO	NO	-	VAR	-
	01_h	COB-ID	COB-ID	uint32	-	RW	NO	0x180+Node_ID	VAR	-
	02_h	Transmission Type	Transmission Type	Uint8	-	RW	NO	0xFF	VAR	-
	03_h	Inhibit Time	Inhibit Time	Uint16	-	RW	NO	0x0	VAR	-
	04_h	Compatibility Entry	Compatibility Entry	Uint8	-	RW	NO	-	VAR	-
	05_h	Event Timer	Event Timer	Uint16	-	RW	NO	0x0	VAR	-
1801_h	-	Transmit PDO 2 Communication Parameter	TPDO2 parameters	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RO	NO	-	VAR	-
	01_h	COB-ID	COB-ID	uint32	-	RW	NO	0x280+Node_ID	VAR	-
	02_h	Transmission Type	Transmission Type	Uint8	-	RW	NO	0xFF	VAR	-
	03_h	Inhibit Time	Inhibit Time	Uint16	-	RW	NO	0x0	VAR	-
	04_h	Compatibility Entry	Compatibility Entry	Uint8	-	RW	NO	-	VAR	-
	05_h	Event Timer	Event Timer	Uint16	-	RW	NO	0x0	VAR	-
1802_h	-	Transmit PDO 3 Communication Parameter	TPDO3 parameters	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RO	NO	-	VAR	-
	01_h	COB-ID	COB-ID	uint32	-	RW	NO	0x380+Node_ID	VAR	-
	02_h	Transmission Type	Transmission Type	Uint8	-	RW	NO	0xFF	VAR	-
	03_h	Inhibit Time	Inhibit Time	Uint16	-	RW	NO	0x0	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	04_h	Compatibility Entry	Compatibility Entry	Uint8	-	RW	NO	-	VAR	-
	05_h	Event Timer	Event Timer	Uint16	-	RW	NO	0x0	VAR	-
1803_h	-	Transmit PDO 4 Communication Parameter	TPDO4 parameters	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RO	NO	-	VAR	-
	01_h	COB-ID	COB-ID	uint32	-	RW	NO	0x480+Node_ID	VAR	-
	02_h	Transmission Type	Transmission Type	Uint8	-	RW	NO	0xFF	VAR	-
	03_h	Inhibit Time	Inhibit Time	Uint16	-	RW	NO	0x0	VAR	-
	04_h	Compatibility Entry	Compatibility Entry	Uint8	-	RW	NO	-	VAR	-
	05_h	Event Timer	Event Timer	Uint16	-	RW	NO	0x0	VAR	-
$1A00_h$	-	Drive Status	Drive Status	Uint8	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60410010	VAR	-
	02_h	2st object to be mapped	2st object to be mapped	uint32	-	RW	NO	0x60610008	VAR	-
$1A01_h$	-	Positioning Status	Positioning Status	Uint8	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60630020	VAR	-
	02_h	2st object to be mapped	2st object to be mapped	uint32	-	RW	NO	0x60640020	VAR	-
$1A02_h$	-	Velocity Status	Velocity Status	Uint8	-	RO	NO	-	REC	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	00_h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x606C0020	VAR	-
$1A03_h$	-	Torque Status	Torque Status	Uint8	-	RO	NO	-	REC	-
	00_h	Number of Entries	Number of Entries	Uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	1st object to be mapped	uint32	-	RW	NO	0x60770010	VAR	-

10.2 Manufacturer Defined Parameter Group 2000_h Description

Table 10-6

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2000_h (Servo Motor Parameter)	06_h	SerialEncoderType	Bus encoder type	uint16	0~1	RW	NO	1	-	Stop activate
	07_h	RatedVoltage	Rated Voltage	uint16	24~48	RW	NO	48	1V	Stop activate
	08_h	RatedPower	Power Rating	uint16	-	RW	NO	40	0.01KW	Stop activate
	09_h	RatedCurrent	Rated Current	uint16	-	RW	NO	1250	0.01A	Stop activate
	$0A_h$	MaximumCurrent	Maximum current	uint16	-	RW	NO	2250	0.01A	Stop activate
	$0B_h$	MaxCurrentDurTime	Maximum current duration	uint16	-	RW	NO	3000	0.1s	Stop activate
	$0C_h$	RatedTorque	Rated Torque	uint16	-	RW	NO	127	0.01Nm	Stop activate
	$0D_h$	MaxTorque	Maximum torque	uint16	-	RW	NO	2000	0.01Nm	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2000h（Servo Motor Parameter）	$0E_h$	RatedMotorSpeed	Rated speed	uint16	-	RW	NO	3000	1rpm	Stop activate
	$0F_h$	MaxMotorSpeed	Maximum speed	uint16	-	RW	NO	6000	1rpm	Stop activate
	10_h	RotorInertiaJm	Rotational inertia Jm	uint16	-	RW	NO	30	0.01kg cm^2	Stop activate
	11_h	PolePairsNum	Number of pole pairs of permanent magnet synchronous motor	uint16	-	RW	NO	4	1	Stop activate
	12_h	StatorResistance	Stator resistance	uint16	-	RW	NO	115	0.001 Ω	Stop activate
	13_h	StatorInductanceLq	Stator inductance Lq	uint16	-	RW	NO	25	0.01mH	Stop activate
	14_h	StatorInductanceLd	Stator inductance Ld	uint16	-	RW	NO	25	0.01mH	Stop activate
	16_h	TorqueCoeffKt	Torque coefficient Kt	uint16	-	RW	NO	11	0.01Nm/Arms	Stop activate
	19_h	EncoderBitPR	Number of absolute bits of encoder	uint32	-	RW	NO	14	1bit	Stop activate
	$1A_h$	EncoderPoleNum	Number of encoder poles	uint16	-	RW	NO	1	1	Stop activate
	$1C_h$	SensorSelect	Position sensor selection	uint16	-	RW	NO	2	-	Stop activate
	$1D_h$	SignalZ_ElecAngle	Z signal corresponds to electric angle	uint16	-	RW	NO	0	1/4096	Stop activate
	$1E_h$	P_hU_UpEdgeElecAngle	U-phase rising edge corresponds to counter	uint16	-	RW	NO	0	1/4096	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			angle							
2001_h(Drive Parameters)	04_h	VoltageClass	Voltage Class	uint16	0~65535	RW	NO	0	1	Stop activate
		Specific description: When the setting value is less than 50, it corresponds to the voltage calculation coefficient of 200/400W servo motor; when the setting value is greater than 50, it corresponds to the voltage calculation coefficient of 750W servo motor								
	07_h	RatedOutputCurrent	Rated output current	Uint16	0~65535	RO	NO	1200	0.01A	Stop activate
	08_h	MaxOutputCurrent	Maximum Output Current	uint16	0~65535	RO	NO	1500	0.01A	Stop activate
	$0D_h$	DeadZoneTime	Switching dead time	uint16	0~65535	RW	NO	100	us	Stop activate
	$0E_h$	BusOverVoltVal	DC busbar overvoltage protection point	uint16	0~65535	RW	NO	100	1V	Stop activate
	$0F_h$	BusDisc$_h$argeVoltVal	DC bus voltage drain point	uint16	0~65535	RW	NO	55	1V	Stop activate
	10_h	BusLowVoltVal	DC bus voltage undervoltage point	uint16	0~65535	RW	NO	24	1V	Stop activate
	11_h	OverCurrentVal	Driver overcurrent protection point	uint16	0~65535	RW	NO	10	1V	Stop activate
	18_h	CurrentFilterCoeff	Current sampling filter parameters	uint16	0~65535	RW	NO	100	0.01	Stop activate
	$1C_h$	CurLoopCutOffFrq	Current loop cut-off frequency	uint16	0~65535	RW	NO	800	1Hz	Stop activate
	$1D_h$	OpenLoopCurrent	Open-loop operation current	uint16	0~65535	RW	NO	625	0.01A	Stop activate
	01_h	CtrlModeSelec	Control mode selection	uint16	0~4	RW	NO	4	-	Immediate activate
		Specific description: 0~CIA402 mode, 1~NiMotion position mode, 2~NiMotion velocity mode, 3~NiMotion torque mode, 4~NiMotion open-loop mode, 5~offline identification of motor parameters								

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2002_h(Basic Control Parameters)	04_h	OutputPulsePhase	Duty cycle of output pulse	uint16	0~100	RW	NO	50	1%	Operation setting Stop activate
	$0A_h$	BrkOffToStopMotorOffDelay	Hold brake output off to motor non-energy delay time	unint16	0~65535	RW	NO	1	1ms	Shutdown Settings Effective immediately
	13_h	BreakResistorType	Internal and external inhibit resistor selection, see section 7.5 for details	uint16	0~2	RW	NO	0	1	Restart activate
	$1D_h$	MCUid0	MCU identification code 0	uint32	-	RO	NO	0	-	-
	$1E_h$	MCUid1	MCU identification code 1	uint32	-	RO	NO	0	-	-
	$1F_h$	MCUid2	MCU identification code 2	uint32	-	RO	NO	0	-	-
2003_h（Entity Terminal Input Parameters）	03_h	DI1FunSelec	DI1 terminal logic selection	uint16	0~45	RW	NO	1	1	Stop activate
		Description:See Section7.1 for details								
	04_h	DI1LogicSelec	DI1 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
		Description:See Section7.1 for details								
	05_h	DI2FunSelec	DI2 terminal function selection	uint16	0~45	RW	NO	2	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2003h（Entity Terminal Input Parameters）	06h	DI2LogicSelec	DI2 terminal logic selection	uint16	0~4	RW	NO	2	1	Stop activate
	07h	DI3FunSelec	DI3 terminal function selection	uint16	0~45	RW	NO	12	1	Stop activate
	08h	DI3LogicSelec	DI3 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	09h	DI4FunSelec	DI4 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	0Ah	DI4LogicSelec	DI4 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	0Bh	DI5FunSelec	DI5 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	0Ch	DI5LogicSelec	DI5 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	0Dh	DI6FunSelec	DI6 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	0Eh	DI6LogicSelec	DI6 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	0Fh	DI7FunSelec	DI7 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	10h	DI7LogicSelec	DI7 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	11h	DI8FunSelec	DI8 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	12h	DI8LogicSelec	DI8 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2003h（Entity Terminal Input Parameters）	13h	DI9FunSelec	DI9 terminal function selection	uint16	0~45	RW	NO	0	1	Stop activate
	14h	DI9LogicSelec	DI9 terminal logic selection	uint16	0~4	RW	NO	0	1	Stop activate
	15h	DinFunAllocation3	Power-on effective function assignment	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: bit0-DI1，bit1-DI2，bit2-DI3;the corresponding bit is 1, indicating that the corresponding DI Port is high-level 3.0V								
	17h	AI1Offset	AI1 Offset	int16	-32768~32767	RW	NO	0	1mV	Stop activate
	18h	AI1FilterTimeConst	AI1 input filter time constant	uint16	0~65535	RW	NO	0	-	Stop activate
	19h	AI1DeadZone	AI1 Deadband	uint16	0~65535	RW	NO	0	1mV	Stop activate
	1Ah	AI1ZeroDrift	AI1 multiplier	int16	-32768~32767	RW	NO	0	0.001	Stop activate
	1Fh	SpeedValAt10V	Analog 10V corresponding to speed value	uint16	0~4000	RW	NO	0	rpm	Stop activate
	20h	TorqueValAt10V	Analog 10V corresponds to torque value	uint16	0~2000	RW	NO	0	0.1%	Stop activate
2004h（Terminal Output Parameters）	01h	DO1FunSelec	DO1 Terminal Function Selection	uint16	0~30	RW	NO	0	1	Stop activate
		Description:See Section7.1 for details								
	02h	DO1LogicSelec	DO1 terminal logic selection	uint16	0~1	RW	NO	0	1	Stop activate
		Description:See Section7.1 for details								
	03h	DO2FunSelec	DO2 terminal function selection	uint16	0~30	RW	NO	0	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2004h（Terminal Output Parameters）	04h	DO2LogicSelec	DO2 terminal logic selection	uint16	0~1	RW	NO	0	1	Stop activate
	05h	DO3FunSelec	DO3 terminal function selection	uint16	0~30	RW	NO	0	1	Stop activate
	06h	DO3LogicSelec	DO3 terminal logic selection	uint16	0~1	RW	NO	0	1	Stop activate
	07h	DO4FunSelec	DO4 terminal function selection	uint16	0~30	RW	NO	0	1	Stop activate
	08h	DO4LogicSelec	DO4 terminal logic selection	uint16	0~1	RW	NO	0	1	Stop activate
	09h	DO5FunSelec	DO5 terminal function selection	uint16	0~30	RW	NO	0	1	Stop activate
	0Ah	DO5LogicSelec	DO5 terminal logic selection	uint16	0~1	RW	NO	0	1	Stop activate
2005h（Position Control Parameters）	01h	PosRefSource	Position instruction source	uint16	0~65535	RW	NO	1	1	Stop activate
		Description:0-pluse，1-step amount								
	05h	StepAmount	Step Amount	int16	-32768~ 32767	RW	NO	100	inc	Stop activate
	18h	HomingMode	Homing Mode	uint16	0~1	RW	NO	0	-	Stop activate
		Specific description: 0~multi-turn position recovery, 1~single-turn position recovery								
	1C	HomingDurationLimit	Go home time limit	uint16	0~65535	RW	NO	10000	ms	Stop activate
	1Dh	HomeMechOffset	Maximum forward rotation position	Int32	0~10000	RW	NO	0		Stop activate
	2Ah	PosingWindowTime	AI1 deadband judgment time window	uint16	0~65535	RW	NO	0	1ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	$2C_h$	EncoderFreqDivPulsOput	Analog 10V to position value	uint32	0~4294967295	RW	NO	0	inc	Stop activate
2006_h (Velocity Control Parameters)	01_h	MainSpeedRefSourceA	Primary speed command A source	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0 ~ digital setting (setting value in 2006_h: 04_h), 3 ~ duty ratio setting								
	02_h	AuxSpeedRefSourceB	Auxiliary speed command B source	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0 ~ digital setting (setting value in 2006_h: 04_h)								
	03_h	SpeedRefSourceSelec	Speed instruction selection	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0-A (only 2006_h: 01_h effective), 1-B (only 2006_h: 02_h effective), 2-A + B (both 2006_h: 01_h and 2006_h: 02_h effective)								
	04_h	KeypadSpeedRef	Speed command keypad setpoint	int16	-32768~32767	RW	NO	10	1rpm	Immediate activate
	06_h	ValidSpeedRef	Effective Speed Setpoint	int16	-32768~32767	RW	NO	0	1rpm	Immediate activate
	07_h	SpeedRefAccelRampTime	Speed command acceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	Immediate activate
	08_h	SpeedRefDecelRampTime	Speed command deceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	Immediate activate
	09_h	MaxSpeedLimit	Maximum Speed Threshold	uint16	0~65535	RW	NO	6000	1rpm	Immediate activate
	$0A_h$	FwrSpeedLimit	Forward speed threshold	uint16	0~65535	RW	NO	4000	1rpm	Immediate activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	$0B_h$	BwrSpeedLimit	Reverse speed threshold	uint16	0~65535	RW	NO	4000	1rpm	Immediate activate
2007_h (Torque Control Parameters)	01_h	MainTorqRefSourceA	Source of main torque command A	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0 ~ digital setting (setting value in 2007_h: 04_h)								
	02_h	AuxTorqRefSourceB	Source of auxiliary torque command B	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0 ~ digital setting (setting value in 2007_h: 04_h)								
	03_h	TorqRefSource	Torque command selection	uint16	0~65535	RW	NO	0	1	Stop activate
		Description: 0-A (only 2007_h: 01_h effective), 1-B (only 2007_h: 02_h effective), 2-A + B (both 2007_h: 01_h and 2007_h: 02_h effective)								
	04_h	KeypadTorqRef	Torque command keyboard set value	int16	-32768~32767	RW	NO	10	0.1%	Stop activate
	05_h	ValidTorqRef	Effective torque command set value	int16	-32768~32767	RW	NO	10	0.1%	Stop activate
	06_h	TorqRefFilterTime	Torque command filter time constant	uint16	0~65535	RW	NO	0	0.01ms	Immediate activate
		Description: low-pass filtering of torque command can make the torque command smoother and reduce vibration. If the setting value is too large, the responsiveness will be reduced.								
	$0A_h$	FwrTorqInterLimit	Positive internal torque limit	uint16	0~65535	RW	NO	1000	0.1%	Immediate activate
	$0B_h$	BwrTorqInterLimit	Anti-internal torque limit	uint16	0~65535	RW	NO	1000	0.1%	Immediate activate
	10_h	TorqCtrlFwrSpeedLimit	Torque control forward speed limit value	uint16	0~65535	RW	NO	3000	1rpm	Immediate activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	11_h	TorqCtrlBwrSpeedLimit	Torque control reverse speed limit value	uint16	0~65535	RW	NO	3000	1rpm	Immediate activate
2008_h (Gain Parameter)	01_h	SpeedLoopGain	Velocity loop gain	uint16	1~20000	RW	NO	500	0.1Hz	Immediate activate
		Description: set the proportional gain of the speed loop. This parameter determines the response of the speed loop. The larger the setting value is, the faster the response is. However, setting it too large may cause vibration. In the position mode, if you want to increase the position loop gain, you need to increase the speed loop gain at the same time.								
	02_h	SpdLoopIntegrTime	Integral time constant of velocity loop	uint16	0~51200	RW	NO	800	0.01ms	Immediate activate
		Description: set the integral time constant of the speed loop. The smaller the setting value is, the stronger the integration effect is, and the deviation is closer to 0 at the stopping time. If this parameter is set to 51200, there is no integration effect.								
	03_h	PosLoopGain	Position loop gain	uint16	0~20000	RW	NO	1500	1	Immediate activate
		Description: set the proportional gain of the position loop. This parameter determines the response of the position loop. Setting a larger position loop gain can shorten the positioning time, but setting too large may cause vibration.								
	04_h	SpeedLoop2ndGain	Second speed loop gain	uint16	1~20000	RW	NO	1	0.1Hz	Immediate activate
	05_h	SpdLoop2ndIntegrTime	Integral time constant of the second velocity loop	uint16	0~51200	RW	NO	15	0.01ms	Immediate activate
	06_h	PosLoop2ndGain	Second position loop gain	uint16	0~20000	RW	NO	0	1	Immediate activate
	08_h	GainSwitchModeSet	Gain switching mode selection	uint16	0~1	RW	NO	0	1	Immediate activate
		Description: description:								

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2008h (Gain Parameter)		0 ~ fixed use of the first set of gains; 1 ~ use external DI to switch. If DI is ineffective, it is the first set of gains; if DI is effective, it is the second set of gains; 2 ~ torque instruction, when the absolute value of torque instruction exceeds (level + delay) [%] and lasts for a period of delay time, it switches to the second set of gains; when the absolute value of torque instruction is less than (level- delay) [%] and lasts for a period of delay time, it returns to the first set of gains; 3 ~ the internal effective speed instruction value (606Bh) is maintained for 25ms, the first set of gains is used fixedly, and the second set of gains is used when it is changed; 4 ~ the variation ratio of speed instruction. When the absolute value of the variation ratio of speed instruction exceeds (level + delay) [rpm / ms] and the state lasts during the delay time, it switches to the second set of gains; when the absolute value of the variation ratio of speed instruction is less than (level-delay)[rpm / ms] and the state lasts during the delay time, it returns to the first set of gains; 5 ~ speed instruction, when the absolute value of speed instruction exceeds (level + delay) [rpm] and lasts for a period of delay time, it switches to the second set of gains; when the absolute value of speed instruction is less than (level-delay) [rpm] and lasts for a period of delay time, it returns to the first set of gains; 6 ~ position deviation, when the absolute value of position deviation exceeds (level + delay) [encoder unit] and lasts during the delay time, it switches to the second set of gains; when the absolute value of position deviation is less than (level-delay) [encoder unit] and lasts during the delay time, it returns to the first set of gains; 7 ~ position instruction, when the state of position deviation instruction is not 0 and lasts in the delay time,it switches to the second set of gains;when the state of position instruction is 0 and lasts in the delay time, it returns to the first set of gains; 8 ~ after the positioning is completed, if the positioning is not completed and lasts in the delay time, it switches to the secondset of gains; if the positioning is completed and lasts in the delay time, it returns to the first set of gains; except for the position control mode, the rest are fixed as the second set of gains; 9 ~ actual speed. When the absolute value of the actual speed exceeds (level + delay) [rpm] and lasts during the delay time,it switches to the second set of gains; when the absolute value of the actual speed is less than (level-delay) [rpm] and lasts during the delay time, it returns to the first set of gains.								
	09h	GainSwitchCondition	Effective gain group number	uint16	0~10	RW	NO	0	1	Immediate activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2008_h (Gain Parameter)		Description: 0 ~ the first group of gains, 1 ~ the second switching conditions								
	$0A_h$	GainSwitch$_h$Delay	Gain switching delay time	uint16	0~10000	RW	NO	0	0.1ms	Immediate activate
	$0B_h$	GainSwitch$_h$Level	Gain switching level	uint16	0~20000	RW	NO	0	1	Immediate activate
		Description: The actual switching action is affected by the two conditions of level and time delay. See the description of 2008_h: 08_h for the specific influence mode. Depending on the gain switching mode, the unit of switching level will change accordingly.								
	$0C_h$	GainSwitch$_h$Hyst	Gain switching delay	uint16	0~20000	RW	NO	0	1	Immediate activate
		Description: The actual switching action is affected by the two conditions of level and time delay. See the description of 2008_h: 08_h for the specific influence mode. Depending on the gain switching mode, the unit of switching level will change accordingly.								
	$0F_h$	SpdFdFwrFilterTime	Speed feedforward filtering time constant	uint16	0~65535	RW	NO	0	0.01ms	Immediate activate
	10_h	SpeedFdFwrGain	Speed feedforward gain	uint16	0~65535	RW	NO	0	0.1%	Immediate activate
		Description: in the position control mode, increasing this parameter can improve the position command response and reduce the position deviation at a fixed speed. During the adjustment, set 2008_h:$0F_h$ as a fixed value, and gradually increase the setting value in 2008_h:10_h from 0 to a certain value, until the speed feedforward effect is achieved. During the adjustment, adjust 2008_h:$0F_h$ and 2008_h:10_h repeatedly to get the value with good balance								
	11_h	TorqFdFwrFilterTime	Torque feedforward filtering time constant	uint16	0~65535	RW	NO	0	0.01ms	Immediate activate
	12_h	TorqFdFwrGain	Torque feedforward gain	uint16	0~65535	RW	NO	0	0.001	Immediate activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2008_h (Gain Parameter)		Description: in non torque mode, increasing this parameter can improve the responsiveness to the changing speed command, improve the position command response, and reduce the position deviation at fixed speed. During the adjustment, set 2008_h:11_h as a fixed value, and gradually increase the setting value in 2008_h:12_h to increase the torque feedforward effect; when speed overshoot occurs, keep 2008_h: 12_h unchanged, and increase 2008_h: 11_h. During the adjustment, adjust 2008_h:11_h and 2008_h:12_h repeatedly to get the value with good balance								
	14_h	SpdFdBckFilterCutOff Freq	Cut-off frequency of velocity feedback filter	uint16	0~4000	RW	NO	900	1Hz	Immediate activate
		Description: the smaller the setting is, the smaller the speed feedback fluctuation is, but the greater the feedback delay is. Generally, the default parameters are maintained.								
2009_h (Gain Parameter)	05_h	LowFreqResonantDe pressMode	Low frequency vibration suppression mode selection	uint16	0~1	RW	NO	0	1	Stop activate
	06_h	OfflInertiaAutoTunMo de	Parameter off-line identification status display	uint16	0~65535	RO	NO	0	1	-
		Specific description: see section 8.2.2 for details								
	$0D_h$	Notch1Freq		uint16	0~2000	RW	NO	0	1Hz	Immediate activate
		Description: set the center frequency of the notch filter, i.e. mechanical resonance frequency.								
	$0E_h$	Notch1Width 调整 Level		uint16	0~2000	RW	NO	0	1Hz	Immediate activate
		Description: set the frequency band width on both sides of the mechanical resonance center frequency								
	$0F_h$	Notch1DepthLevel		uint16	0~100	RW	NO	0	%	Immediate activate
		Description: 0 ~ no notch, 100 ~ maximum notch depth								

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2009h (Gain Parameter)	10h	Notch2Freq	Notch 2 Frequency	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	11h	Notch2WidthLevel	Notch 2 Width Level	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	12h	Notch2DepthLevel	Notch2 Depth Level	uint16	0~100	RW	NO	0	1	Immediate activate
	13h	Notch3Freq	Notch 3 Frequency	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	14h	Notch3WidthLevel	Notch 3 Width Level	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	15h	Notch3DepthLevel	Notch3 Depth Level	uint16	0~100	RW	NO	0	1	Immediate activate
	16h	Notch4Freq	Notch 4 Frequency	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	17h	Notch4WidthLevel	Notch 4 Width Level	uint16	0~2000	RW	NO	0	1Hz	Immediate activate
	18h	Notch4DepthLevel	Notch 4 Depth Level	uint16	0~100	RW	NO	0	1	Immediate activate
	1Dh	LowFreqResonantFreq	Low frequency resonance frequency	uint16	10~1000	RW	NO	10	0.1Hz	Stop activate
		Specific description: See section 8.4.2 for details								
	1Eh	LowFreqResonantFreqFilterSet	Vibration suppression width level	uint16	0~100	RW	NO	0	1	Stop activate
		Description: See section 8.4.2 for details								
	23h	Reserve0	Damping ratio coefficient	uint16	0~10	RW	NO	5	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
		Description: See section 8.4.2 for details								
200Ah（Fault and Protection Parameters）	06_h	OverSpeedThrh	Overspeed fault threshold	uint16	0~65535	RW	NO	0	1	Immediate activate
	$1B_h$	Encoder multi-turn overflow fault selection	Encoder multi-turn overflow fault selection	uint16	0~1	RW	NO	0	1	Immediate activate
	$1F_h$	BrakeDetectionEn	Lock detection enable	uint16	0~65535	RW	NO	0	1	Stop activate
	20_h	BrakeLoadGravity	Brake load gravity	int16	0~2000	RW	NO	0	0.1%	Stop activate
200B$_h$ (Monitoring Parameters)	01_h	DriveState	Internal state of servo motor driver	uint16	0~12	RO	NO	0	1	-
		Description: 0 ~ not ready, 1 ~ ready, 6 ~ position closed-loop, 8 ~ speed closed-loop, 9 ~ torque control, 10 ~ open-loop control, 12 ~ error								
	02_h	ActualMotorSpeed	Actual motor speed	int16	-32768~32767	RO	NO	-	1rpm	-
	04_h	InterTorqueRef	Internal torque command	int16	-32768~32767	RO	NO	0	0	-
	05_h	MonitoredDiStates	Input signal (DI signal) monitoring	uint16	0~65535	RO	NO	0	1	-
	06_h	MonitoredDoStates	Output signal (DO signal) monitoring	uint16	0~65535	RO	NO	0	1	-
	$0A_h$	PosRefToSpeed	Input PWM frequency	uint16	0~65535	RO	NO	0	1Hz	-
	$0F_h$	TotalPowerOnTime	Total power-up time	uint32	0~4294967295	RO	NO	0	0.1s	-
	12_h	P_hCurrentEffectValA	RMS A-phase current	uint16	0~65535	RO	NO	0	0.01A	-
	13_h	P_hCurrentEffectValB	RMS value of phase B current	uint16	0~65535	RO	NO	0	0.01A	-
	14_h	P_hCurrentEffectValC	RMS value of phase C current	uint16	0~65535	RO	NO	0	0.01A	-
	15_h	BusVoltage	Bus voltage value	uint16	0~65535	RO	NO	0	0.1V	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	16_h	ModuleTemperature	Module temperature value	int16	-32768~32767	RO	NO	0	1°C	-
	$2D_h$	ActualMotorSpeed01	Actual motor speed	uint16	0~65535	RO	NO	-	0.1rpm	-
$200C_h$（Communication parameters）	01_h	CommunicationSelect	Communication mode selection	uint16	0~2	RW	NO	1	1	Restart activate
		Description: 0 ~ external pulse control (default CAN communication),1~EtherCAT，2~CAN								
	02_h	ServoShaftAddress	Drive axis address	uint16	1~247	RW	NO	1	1	Restart activate
		Description: set the drive shaft address when using CANopen communication. 1 ~ 247: when multiple servo motors are networked, each servo motor can only have a unique address, otherwise, the communication will be abnormal or unable to communicate.								
	03_h	SerialPortBaudRate	Serial ports baud-rate setting	uint16	0~8	RW	NO	8	1	Restart activate

<table>
<tr><th>Index</th><th>Sub-index</th><th>Name</th><th>Description</th><th>Data type</th><th>Data range</th><th>Access-ibility</th><th>Mapp-ability</th><th>Factory setting</th><th>Unit</th><th>Activate method</th></tr>
<tr><td></td><td></td><td colspan="9">Description: set the communication rate between the servo motor and the master computer, as shown in the table below:
<table>
<tr><th>Setting value</th><th>Baud-rate setting</th></tr>
<tr><td>8</td><td>1MBit/sec</td></tr>
<tr><td>7</td><td>800kBit/sec</td></tr>
<tr><td>6</td><td>500kBit/sec</td></tr>
<tr><td>5</td><td>250kBit/sec</td></tr>
<tr><td>4</td><td>125kBit/sec</td></tr>
<tr><td>3</td><td>100kBit/sec</td></tr>
<tr><td>2</td><td>50kBit/sec</td></tr>
<tr><td>1</td><td>20kBit/sec</td></tr>
<tr><td>0</td><td>10kBit/sec</td></tr>
</table>
The communication rate of the servo motor must be the same as that of the master computer, otherwise it cannot be communicated</td></tr>
<tr><td rowspan="2">200D$_h$ (auxiliary function parameter)</td><td>0A$_h$</td><td>Encoder reset enable</td><td>Encoder reset enable</td><td>uint16</td><td>0~2</td><td>RW</td><td>NO</td><td>-</td><td>1</td><td>Restart activate</td></tr>
<tr><td>0B$_h$</td><td>BrakeLoadTun</td><td>Load identification of holding brake</td><td>uint16</td><td>0~1</td><td>RW</td><td>NO</td><td>0</td><td>1</td><td>Stop activate</td></tr>
<tr><td rowspan="2">200E$_h$ （Errror Code)</td><td>01$_h$</td><td>MotorOverLoad</td><td>Motor overload</td><td>int32</td><td>-2147483648~2147483647</td><td>RW</td><td>NO</td><td>67183377</td><td>1</td><td>Immediate activate</td></tr>
<tr><td>02$_h$</td><td>MotorlockRotor</td><td>Motor blockage</td><td>int32</td><td>-2147483648~2147483647</td><td>RW</td><td>NO</td><td>8978</td><td>1</td><td>Immediate activate</td></tr>
</table>

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
200E$_h$ (Errror Code)	03$_h$	OverVoltage	Power supply over-voltage	int32	-2147483648~2147483647	RW	NO	78352	1	Immediate activate
	04$_h$	UnderVoltage	Power supply undervoltage	int32	-2147483648~2147483647	RW	NO	16855584	1	Immediate activate
	05$_h$	OverDriveTemp	Temperature too high	int32	-2147483648~2147483647	RW	NO	82448	1	Immediate activate
	06$_h$	UnderDriveTemp	Temperature too low	int32	-2147483648~2147483647	RW	NO	82464	1	Immediate activate
	07$_h$	ParameterError	Parameter setting error	int32	-2147483648~2147483647	RW	NO	90912	1	Immediate activate
	08$_h$	OverSpeed	Motor over speed	int32	-2147483648~2147483647	RW	NO	94992	1	Immediate activate
	09$_h$	CommError	Communication failure	int32	-2147483648~2147483647	RW	NO	95489	1	Immediate activate
	0A$_h$	HomingError	Home return timeout error	int32	-2147483648~2147483647	RW	NO	34320	1	Immediate activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
200E$_h$ (Errror Code)	0B$_h$	FollowingError	Position overrun error	int32	-2147483648~2147483647	RW	NO	34321	1	Immediate activate
	0C$_h$	PosSwlimitError	Software overrun fault	int32	-2147483648~2147483647	RW	NO	50431507	1	Immediate activate
	0D$_h$	PosGaugeError	Over-limit switch fault	int32	-2147483648~2147483647	RW	NO	50431507	1	Immediate activate
	OE$_h$	PhaseOutputError	Output phase fault	int32	-2147483648~2147483647	RW	NO	12933	1	Immediate activate
	0F$_h$	ProfileCalcFault	Curve planning calculation error	int32	-2147483648~2147483647	RW	NO	67208725	1	Immediate activate
	10$_h$	ProfilePosTargetOverflow	Target position overflow	int32	-2147483648~2147483647	RW	NO	99862	1	Immediate activate
	11$_h$	ProfileParaTooSmall	Curve planning parameter too small	int32	-2147483648~2147483647	RW	NO	83985943	1	Immediate activate
	01$_h$	MultiPosRunMode	Multi-segment position operation	uint16	0~3	RW	NO	0	1	Stop activate
		Specific description: see chapter 6.16.4 for details								
	02$_h$	PosRefProfileNum	Position command end segment number	uint16	1~16	RW	NO	1	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)		Detailed description: Set the total number of segments of the multi-segment position instruction								
	04h	MultiPosTimeUnit	Multi-segment position cycle times	uint16	0~65535	RW	NO	0	1	Stop activate
		Detailed description: The number of cycles of multi-segment position operation mode 1 and 3								
	05h	DisplaceRefType	Position command type selection	uint16	0~1	RW	NO	0	1	Stop activate
		Detailed description: 0-relative displacement; 1-absolute displacement								
	06h	StartPosSequRunSelec	Multi-segment position cycle starting segment	uint16	0~16	RW	NO	0	1	Stop activate
		Detailed description: Multi-segment position starting mode is 1 and 3								
	07h	Displace1	1st stage displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	08h	Displace1MaxSpeed	1st stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	09h	Displace1AccelDecelTime	1st stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
		Detailed description: Set the time for the first stage of the multi-stage position to change from 0rpm to 1000rpm.								
	0Ah	Displace1DoneWaitTime	Waiting time after completion of the first stage of displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
		Detailed description: After the first stage of the multi-stage position								
	0Bh	Displace2	2nd stage displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	0Ch	Displace2MaxSpeed	2nd stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	0Dh	Displace2AccelDecel Time	2nd stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	0Eh	Displace2DoneWaitTi me	Waiting time after completion of the second stage displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	0Fh	Displace3	3rd stage displacement	int32	-2147483648~2 147483647	RW	NO	- 1073741824	User unit	Stop activate
	10h	Displace3MaxSpeed	3rd stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	11h	Displace3AccelDecel Time	Stage 3 displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	12h	Displace3DoneWaitTi me	Waiting time after the third stage displacement is completed	uint16	0~65535	RW	NO	0	ms	Stop activate
	13h	Displace4	4th stage displacement	int32	-2147483648~2 147483647	RW	NO	- 1073741824	User unit	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	14h	Displace4MaxSpeed	4th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	15h	Displace4AccelDecel Time	4th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	16h	Displace4DoneWaitTi me	Waiting time after completion of the fourth stage displacement Step 5 displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	17h	Displace5	5th stage displacement maximum running speed	int32	-2147483648~2 147483647	RW	NO	- 1073741824	User unit	Stop activate
	18h	Displace5MaxSpeed	Stage 5 displacement acceleration / deceleration time	uint16	0~3000	RW	NO	1	rpm	Stop activate
	19h	Displace5AccelDecel Time	Stage 5 displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	1Ah	Displace5DoneWaitTi me	Waiting time after completion of the 5th stage displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	1Bh	Displace6	Step 6 displacement	int32	-2147483648~2 147483647	RW	NO	- 1073741824	User unit	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	$1C_h$	Displace6MaxSpeed	6th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	$1D_h$	Displace6AccelDecel Time	6th stage displacement maximum running speed	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1E_h$	Displace6DoneWaitTi me	Waiting time after completion of the 6th stage displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1F_h$	Displace7	Step 7 displacement	int32	-2147483648~2 147483647	RW	NO	-1073741824	User unit	Stop activate
	20_h	Displace7MaxSpeed	7th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	21_h	Displace7AccelDecel Time	7th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	22_h	Displace7DoneWaitTi me	Waiting time after completion of the 7th displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	23_h	Displace8	8th stage displacement	int32	-2147483648~2 147483647	RW	NO	-1073741824	User unit	Stop activate
	24_h	Displace8MaxSpeed	8th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	25_h	Displace8AccelDecelTime	8th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	26_h	Displace8DoneWaitTime	Waiting time after completion of the 8th stage displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	27_h	Displace9	Step 9 displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	28_h	Displace9MaxSpeed	9th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	29_h	Displace9AccelDecelTime	9th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2A_h$	Displace9DoneWaitTime	Waiting time after completion of the 9th displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2B_h$	Displace10	Step 10 displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	$2C_h$	Displace10MaxSpeed	10th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	$2D_h$	Displace10AccelDecelTime	10th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	$2E_h$	Displace10DoneWaitTime	Waiting time after completion of the 10th shift	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2F_h$	Displace11	Step 11 Displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	30_h	Displace11MaxSpeed	Maximum operating speed of the 11th displacement	uint16	0~3000	RW	NO	1	rpm	Stop activate
	31_h	Displace11AccelDecelTime	11th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	32_h	Displace11DoneWaitTime	Wait time after completion of the 11th shift	uint16	0~65535	RW	NO	0	ms	Stop activate
	33_h	Displace12	Step 12 displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	34_h	Displace12MaxSpeed	12th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	35_h	Displace12AccelDecelTime	12th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)	36_h	Displace12DoneWaitTime	Waiting time after the 12th displacement is completed	uint16	0~65535	RW	NO	0	ms	Stop activate
	37_h	Displace13	Step 13 displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	38_h	Displace13MaxSpeed	13th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	39_h	Displace13AccelDecelTime	13th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$3A_h$	Displace13DoneWaitTime	Waiting time after completion of the 13th displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	$3B_h$	Displace14	14th stage displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	$3C_h$	Displace14MaxSpeed	14th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	$3D_h$	Displace14AccelDecelTime	14th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$3E_h$	Displace14DoneWaitTime	Waiting time after the 14th stage	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (multi segment position function parameter)			displacement is completed							
	3F$_h$	Displace15	Step 15 Displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	40$_h$	Displace15MaxSpeed	15th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	41$_h$	Displace15AccelDecelTime	15th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	42$_h$	Displace15DoneWaitTime	Waiting time after completion of the 15th displacement	uint16	0~65535	RW	NO	0	ms	Stop activate
	43$_h$	Displace16	16th stage displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	User unit	Stop activate
	44$_h$	Displace16MaxSpeed	16th stage displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Stop activate
	45$_h$	Displace16AccelDecelTime	16th stage displacement acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	46$_h$	Displace16DoneWaitTime	Waiting time after the 16th stage	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			displacement is completed							
2012h (multi segment speed function parameter)	01_h	MultiSpeedRunMode	Multi-step speed instruction operation mode	uint16	0~2	RW	NO	0	1	Stop activate
		Specific description: see chapter 6.17.4 for details								
	02h	SpeedRefProfileNum	Speed command end segment selection	uint16	0~16	RW	NO	1	1	Stop activate
	03h	MultiSpeedTimeUnit	Multi-stage speed cycle times	uint16	0~65535	RW	NO	0	1	Stop activate
	$0C_h$	SpeedRef1	1st speed command	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$0D_h$	SpeedRef1RunTime	The first instruction run time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$0E_h$	SpeedRef1AccelDecelTime	1st stage acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$0F_h$	SpeedRef2	2nd instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	10_h	SpeedRef2RunTime	2nd instruction run time	uint16	0~65535	RW	NO	0	ms	Stop activate
	11_h	SpeedRef2AccelDecelTime	2nd stage acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	12_h	SpeedRef3	Paragraph 3	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	13_h	SpeedRef3RunTime	3rd instruction run time	uint16	0~65535	RW	NO	0	ms	Stop activate
	14_h	SpeedRef3AccelDecelTime	4th Stage acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	15_h	SpeedRef4	4th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	16_h	SpeedRef4RunTime	4th Section Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012h (multi segment speed function parameter)	17_h	SpeedRef4AccelDecelTime	4th acceleration / deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	18_h	SpeedRef5	5th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	19_h	SpeedRef5RunTime	5th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1A_h$	SpeedRef5AccelDecelTime	5th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1B_h$	SpeedRef6	6th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$1C_h$	SpeedRef6RunTime	6th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1D_h$	SpeedRef6AccelDecelTime	6th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$1E_h$	SpeedRef7	7th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$1F_h$	SpeedRef7RunTime	7th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	20_h	SpeedRef7AccelDecelTime	7th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	21_h	SpeedRef8	8th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	22_h	SpeedRef8RunTime	8th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	23_h	SpeedRef8AccelDecelTime	8th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012h (multi segment speed function parameter)	24_h	SpeedRef9	9th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	25_h	SpeedRef9RunTime	9th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	26_h	SpeedRef9AccelDecelTime	9th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	27_h	SpeedRef10	10th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	28_h	SpeedRef10RunTime	10th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	29_h	SpeedRef10AccelDecelTime	10th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2A_h$	SpeedRef11	11th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$2B_h$	SpeedRef11RunTime	11th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2C_h$	SpeedRef11AccelDecelTime	11th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2D_h$	SpeedRef12	12th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$2E_h$	SpeedRef12RunTime	12th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$2F_h$	SpeedRef12AccelDecelTime	12th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	30_h	SpeedRef13	13th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012h (multi segment speed function parameter)	31_h	SpeedRef13RunTime	13th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	32_h	SpeedRef13AccelDecelTime	13th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	33_h	SpeedRef14	14th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	34_h	SpeedRef14RunTime	14th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	35_h	SpeedRef14AccelDecelTime	14th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	36_h	SpeedRef15	15th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	37_h	SpeedRef15RunTime	15th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	38_h	SpeedRef15AccelDecelTime	15th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	39_h	SpeedRef16	16th Instruction	int16	0~3000	RW	NO	-6000	rpm	Stop activate
	$3A_h$	SpeedRef16RunTime	16th Instruction Run Time	uint16	0~65535	RW	NO	0	ms	Stop activate
	$3B_h$	SpeedRef16AccelDecelTime	16th Instruction acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Stop activate
	01_h	VDI1FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
		Description:See Section7.1 for details								
	02_h	VDI1LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2017_h（Virtual Input Terminal parameters）		Description:See Section7.1 for details								
	03_h	VDI2FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	04_h	VDI2LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	05_h	VDI3FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	06_h	VDI3LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	07_h	VDI4FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	08_h	VDI4LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	09_h	VDI5FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	$0A_h$	VDI5LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$0B_h$	VDI6FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	$0C_h$	VDI6LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$0D_h$	VDI7FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	$0E_h$	VDI7LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$0F_h$	VDI8FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	10_h	VDI8LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	11_h	VDI9FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	12_h	VDI9LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	13_h	VDI10FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	14_h	VDI10LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	15_h	VDI11FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	16_h	VDI11LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	17_h	VDI12FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	18_h	VDI12LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	19_h	VDI13FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	$1A_h$	VDI13LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$1B_h$	VDI14FunSelec		uint16	0~45	RW	NO	0	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	$1C_h$	VDI14LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$1D_h$	VDI15FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	$1E_h$	VDI15LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
	$1F_h$	VDI16FunSelec		uint16	0~45	RW	NO	0	1	Stop activate
	20_h	VDI16LogicSelec		uint16	0~1	RW	NO	0	1	Stop activate
2031_h（Communication Given Servo Related Variables）	01_h	VDI_VirtualLevelCommSet		uint16	0~65535	RW	NO	0	1	Stop activate
		Description: bit (n) = 1 in 2031_h:01_h means VDI (n + 1) is high-level; bit (n) = 0 means VDI (n + 1) is low-level								
	02_h	DoStateCommSet		uint16	0~65535	RW	NO	0	1	Stop activate
		description: when the logical selection in 2004_h is positive, if 2031_h: 02_h = 1, Do output high-level, if 2031_h: 02_h = 0, Do output low-level; when the logic selection in 2004_h is negative, if 2031_h: 02_h = 1, Do output low-level; if 2031_h: 02_h = 0, Do output high-level								

10.3 Sub-protocol Definition Parameter Group 6000_h Description

Table 10-7

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
$603F_h$	00_h	Error Code	See section 9.1 for details	uint16	0~65535	RO	TPDO	0	-	-
6040_h	00_h	Control word	Control word	uint16	0~65535	RW	RPDO	0	1	Stop activate
6041_h	00_h	Status word	Status word	uint16	0~65535	RO	TPDO	0	1	-
6042_h	00_h	VI target velocity	Target velocity in VM mode	int16	-3000~3000	RW	RPDO	0	1rpm	Stop activate
6043_h	00_h	VI velocity demand	Velocity demand in VM	int16	-3000~3000	R0	RPDO	0	1rpm	-
6044_h	00_h	vl velocity actual value	Actual speed in VM mode	int16	-32768~32767	RO	TPDO	0	1rpm	-
6046_h	01_h	VI velocity min amount	VM mode speed min.	uint32	0~3000	RW	RPDO	10	1rpm	Stop activate
	02_h	VI velocity max amount	VM mode speed maximum	uint32	0~3000	RW	RPDO	3000	1rpm	Stop activate
6048_h	01_h	delta speed	Acceleration in VM=delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Stop activate
	02_h	delta time		uint16	1~65535	RW	RPDO	1	1s	Stop activate
6049_h	01_h	delta speed	Deceleration in VM=delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Stop activate
	02_h	delta time		uint16	1~65535	RW	RPDO	1	1s	Stop activate
$604A_h$	01_h	delta speed	Braking acceleration in VM=delta speed/delta time	uint32	0~4294967295	RW	RPDO	800	1rpm	Stop activate
	02_h	delta time		uint16	1~65535	RW	RPDO	1	1s	Stop activate
$604C_h$	01_h	vl dimension factor numerator	Scaling factor in VM	int32	-2147483648~2147483647	RW	RPDO	0	1	Stop activate
	02_h	vl dimension factor denominator		int32	-2147483648~2147483647	RW	RPDO	0	1	Stop activate
	Sub index 1 contains the numerator (multiplier) used to calculate speed, and sub index 2 contains the denominator (divisor). Speed unit: rpm * sub index 1 / sub index 2. For example, if sub index 1 is set to 2 and sub index 2 is set to 1, then speed unit: 2rpm. If any of the sub indexes 1 and 2 is 0, the speed unit is rpm.									

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
605A$_h$	00$_h$	Quick stop option code	See section 4.1.4 for details	int16	0~2	RW	NO	2	1	Stop activate
605D$_h$	00$_h$	Halt option code	See section 4.1.4 for details	int16	1~2	RW	NO	1	1	Stop activate
605E$_h$	00$_h$	Fault reaction option code	See section 4.1.4 for details	int16	0~2	RO	NO	0	1	Stop activate
6060$_h$	00$_h$	ModeOperation	See section 6.1 for details	int8	0~10	RW	RPDO	1	1	Stop activate
6061$_h$	00$_h$	Modes of operation display	See section 6.2 for details	int8	0~10	RO	TPDO	1	1	-
6062$_h$	00$_h$	Position demand value	Position instruction value in effect inside the controller	int32	-2147483648~2147483647	RO	TPDO	0	User unit	-
6063$_h$	00$_h$	Position actual encoder value		int32	-2147483648~2147483647	RO	TPDO	0	编码器单位	-
6064$_h$	00$_h$	Position actual user value		int32	-2147483648~2147483647	RO	TPDO	0	User unit	-
		Description: position feedback 6064$_h$ × gear ratio (6091$_h$) = position feedback 6063$_h$								
6065$_h$	00$_h$	Following error Window		uint32	0~4294967295	RW	RPDO	50	User unit	Stop activate
		Description: when the position deviation exceeds ± 6065h and the duration time reaches 6066$_h$, Tracking error is triggered								
6066$_h$	00$_h$	Following error time out		uint16	0~65535	RW	RPDO	30000	1ms	Stop activate
		Description: when the position deviation exceeds ± 6065$_h$ and the duration time reaches 6066$_h$, Tracking error is triggered								
6067$_h$	00$_h$	Position Window		uint32	0~4294967295	RW	RPDO	10	User unit	Stop activate
		Description: when the position deviation exceeds ± 6067$_h$ and the duration reaches 6068$_h$, Target position is reached.Bit10 of status word 6041$_h$=1 in position mode								
6068$_h$	00$_h$	Position Window time		uint16	0~65535	RW	RPDO	5	1ms	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
		Description: when the position deviation exceeds ± 6067$_h$ and the duration reaches 6068$_h$, Target position is reached.Bit10 of status word 6041$_h$=1 in position mode								
6069$_h$	00$_h$	Velocity sensor actual value		int32	-2147483648~2147483647	RO	TPDO	0	User unit/s	-
606B$_h$	00$_h$	Velocity demand value	Velocity instruction value in effect inside the controller	int32	-2147483648~2147483647	RO	TPDO	0	User unit/s	-
606C$_h$	00$_h$	Velocity actual value	Current velocity actual value feedback	int32	-2147483648~2147483647	RO	TPDO	0	rpm	-
606D$_h$	00$_h$	Velocity Window		uint16	0~65535	RW	RPDO	500	rpm	Stop activate
		Description: when the difference between the target speed in 60FF$_h$ (converted into motor speed rpm) and the actual speed of the motor is within ± 606D, and the duration time reaches 606E$_h$, means that the speed is reached, and the bit10 in status word 6041$_h$ = 1								
606E$_h$	00$_h$	Velocity Window time		uint16	0~65535	RW	RPDO	5	ms	Stop activate
		Description: when the difference between the target speed in 60FF$_h$ (converted into motor speed rpm) and the actual speed of the motor is within ± 606D, and the duration time reaches 606E$_h$, means that the speed is reached, and the bit10 in status word 6010$_h$ = 1								
606F$_h$	00$_h$	Velocity t$_h$res$_h$old	Threshold of 0° arrival	uint16	0~65535	RW	RPDO	5	User unit/s	Stop activate
		Description: the current speed is under 6069$_h$ within ± 606F$_h$, and the duration time reaches 6070$_h$,means that the speed is 0, and the bit12 in status word 6041$_h$ = 1								
6070$_h$	00$_h$	Velocity t$_h$res$_h$old time	Time window when 0° arrival	uint16	0~65535	RW	RPDO	5	ms	Stop activate
		Description: the current speed is under 6069$_h$ within ± 606F, and the duration time reaches 6070$_h$,means that the speed is 0, and the bit12 in status word 6041$_h$ = 1								
6071$_h$	00$_h$	Target torque		int16	-1000~1000	RW	RPDO	0	0.10%	Stop activate
		Description: 100.0% corresponds to 1 times of motor rated torque								

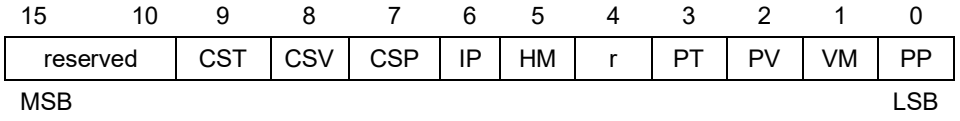
Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6072h	00h	Max torque		uint16	0~2000	RW	RPDO	1000	0.10%	Stop activate
6073h	00h	Max current		uint16	0~2000	RW	NO	1500	0.10%	Stop activate
6074h	00h	Torque demamd value	Torque instruction, reflecting the input torque command in the Enable state	int16	-32768~32767	RO	TPDO	0	0.10%	-
6075h	00h	Motor rated current		uint32	0~11700	RW	RPDO	11700	0.001A	Stop activate
6076h	00h	Motor rate torque		uint32	0~1270	RW	RPDO	1270	0.01N.m	Stop activate
6077h	00h	Torque actual value		int16	-32768~32767	RO	TPDO	-	0.001N • m	-
6078h	00h	Current actual value		int16	-32768~32767	RO	TPDO	-	0.01A	-
607Ah	00h	Target position		int32	-2147483648~2147483647	RW	RPDO	0	User unit	Stop activate
607Bh	01h	MinPosRangLimt		int32	-2147483648~2147483647	RW	RPDO	-1048576	User unit	Stop activate
	02h	MaxPosRangLimt		int32	-2147483648~2147483647	RW	RPDO	1048576	User unit	Stop activate
607Ch	00h	Home offset		int32	-2147483648~2147483647	RW	RPDO	0	User unit	Stop activate
		Description: set the physical position where the mechanical home position deviates from the motor home position in the Homing mode; when the homing operation is completed, takes effect when bit15= 1 in status word 6041h								
607Dh	01h	MinPosLimt		int32	-2147483648~2147483647	RW	RPDO	-65535	User unit	Stop activate
	02h	MaxPosLimt		int32	-2147483648~2147483647	RW	RPDO	65535	User unit	Stop activate
607Eh	00h	Polarity	See section 4.1.4 for details	uint8	0~255	RW	NO	0	1	Stop activate

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
607Fh	00h	Max profile velocity		uint32	0~500000	RW	RPDO	500000	User unit/s	Stop activate
6080h	00h	Max motor speed		uint32	0~4000	RW	RPDO	4000	rpm	Stop activate
6081h	00h	Profile velocity		uint32	0~500000	RW	RPDO	500000	User unit/s	Stop activate
		Description: set the uniform speed in this position command section of the Profile position mode,								
6082h	00h	End velocity		uint32	0~500000	RW	RPDO	0	User unit/s	Stop activate
6083h	00h	Profile acceleration		uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Stop activate
6084h	00h	Profile deceleration		uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Stop activate
6085h	00h	Quick Stop deceleration		uint32	0~4294967295	RW	RPDO	500000	User unit/s^2	Stop activate
6086h	00h	Motion profile type		int16	0~3	RW	RPDO	3	1	Stop activate
		Description: set the curve type of motor position command or speed command. 0 ~ linear, 3 ~ s curve								
6087h	00h	Torque slope		uint32	0~4294967295	RW	RPDO	10	0.10%	Stop activate
		Description: set the torque command acceleration under position torque mode, which means: torque command increments per second								
6088h	00h	Torque profile type		int16	0~2	RW	RPDO	0	1	Stop activate
		0-No profile type，2-profile type								
608Fh	01h	Encoder increments		uint32	0~4294967295	RW	NO	10000	1	Stop activate
	02h	Motor revolutions		uint32	0~4294967295	RW	NO	1	1	Stop activate
	Description: used to set the turns of the motor corresponding to the encoder increment Resolution of position encoder= $\frac{\text{increment of encoder(608Fh:01h)}}{\text{motor turns (608Fh:02h)}}$									

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6091h	01h	Motor revolutions	Transmission ratio	uint32	0~4294967295	RW	NO	1	1	Stop activate
	02h	Shaft revolutions		uint32	0~4294967295	RW	NO	1	1	Stop activate
	Description: it is used to establish the proportional relationship between the load axis displacement and the motor axis displacement specified by the user. See Section 5.2 for details									
6098h	00h	Homing method	See section 6.7.4 for details	int8	17~30	RW	RPDO	20	1	Stop activate
6099h	01h	Speed during search for switch	（high speed）	uint32	0~4294967295	RW	RPDO	10000	User unit/s	Stop activate
	02h	Speed during search for zero	（low speed）	int32	-2147483648~ 2147483647	RW	RPDO	2730	User unit/s	Stop activate
609Ah	00h	Homing acceleration		uint32	0~4294967295	RW	RPDO	409600	User unit/s	Stop activate
60A3h	00h	Profile jerk use	Number of position jerk applications,This device only uses 60A4h:01h and 60A4h:02h	Usint8	2	RW	RPDO	2	1	Stop activate
60A4h	01h	Profile jerk 1		uint32	0~4294967295	RW	NO	50000	User unit/s^2	Stop activate
	02h	Profile jerk 2		uint32	0~4294967295	RW	NO	50000	User unit/s^2	Stop activate
60B0h	00h	Position offset		int32	-2147483648~ 2147483647	RW	RPDO	0	User unit	Stop activate
		Description: set the servo position command offset in the Cyclic sync position mode, after the offset: Servo target velocity = 607Ah+60B0h								
60B1h	00h	Velocity offset		int32	-2147483648~ 2147483647	RW	NO	0	User unit	Stop activate
		Description: set the servo velocity command offset in the Cyclic sync velocity mode, after the offset:								

<table>
<tr><th>Index</th><th>Sub-index</th><th>Name</th><th>Description</th><th>Data type</th><th>Data range</th><th>Access-ibility</th><th>Mapp-ability</th><th>Factory setting</th><th>Unit</th><th>Activate method</th></tr>
<tr><td></td><td></td><td colspan="9">Servo target velocity = 60FF$_h$+60B1$_h$</td></tr>
<tr><td rowspan="2">60B2$_h$</td><td rowspan="2">00$_h$</td><td>Torque offset</td><td></td><td>int32</td><td>-2147483648~ 2147483647</td><td>RW</td><td>RPDO</td><td>0</td><td>User unit</td><td>Stop activate</td></tr>
<tr><td colspan="9">Description: set the servo torque command offset in the Cyclic sync torque mode, after the offset:
Servo target torque = 6071$_h$ + 60B2$_h$</td></tr>
<tr><td>60C1$_h$</td><td>01$_h$</td><td>Interpolation data record</td><td></td><td>int32</td><td>-2147483648~ 2147483647</td><td>RW</td><td>RPDO</td><td>0</td><td>User unit</td><td>Stop activate</td></tr>
<tr><td rowspan="3">60C2$_h$</td><td>01$_h$</td><td>Interpolation time period value</td><td></td><td>uint8</td><td>0~255</td><td>RW</td><td>RPDO</td><td>20</td><td>s</td><td>Stop activate</td></tr>
<tr><td>02$_h$</td><td>Interpolation time index</td><td></td><td>Int8</td><td>-128~127</td><td>RW</td><td>NO</td><td>-1</td><td>1</td><td>Stop activate</td></tr>
<tr><td colspan="9">Description: interpolation period = t×10^n seconds</td></tr>
<tr><td>60C5$_h$</td><td>00$_h$</td><td>Max acceleration</td><td></td><td>uint32</td><td>0~4294967295</td><td>RW</td><td>RPDO</td><td>500000</td><td>User unit/s2</td><td>Stop activate</td></tr>
<tr><td>60C6$_h$</td><td>00$_h$</td><td>Max deceleration</td><td></td><td>uint32</td><td>0~4294967295</td><td>RW</td><td>RPDO</td><td>500000</td><td>User unit/s^2</td><td>Stop activate</td></tr>
<tr><td rowspan="2">60F2$_h$</td><td rowspan="2">00$_h$</td><td>Positioning option code</td><td></td><td>uint16</td><td>0~2</td><td>RW</td><td>RPDO</td><td>0</td><td>1</td><td>Stop activate</td></tr>
<tr><td colspan="9">Description as follow:
<table>
<tr><th>Bit1</th><th>Bit0</th><th>Description:</th></tr>
<tr><td>0</td><td>0</td><td>Relative position shift, relative to the target position（607A$_h$）</td></tr>
<tr><td>0</td><td>1</td><td>Relative position shift, relative to the position command（60FC$_h$）</td></tr>
<tr><td>1</td><td>0</td><td>Relative position shift, relative to the position command（6064$_h$）</td></tr>
</table></td></tr>
<tr><td>60F4$_h$</td><td>00$_h$</td><td>Following error actual value</td><td></td><td>int32</td><td>-2147483648~ 2147483647</td><td>RO</td><td>TPDO</td><td>0</td><td>User unit</td><td>-</td></tr>
</table>

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
60FA$_h$	00$_h$	Control effort	Output from regulator	int32	-2147483648~ 2147483647	RO	TPDO	0	指令单位	-
60FC$_h$	00$_h$	Position demand internal value		int32	-2147483648~ 2147483647	RO	TPDO	0	User unit	-
		Description:Position command (60FCh)= Position command(6062h)x Electronic gear ratio(6091h)								
60FD$_h$	00$_h$	Input signal (DI signal) monitoring	Input signal (DI signal) monitoring	uint32	0~4294967295	RO	TPDO	0	1	-
60FE$_h$	01$_h$	Physical output	Physical output	uint32	0~4294967295	RO	TPDO	0	1	-
	02$_h$	Physical output enable	Physical output enable	uint32	0~4294967295	RO	TPDO	0	1	-
60FF$_h$	00$_h$	Target velocity		int32	-2147483648~ 2147483647	RW	RPDO	0	User unit/s	Stop activate
		Description: set the speed command in the the Profile velocity mode and Cyclic sync velocity mode								
6502$_h$	00$_h$	Supported drive modes	Supported drive modes	uint32	0~4294967295	RO	NO	1023	1	-



Description: CST, CSV, CSP, IP, HM, PT, PV, VM, and PP bits: 1 = supported; 0 = not supported

11 Version Information

Manual version	Date	Revision History
A	2025/9/9	Establishing
A01	2026/1/15	Modify description
A02	2026/6/27	1. Modify some parts 2. Delete the anti-hand-clinging function

- The whole or part contents of this manual is prohibited from being reprinted or copied without authorization.
- Performance, specifications and appearance of products may change without prior notice due to improvements, please understand.
- We strive to make the contents of the manual as correct as possible. if you find any problems、errors or omissions, please contact Beijing NiMotion Control Technology Co., Ltd.

Beijing NiMotion Control Technology Co., Ltd.
Building 3, yard 12,Jinxing Road, Daxing District, Beijing
Zip code： 102628
Tel： (010)60213882 Fax： (010)60213882
E-mail： nimotion@nimotion.com
http://www.nimotion.com

