PMM60L、STM-M、BLM 、TSM series integrated motor

Modbus communication user manual

Version No.A03

Beijing NiMotion Control Technology Co., Ltd

Contents

1 About the manual

1.1 Introduction

This manual is used to explain the programming and operation methods of the integrated brushless motor and stepper servo motor produced by Beijing NiMotion Control Technology Co., Ltd.

1.2 Scope of Application

Suitable for Modbus communication protocol PMM60L、STM-M、BLM 、TSM series integrated motor.

1.3 The Meaning of Numerical Values

In this manual, values are generally expressed in decimal format. If it must be expressed in hexadecimal, it is indicated by the subscript "h" at the end of the number, for example: 1003_h.

The object directory index and sub-index of the object, expressed as: <index>: <sub-index>

For example, the sub-index 02 of the object 1003_h is represented as 1003_h:02_h(0004_h);

1.4 Bit Definition

Each bit of the object always starts at 0 in the LSB number.

	MSB							LSB	
Bit Number	7	6	5	4	3	2	1	0	
Bits	0	1	0	1	0	1	0	1	$55_{hex}=85_{dec}$

2 Communication Network

2.1 Overview of Modbus

Beijing NiMotion Control Technology Co., Ltd. Integrated motor uses RS-485 bus communication control, and the protocol supports the standard Modbus-RTU protocol.

The communication method of Modbus protocol is single master station/multiple slave stations. Only the master station can issue inquiries (inquiries). The slave station performs the processing required by the query and replies to the response message.

2.2 Modbus-RTU Transmission Mode

When the device uses RTU (Remote Terminal Unit) mode to communicate on the Modbus serial link, each 8-bit byte in the message contains two 4-bit hexadecimal characters. The main advantage of this mode is higher data density and higher throughput than ASCII mode at the same baud rate. Each message must be transmitted in a continuous stream of characters.

Frame description:

Table 2-1

Child node address	Function code	Data	CRC
1 byte	1 byte	0 to 252 bytes	2 bytes Low CRC High CRC

The message frame is distinguished by an idle interval of at least 3.5 character times, called t3.5. This module uses the standard t3.5 as the idle interval. The schematic diagram of RTU frame transmission idle interval is shown

in Figure 1.

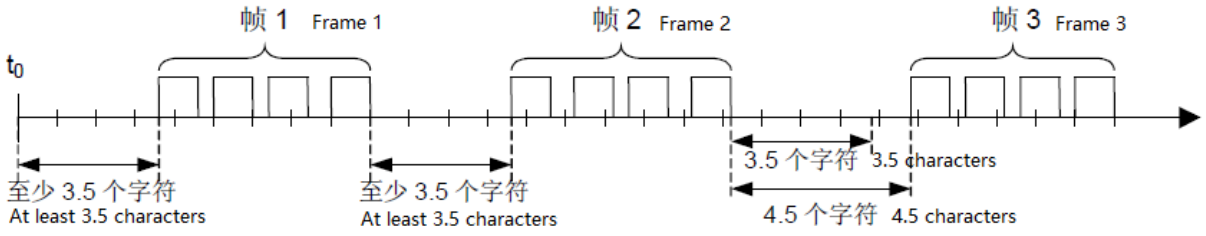


Figure 2-1 RTU frame transmission idle interval distinction diagram

2.3 Data Type

Modbus is based on a series of data models on tables with different characteristics. The two basic forms are:

Table 2-2

Basic form	Object type	Access type	Content
Input register	16 bits	Read only	I/O systems provide this type of data
Holding register	16 bits	Read and write	Change this type of data through the application

2.4 Modbus-RTU Function Code

Table 2-3

	Function code name	Function code	Sub-function code	Broadcast message
16-bit access	Read input register	0x04	—	Not support
	Read holding register	0x03	—	Not support
	Write a single register	0x06	—	Support
	Write multiple registers	0x10	—	Support

2.4.1 Read Holding Register (0x03)

Read the current value of baud rate 200C$_h$: 03$_h$ (0231$_h$).

Request message

Table 2-4

Slave address	Function code	Register address	Number of registers	CRC check value
01	0x03	02 31	00 01	Skip

Response message

Table 2-5

Slave address	Function code	Bytes	Register value	CRC check value
01	0x03	02	00 00	Skip

The current baud rate value is 0 (9.6kbps).

2.4.2 Read Input Register (0x04)

Read the value of the input voltage $200B_h$: 15_h ($01F7_h$).

Request message

Table 2-6

Slave address	Function code	Register address	Number of registers	CRC check value
01	0x04	01 F7	00 01	Skip

Response message

Table 2-7

Slave address	Function code	Bytes	Register value	CRC check value
01	0x04	02	00 E7	Skip

The current input voltage value is 0x00E7 (23.1V).

2.4.3 Write a Single Holding Register (0x06)

Set the baud rate of the slave address 0x01 $200C_h$: 03_h (0231_h) to 0x01 (9.6kbps).

Request message

Table 2-8

Slave address	Function code	Register address	Register value	CRC check value
01	0x06	02 31	00 01	Skip

Response message

Table 2-9

从机地址 Slave address	功能码 Function code	Register address	Register value	CRC check value
01	0x06	02 31	00 01	Skip

2.4.4 Write Multiple Holding Registers (0x10)

Set the slave address $200C_h$: 02_h (0230_h) of the slave address 0x01 to 0x02 and the baud rate 200Ch: 03_h (0231_h) to 0x01 (9.6kbps).

Request message

Table 2-10

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
01	0x10	02 30	00 02	04	00 02 00 01	Skip

Response message

Table 2-11

Slave address	Function code	Register address	Number of registers	CRC check value
01	0x10	02 30	00 02	Skip

2.5 Communication Exception Response

When the master station device sends a request to the slave station device, the master station expects a normal response. The inquiry from the master station may cause one of the following four events:

- If the slave device receives the request without communication error and can process the inquiry normally, then the slave device will return a normal response.
- If the slave station does not receive the request due to a communication error, it cannot return a response. The master program will be processed according to the timeout.
- If the slave station receives the request, but detects a communication error (parity, CRC...), then the response cannot be returned. The master program will be processed according to the timeout.
- If the slave device receives a request without communication error, but cannot process the request (for example, if the request reads a non-existing input register), then the slave will return an abnormal response to notify The cause of the error on the client.

The abnormal response message has two different fields from the normal response message:

Function code field:

In the normal response, the slave station assigns the function code of the original request to the function code field of the response. The MSB of all function codes is 0 (their values are all lower than 80 hex). In the abnormal response, the slave station sets the MSB of the function code to 1. This makes the function code value in the abnormal response higher than the function code value in the normal response by 80 hexadecimal.

By setting the function code MSB, the application of the master station can recognize the abnormal response, and can detect the data field of the abnormal code.

Data field:

In a normal response, the slave can return data or statistics (any information requested in the request) in the data field.

In the exception response, the slave returns the exception code in the data field. This explains the cause of the exception.

The exception response message of all supported function codes is the function code plus 0x80.

All abnormal function codes are 0x83, 0x84, 0x86, 0x90.

The exception response message corresponding to function code 0x06 is as follows:

Table 2-12

Slave address	Function code	Exception code	CRC check value
01	0x86	01	-

Table 2-13

Exception code	Exception name	Description
01	Illegal function code	The function code cannot be recognized and is not within 0x00~0x0F
02	Illegal data address	Data address exceeds definition
03	Illegal data value	Values other than register storage
04	Slave failure	Produces an unrecoverable error
05	Confirm	The processing time of the slave is long, and the master often ask
06	Slave busy	The master sends a request when the slave device is idle
12	Slave alarm	There is an alarm for the current motor status

2.6 Communication Parameter Configuration

2.6.1 Baud Rate Setting

The communication rate of the device can be changed by setting the communication baud rate register of the RS-485 communication interface, but the parameter needs to be saved after setting the baud rate, which will take effect after the device is turned on or restarted next time.

The corresponding index of the baud rate parameter is $200C_h$: 03_h (0231_h), and the function codes of the operable holding register are 0x03, 0x06, 0x10. The factory default baud rate is 115.2Kbps.

Example of setting the communication baud rate of the slave address to 115.2 Kbps

Request message sent

Table 2-14

Slave address	Function code	Register address	Register value	CRC check value
01	0x06	02 31	00 05	Skip

Response message

Table 2-15

Slave address	Function code	Register address	Register value	CRC check value
01	0x06	02 31	00 05	Skip

Save the set communication baud rate parameters

Request message sent

Table 2-16

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
01	0x10	00 26	00 02	04	65 76 61 73	Skip

Response message

Table 2-17

Slave address	Function code	Register address	Number of registers	CRC check value
01	0x10	00 26	00 02	Skip

Set value and corresponding table with baud rate:

Table 2-18

Hex	Baud rate
0	9.6Kbps
1	9.6Kbps
2	19.2Kbps
3	38.4Kbps
4	57.6Kbps
5	115.2Kbps
6	256Kbps
7	500Kbps
8	1Mbps
9	1.5Mbps

2.6.2 Node ID

The address of the device can be set from the device address $200C_h$: 02_h (0230_h), and the range can be set from 1 to 247. The factory default slave address is 0x01. After setting the slave address successfully, you need to save the parameter settings, which will take effect after the device is turned on or restarted next time. Refer to the baud rate setting for specific operations.

2.6.3 Network Data Format

The network data format of the device communication is changed by setting the network data format $200C_h$: 04_h (0232_h). The setting value correspondence table is as follows

Table 2-19

Hex	Description
0	8 data bits, even parity, 1 stop bit
1	8 data bits, odd parity, 1 stop bit
2	8 data bits, no parity, 1 stop bit
3	8 data bits, no parity, 2 stop bits

2.6.4 Communication Line Connection

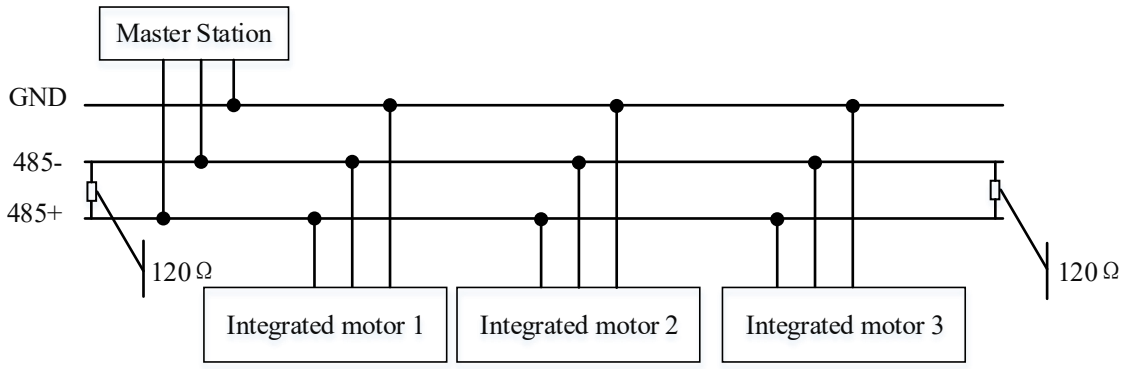


Figure 2-1

2.6.5 Broadcast Preemption

When the communication parameters of the motor are uncertain and forgotten, the power can be restarted after being seized through broadcast messages. The broadcast preemption request message needs to be sent within 1 second before the motor is powered on. After the preemption is successful, the RUN indicator flashes. After the preemption is successful, the communication parameters of the motor are restored to the default parameters, that is, the communication parameters are restored to the factory settings, the slave address is equal to 1, the baud rate is equal to 115200, the parity bit is none, 8 data bits, and 1 stop bit. The serial number of the motor adopts the byte order of hexadecimal high byte first.

Table 2-20 Broadcast preemption request message

Slave address (1B)	Function code (1B)	Motor serial number (4B)	CRC check value (2B)
0x00	0xD2	0x00 00 00 00	-

3 Device Control

3.1 CiA402 State Machine

By operating a state machine, the operating state of the integrated low-voltage motor is controlled and switched, which is defined in the standard CiA402. The motor driver is controlled by the control word 6040_h (0380_h), and the actual status of the device can be read back from the status word object 6041h (0381_h).

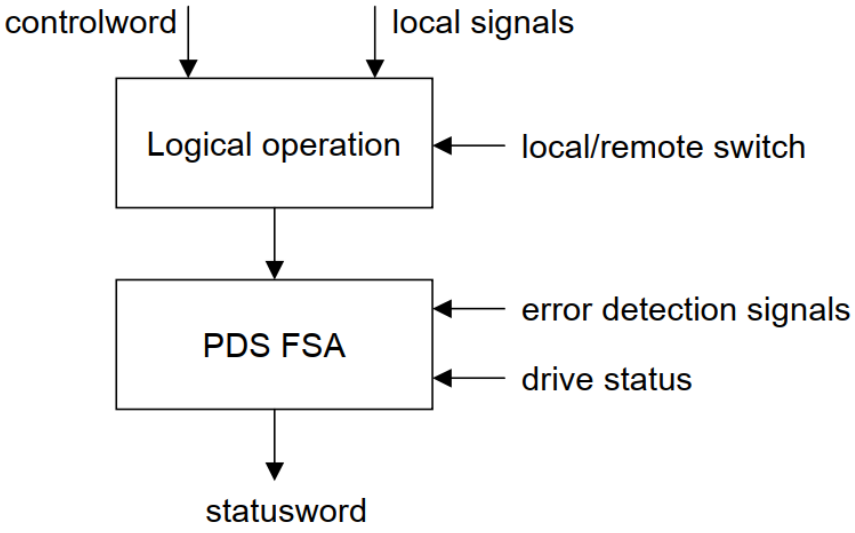


Figure 3-1

3.1.1 State transition diagram

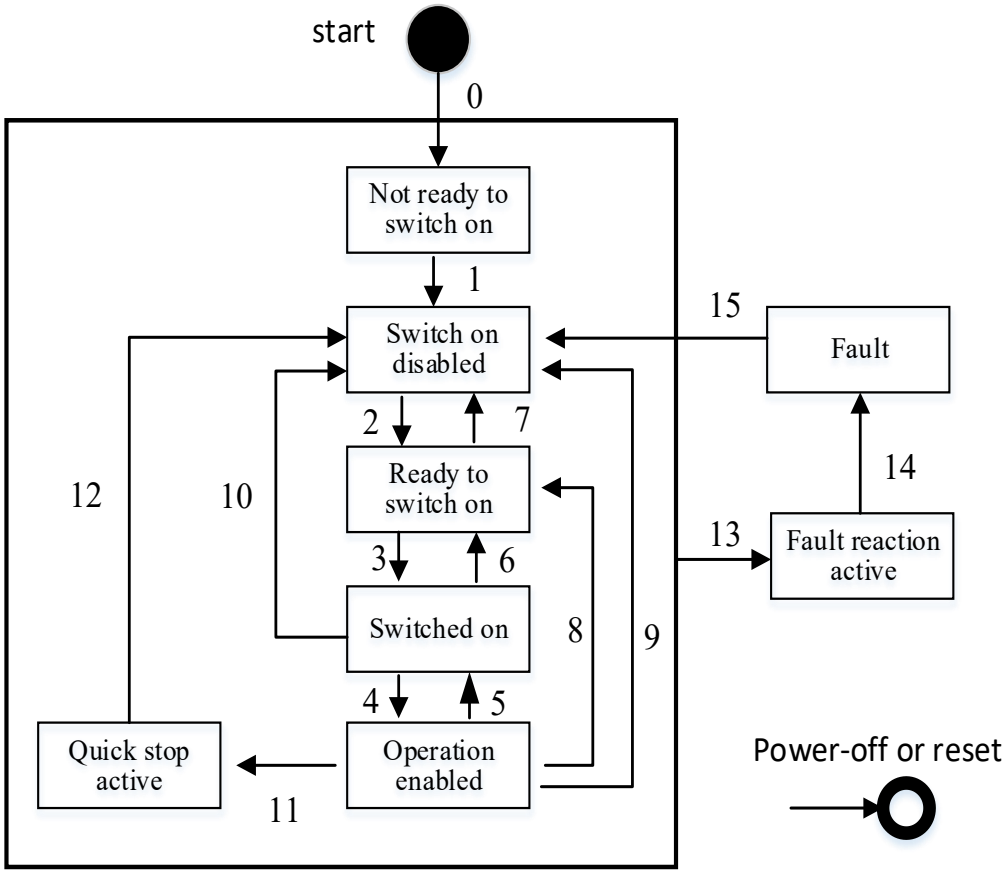


Figure 3-2

The corresponding motor conditions in each state are shown in the following table:

Table 3-1

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

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
Configuration allowed	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

The control word and state machine state switching is as follows:

Table 3-2

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 SWITCHED ON state functionality

The status words in each status are as follows:

Table 3-3

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

Note: x means that the bit does not care, bit8 to bit15 have different meanings in different modes, please refer to each operating mode for the specific status of everyone.

3.1.2 Control word

The bits of the control word are defined as follows:

15	11	10	9	8	7	6	4	3	2	1	0
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

- Note

It is meaningless to assign each bit of the control word individually, and it must be combined with other control bits to form a certain control instruction. Bit0~bit3 and bit7 have the same meaning in each motion mode. The commands must be sent in order before the motor driver can be guided into the expected state according to the CiA402 state machine switching process. Each command corresponds to a certain state. bit4~bit6 are related to each sport mode (please check the control commands in different modes)

3.1.3 Status word

The status word bits 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 3-4

位 Bit	描述 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

- **Note**

It is meaningless to assign each bit of the status word individually, and it must be combined with other control bits to form the current status of the feedback motor.

Bit 0-bit9 has the same meaning,by which the motor feeds back a certain status after sending commands by the control word 6040h sequentially.

Bit10, bit11, and bit15 have the same meaning in each sport mode, and feedback the state of the motor after executing a certain sport mode.

3.1.4 Stop Mode

The stop mode is the action when the state machine leaves the "running" state. The corresponding setting objects are shown in the following figure:

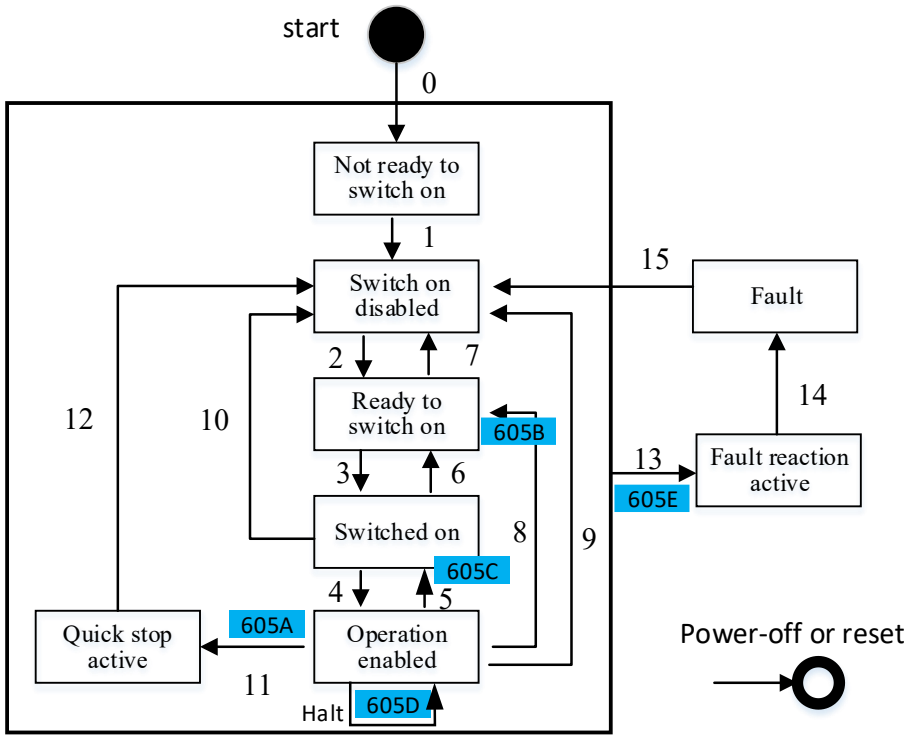


Figure 3-3

- **Fault shutdown**

When a fault or warning occurs, it automatically enters the fault shutdown state. The fault shutdown mode is related to $605E_h$ ($03C1_h$) (Fault reaction option code). When $605E_h$($03C1_h$) = -1, the stop mode is related to the reaction code in the fault code, which is a configurable parameter; when $605E_h$（$03C1_h$）> 0, the stop mode is selected through the object dictionary $605E_h$（$03C1_h$）, and the relationship between the set value of $605E_h$（$03C1_h$） and the stop mode is as follows:

Table 3-5

Set value	Stop mode
-1	The stop mode is related to the reaction code in the fault code, see "9.1 Alarm Code" for details
0x00	Free stop, keep free running state.
0x01	Slowly stop with the deceleration ramp set at 6084_h($03FE_h$)
0x02	Quick stop with the deceleration ramp set at 6085_h(0400_h)

- **Quick stop**

In the non-fault state, when the control word 6040_h（0380_h） = 0x02, a quick stop is executed.

The quick stop mode is selected through the object dictionary 605Ah （$03BF_h$）. The meaning of the set value of $605A_h$ ($03BF_h$）is shown in the table below:

Table 3-6

Setting Value	Stop Mode
0x00	Slow down freely and keep operating freely.

Setting Value	Stop Mode
0x01	Slow down on slow down ramp set in 6084_h($03FE_h$),and keep operating freely.
0x02	Slow down on slow down ramp set in 6085_h(0400_h),and keep operating freely.

- **Halt**

Bit 8 of the control word 6040_h (0380_h) is a pause function bit, and a pause is performed when bit 8 changes from 0 to 1.

The stop mode is selected through the object dictionary $605D_h$（$03C0_h$） (Halt option code). The meaning of the setting value of $605D_h$（$03C0_h$） is shown in the table below:

Table 3-7

Set value	Stop Mode
0x00	Reserved
0x01	Slow down on slow down ramp set in 6084_h($03FE_h$),and stay in Clamped state.
0x02	Slow down on slow down ramp set in 6085_h(0400_h),and stay in Clamped state.

- **Disabled operation and shutdown**

The disabled operation mode is selected through the object dictionary $605C_h$ （$03BE_h$）. The set value and meaning of $605C_h$ （$03BE_h$）are shown in the table below:

Table 3-8

Set value	Stop Mode
0x00	Slow down freely and keep operating freely..
0x01	Slow down on slow down ramp set in 6084_h($03FE_h$),and keep operating freely.

- **Shutdown**

Shutdown is selected through the object dictionary $605B_h$（$03BD_h$）. The meaning of the set value of $605B_h$($03BD_h$）is shown in the table below:

Table 3-9

Set value	Stop mode
0x00	Slow down freely and keep operating freely.
0x01	Slow down on slow down ramp set in 6084h(03FEh),and keep operating freely.

4 Unit Conversion

In order to adapt to the application of all walks of life, it is convenient for users to define their own command units. The internal unit conversion unit of this device can convert any user unit (User Unit: UU) into the internal operating unit of the drive (Encoder Increment: Inc.)

4.1 $608F_h$: Position Encoder Resolution

This object should indicate the configured encoder increment and the number of motor revolutions. The calculation formula of position encoder resolution is as follows:

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

The following table describes the objects:

表 Table 4-1

Attributes	Description
Index	$608F_h$（0406_h、0408_h）
Name	Position encoder resolution
Data structure	Array
Type of data	Unsigned32

The following table describes the sub-index:

Table 4-2

Attributes	Description	
Subindex	$01_h(0406_h)$	$02_h(0408_h)$
Description	Increment of encoder	Motor turns
Accessibility	rw	rw
Can it be mapped	No	No
Type of data	Unsigned32	Unsigned32
Defaults	10000	1

4.2 6091_h: Transmission Ratio

The object should indicate the number of motor shaft revolutions and drive shaft revolutions. The calculation formula of the transmission ratio is as follows:

$$\text{Transmissi on ratio} = \frac{\text{Motor shaft revolution}\ (6091_h:01_h)}{\text{Drive shaft revolution}\ (6091_h:02_h)}$$

The following table describes the objects:

Table 4-3

Attributes	Description
Index	6091_h（$040E_h$、0410_h）
Name	Gear ratio
Data structure	Array
Type of data	Unsigned32

The following table describes the sub-index:

Table 4-4

Attributes	Description	
Sub-index	01_h ($040E_h$)	02_h (0410_h)
Description	Motor shaft revolutions	Drive shaft revolutions
Accessibility	rw	rw
Can it be mapped	No	No
Data range	Unsigned32	Unsigned32
Defaults	1	1

The transmission ratio is used for the proportional relationship between the drive shaft displacement and the motor shaft displacement specified by the user.

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

  Motor shaft position feedback = drive shaft position feedback × gear ratio

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

$$\text{Motor shaft speed (rpm)} = \frac{\text{Drive shaft speed} * \text{transmiss ion ratio } 6091_h}{\text{Encoder resolution}} * 60$$

- Relationship between motor shaft acceleration (rpm/s) and drive shaft acceleration (unit/s2):

$$\text{Motor shaft acceleration} = \frac{\text{Drive shaft acceleration} * \text{transmiss ion ratio } 6091_h}{\text{Encoder resolution}} * 60$$

4.3 $607E_h$: Polarity

The object dictionary is used to set whether the position command and speed command are multiplied by 1 or -1. The position polarity is only used inside the profile position mode and the cycle synchronized position mode, and has no effect on the homing mode. The speed polarity is only used in profile Velocity mode and cyclic Velocity mode. The specific definition is shown in the following table:

Table 4-5

Bit-value meaning		
7	6	5~0
Position polarity	Velocity polarity	reserved (0)

The bit values in the table above are expressed as: 0 means multiply by 1, and 1 means multiply by -1. For example, Position Polarity = 0, which means that the Position demand value will be multiplied by 1. The following table describes the objects:

Table 4-6

Attributes	Description
Index	$607E_h$（$03F3_h$）
Name	Polarity
Data structure	Variable
Accessibility	rw
Type of data	Unsigned8
Defaults	0

4.4 Examples

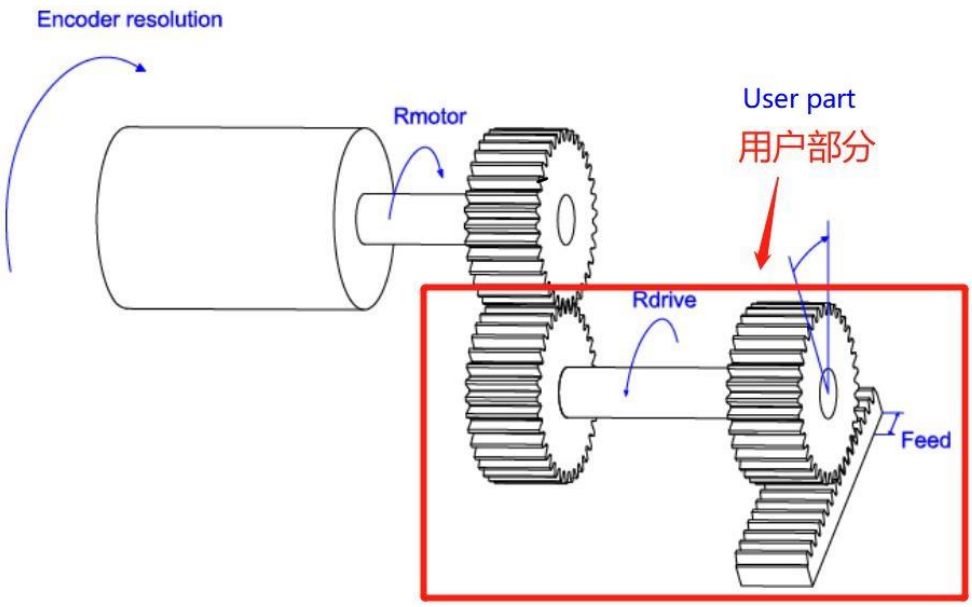


Figure 4-1

As shown in the figure above, Rmotor is the motor shaft and Rdrive is the drive shaft. Assumption: Encoder resolution = 10000 when the motor shaft rotates one revolution; when Rdrive rotates one revolution, Rmotor rotates three revolutions; from the above conditions, it can be seen that if the drive shaft turns one revolution, the motor needs to take 3*10000 encoder codes , That is, the motor rotates 3 times. The set parameters can be calculated according to the following formula:

position encoder resolution（$608F_h$）：

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

gear ratio（6091_h）：

$$\text{Transmissi on ratio} = \frac{\text{Motor shaft revolution s } (6091_h : 01_h)}{\text{Drive shaft revolution s } (6091_h : 02_h)} = \frac{3}{1} = 3$$

5 Operating Mode

5.1 Overview of Operation Model

The control mode of the integrated motor is divided into CiA402 mode and NiMotion mode. 2002_h: 01_h ($00B1_h$) (control mode selection) is used to determine whether the motor is in CiA402 mode or NiMotion mode. The objects are described as follows:

Table 5-1

Attributes	Description
Index	2002_h
Name	Basic control parameter
Data structure	Record
Type of data	-

Sub-index 01_h to determine the mode:

Table 5-2

Subindex	01_h（$00B1_h$）
Description	Control Mode Select
Accessibility	rw
Can it be mapped	No
Data range	Unsigned8
Defaults	0
Value meaning	0x00: CiA402 mode 0x01: NiMotion position mode 0x02: NiMotion Velocity mode 0x03: NiMotion torque mode 0x04: NiMotion open loop mode

After selecting CiA402 mode, you need to set 6060_h ($03C2_h$) to select the specific sport mode (control mode selection, invalid running state setting), the object description is as follows:

Table 5-3

Subindex	6060_h（0x3C2）
Description	Mode Operation
Accessibility	rw
Can it be mapped	No
Data range	Unsigned8
Defaults	0x01
Sub-index	-
Meaning of Value	0x00: No definition 0x01: Profile Position Mode (PP) 0x02: Velocity Mode (VM) 0x03: Profile Velocity Mode (PV)

Meaning of Value	0x04: Profile Torque Mode (PT) 0x05: No definition 0x06: Home return mode (HM) 0x07: Interpolation mode (IP) 0x08: Cyclic synchronized position mode (CSP) 0x09: Cyclic Synchronized Velocity Mode (CSV) 0x0A: Cyclic synchronous torque mode (CST)

5.2 Position Control Function

5.2.1 Structure Diagram

Position following error

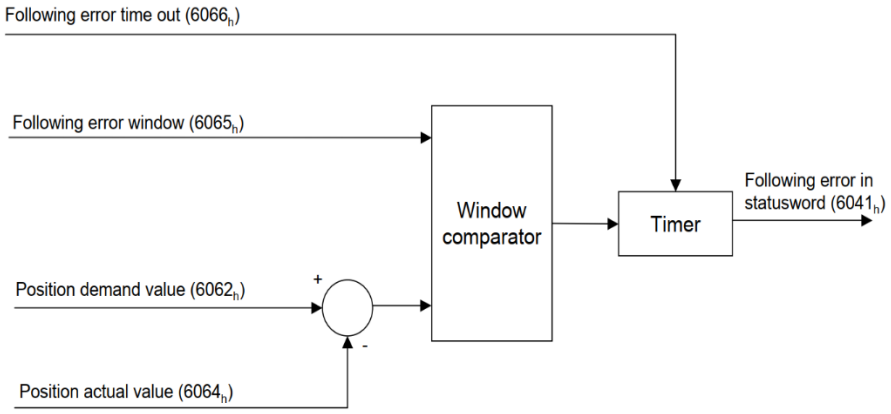


Figure 5-1

Location reached

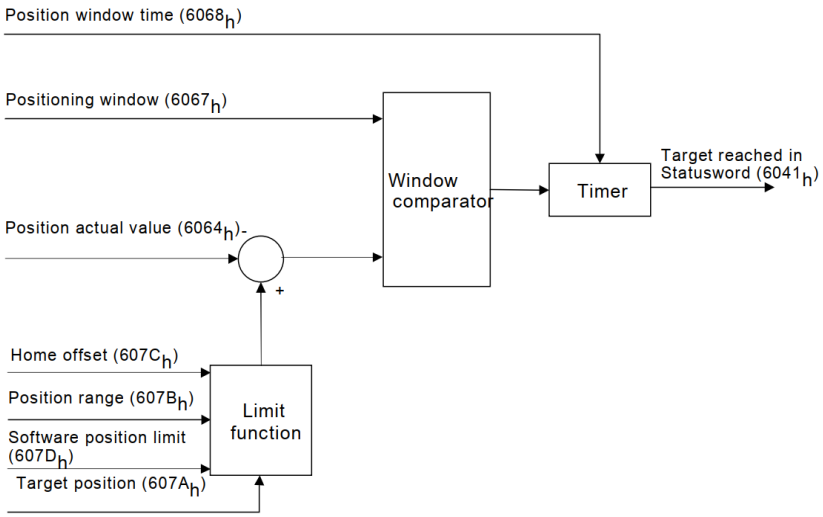


Figure 5-2

5.2.2 Related Objects

Following error

Table 5-4

Object index	Description
$6041_h(0381_h)$	Status word
$6062_h(03C4_h)$	Command value of current target position inside the drive (user unit)
$6064_h(03C8_h)$	Motor current user absolute position feedback (user unit)
$6065_h(03CA_h)$	Following deviation threshold window (user unit)
$6066_h(03CC_h)$	Threshold for following deviation timeout time (unit: ms)

Target reached

Table 5-5

Object index	Description
$6041_h(0381_h)$	Status word
$6064_h(03C8_h)$	Motor current user absolute position feedback (user unit)
$6067_h(03CD_h)$	Location arrival window (user unit)
$6068_h(03CF_h)$	Position arrival window time (unit: ms)
$607A_h(03E7_h)$	Preset target location (user unit)
$607B_h(03E9_h$ 和 $03EB_h)$	Location range limitation (user unit)
$607C_h(03ED_h)$	Origin offset value (user unit) Specific description: Set the physical position where the motor origin deviates from the mechanical zero point under the origin return. Set the physical position where the mechanical zero point deviates from the motor origin in the position control mode (profile position mode, interpolation mode, zero point return). Origin offset effective condition: this time after power-on operation, the origin zero return operation has been completed, bit 15 of status word $6041_h(0381_h)$=1.
$607D_h(03EF_h$ 和 $03F1_h)$	Target position limit (user units)

5.2.3 Functional Description

Position following error

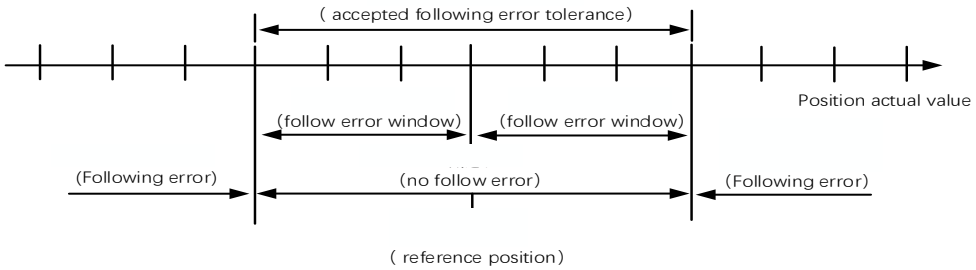


Figure 5-3

The following error refers to the deviation between the reference position 6062_h ($03C4_h$) and the actual position 6064_h ($03C8_h$). Within the time set by 6066_h ($03CC_h$), if the following error value is always greater than the value of the following error window 6065_h ($03CA_h$) (as shown in Figure 6.2, the following error exceeds the acceptable following error range), then the status The bit 13 (following error) of the word 6041_h (0381_h)will be set to 1. The timing diagram is shown in the figure below:

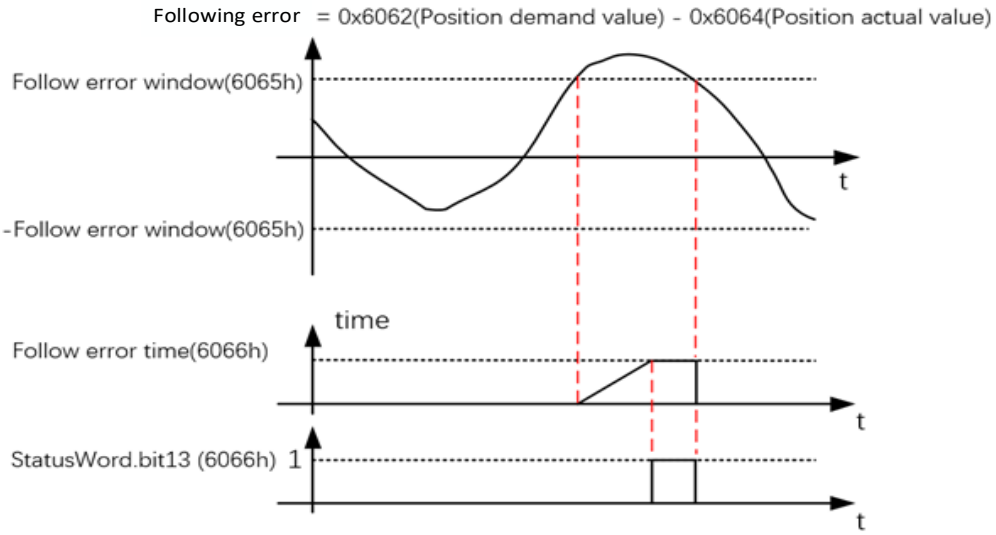


Figure 5-4

Target reached

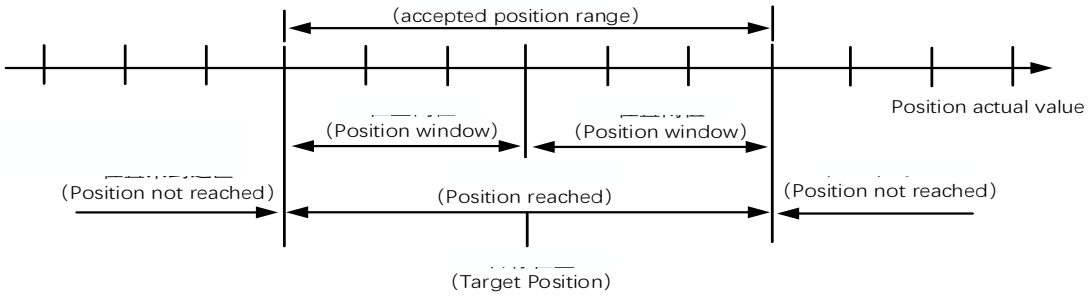


Figure 5-5 Schematic diagram of the acceptable range of error:

The position difference is the difference between the target position $607A_h$ ($03E7_h$) and the actual position 6064_h ($03C8_h$). If the difference is stable within the acceptable position range (as shown in the above figure) and reaches the set time 6068_h ($03CF_h$), then bit10 (target reached) of the status word (Status word) will be set to 1, indicating the target position Arrivals. The timing diagram is shown below:

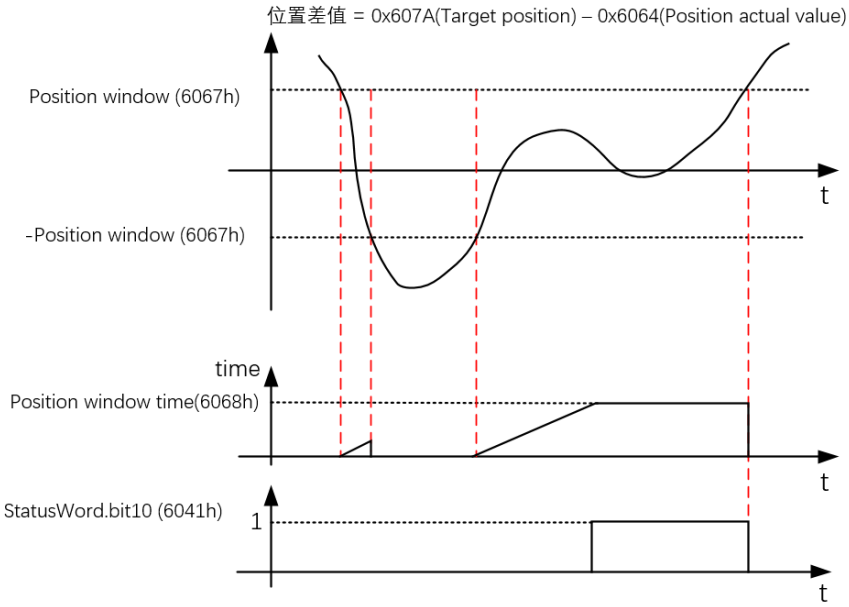


Figure 5-5

5.3 Profile Position Mode (PP)

This mode is mainly used for point-to-point positioning applications. In this mode, the upper computer gives the target position (absolute or relative), the speed of the position curve, acceleration, deceleration and deceleration, the internal track generator will generate the command of the target position curve according to the settings, and the internal driver will complete the position control, speed control and torque control. The overall structure is as follows:

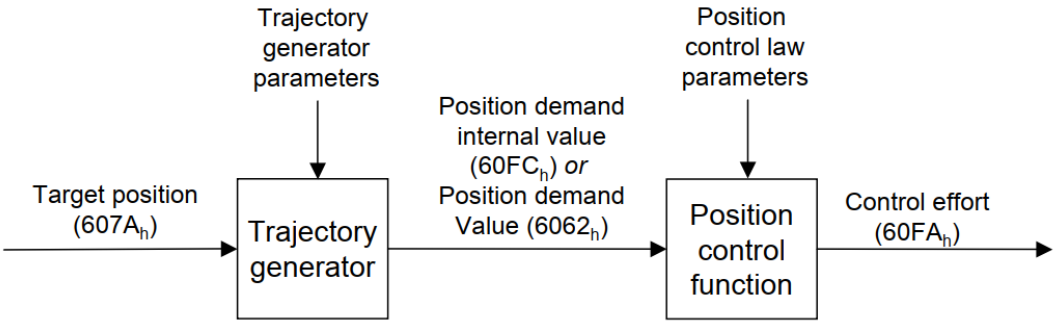


Figure 5-6

5.3.1 Structure Diagram

The following figure defines the detailed structure of the trajectory generator:

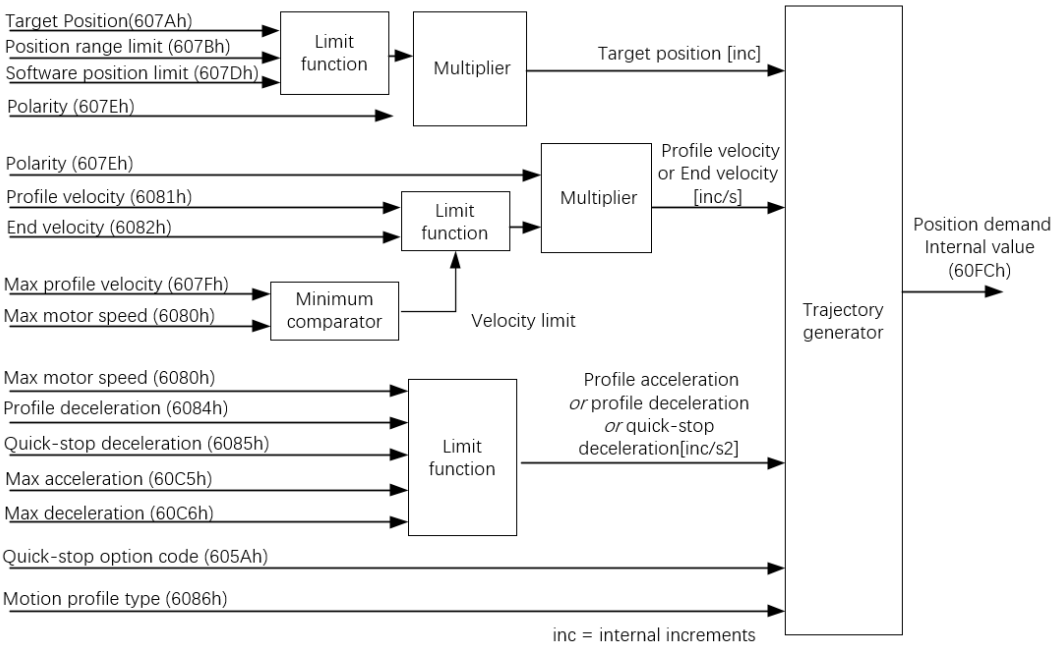


Figure 5-7

5.3.2 Related Objects

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

Table 5-6

Object index	Description
605D$_h$(03C0$_h$)	Pause mode selection
605A$_h$(03BF$_h$)	Quick stop mode selection
6062$_h$(03C4$_h$)	Command value of current target position inside the drive (user unit)
6063$_h$(03C6$_h$)	Motor current absolute position feedback (encoder unit)
6064$_h$(03C8$_h$)	The current position of the motor (user unit)
607A$_h$(03E7$_h$)	Preset target location (user unit)
607B$_h$(03E9$_h$ and 03EB$_h$)	Location range limitation (user unit)
607D$_h$(03EF$_h$ and 03F1$_h$)	Limitation of target location (user unit)
607E$_h$(03F3$_h$)	Direction of rotation (polarity), see "4.3 607E$_h$（03F3$_h$）: Polarity" for details
607F$_h$(03F4$_h$)	The maximum profile speed (user units/s) during operation plays a role in limiting the speed.
6080$_h$(03F6$_h$)	Maximum speed of motor (rpm)
6081$_h$(03F8$_h$)	The profile speed (user unit/s) of the constant speed stage during the operation of this segment of displacement command, that is, the speed to reach the end of the acceleration ramp during positioning. The size is limited by 607F$_h$(03F4$_h$).
6082$_h$(03FA$_h$)	The profile end speed, the speed when reaching the target position (user units/s), and the speed at the end of the slope, usually set the object to zero, so that the speed just decreases to 0 when reaching the target position. The size is limited by 607F$_h$(03F4$_h$).
6083$_h$(03FC$_h$)	The profile acceleration (user units/s2) during operation is limited by 60C5h（043B$_h$）.
6084$_h$(03FE$_h$)	The profile deceleration (user units/s2) during operation is limited by 60C6$_h$（043D$_h$）.
6085$_h$(0400$_h$)	Stop deceleration when executing "fast stop" (user unit/s^2)
6086$_h$(0402$_h$)	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 60A4$_h$: 01$_h$-02$_h$ (041E$_h$、0420$_h$)will be used to limit the impact (acceleration), that is, the S-curve, the device only uses the 01$_h$ and 02$_h$ indexes.
60C5$_h$(043B$_h$)	Maximum acceleration limit (user units/s2)
60C6$_h$(043D$_h$)	Maximum deceleration limit (user units/s2)
60F2 $_h$(043F$_h$)	Targeting options
60FC $_h$(0446$_h$)	The output of the trajectory generator is the internally planned real-time position command (encoder unit)

The following figure shows the effect of speed, acceleration and impact parameters on the running process:

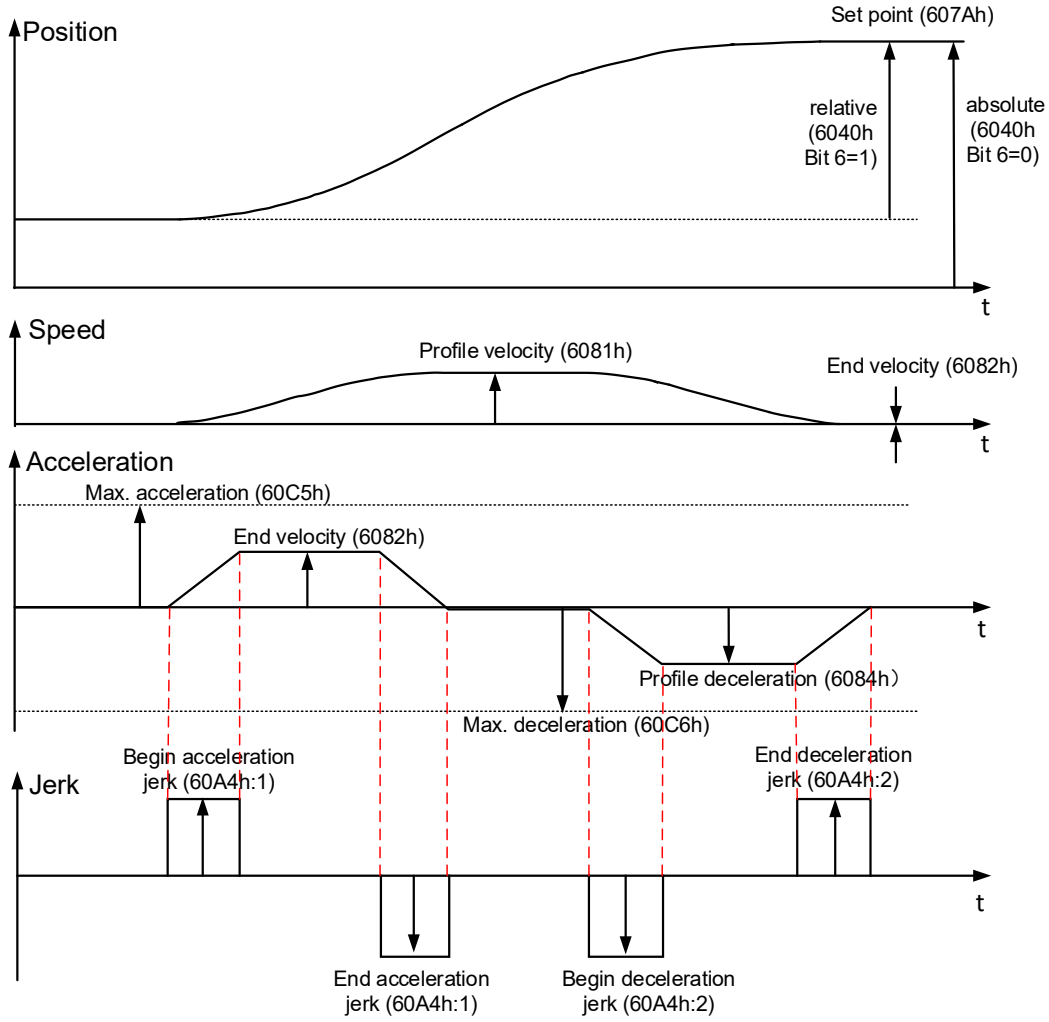


Figure 5-8

5.3.3 Control Commands and Status Messages

Enable

To enable this mode, the value must be set to "0" in object 2002_h: 01_h （$00B1_h$）and "1" in object 6060_h ($03C2_h$).

Control word

15 10	9	8	7	6	5	4	3 0
(see 3.1.2)	Change on set-point	Halt	(see3.1.2)	abs/rel	Change set immediately	New set-point	(see3.1.2)

MSB LSB

In profile position mode, the following bits in object 6040_h (0380_h) have special functions

Table 5-7

Bit 5	Bit 4	Meaning
0	0->1	For non-immediate updates, the running task being executed will be completed before the next running task is started
1	0-> 1	Update immediately, the running task triggered by bit 4 will be executed immediately

Table 5-8

Bit	Value	Meaning
6	0	Target position is absolute position command
	1	The target position is a relative position command. The target position is a relative position based on the current position. The reference position depends on bits 0 and 1 of $60F2_h$($043F_h$)
8	0	Should perform or continue positioning
	1	The motor will slow down and stop moving, the deceleration depends on the object $605D_h$($03C0_h$)

Status word

15 14	13	12	11	10	9 0
(see 3.1.3)	Following error	Set-point acknowledge	(see 3.1.3)	reached Target	(see 3.1.3)

MSB LSB

In profile position mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-9

Bit	Value	Meaning
10	0	In non-paused state (bit8 = 0 at 6040_h（0380_h）): The target has not been reached In the pause state (bit8 = 1 at 6040_h（0380_h）): the motor decelerates
	1	In non-paused state (bit8 = 0 at 6040h （0380_h）): target arrived In the pause state (bit8 = 1 at 6040_h（0380_h）): the motor speed is 0
12	0	The previous positioning point is completed, waiting for the new set point
	1	The previous anchor point is still being processed, the new anchor point will overwrite the old anchor point

13	0	No following error
	1	Follow error

5.3.4 Functional Description

The following is an explanation of the timing of the control commands with immediate update and non-immediate update:

1） Control instruction timing-update immediately

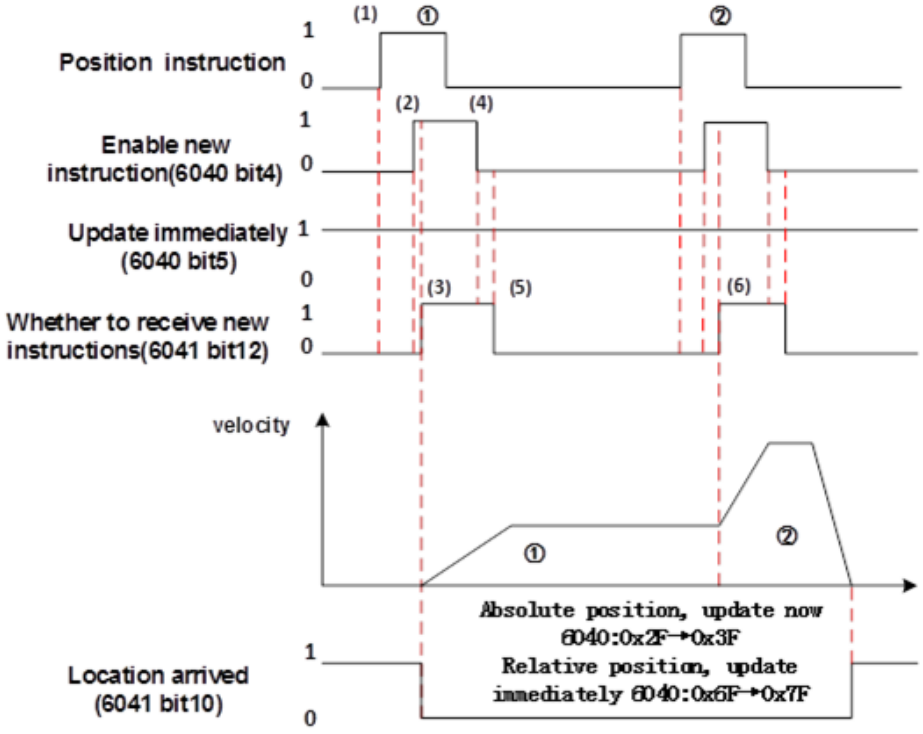


Figure 5-9

The following table explains some explanations in the process of the above picture:

Table 5-10

（1）	The host computer updates the displacement command (including target displacement $607A_h$($03E7_h$), acceleration time 6083h($03FC_h$), deceleration time 6084_h($03FE_h$), profile speed 6081_h($03F8_h$), etc.)
（2）	Set bit 4 of 6040_h （0380_h）from 0 to 1, indicating that the slave has a new displacement command (bit 5 of 6040_h （0380_h） is set to 1 when updating immediately)
（3）	After receiving the rising edge of bit4 at 6040_h（0380_h）, the slave makes a judgment on whether the new displacement command can be received: if bit12 at 6041_h （0381_h）is 0 at this time, it indicates that the slave can receive the new displacement command ①; the slave receives the new After the displacement command, set bit 12 of 6041_h （0381_h）from 0 to 1, indicating that the new displacement command ①

	has been received and the current slave station is in a state where it cannot continue to receive the new displacement command. In the immediate update mode, once a new displacement command is received (bit 12 of 6041_h（0381_h）changes from 0 to 1), the motor immediately executes the displacement command
（4）	After the host computer receives the slave station status word 6041_h（0381_h） bit12 becomes 1, it can release the displacement command data, and set the control word 6040_h（0380_h） bit4 from 1 to 0, indicating that there is no new position command
（5）	When the slave detects that bit 4 of control word 6040h changes from 1 to 0, it can set bit 12 of status word 6041h from 1 to 0, indicating that the slave is ready to receive a new displacement command.

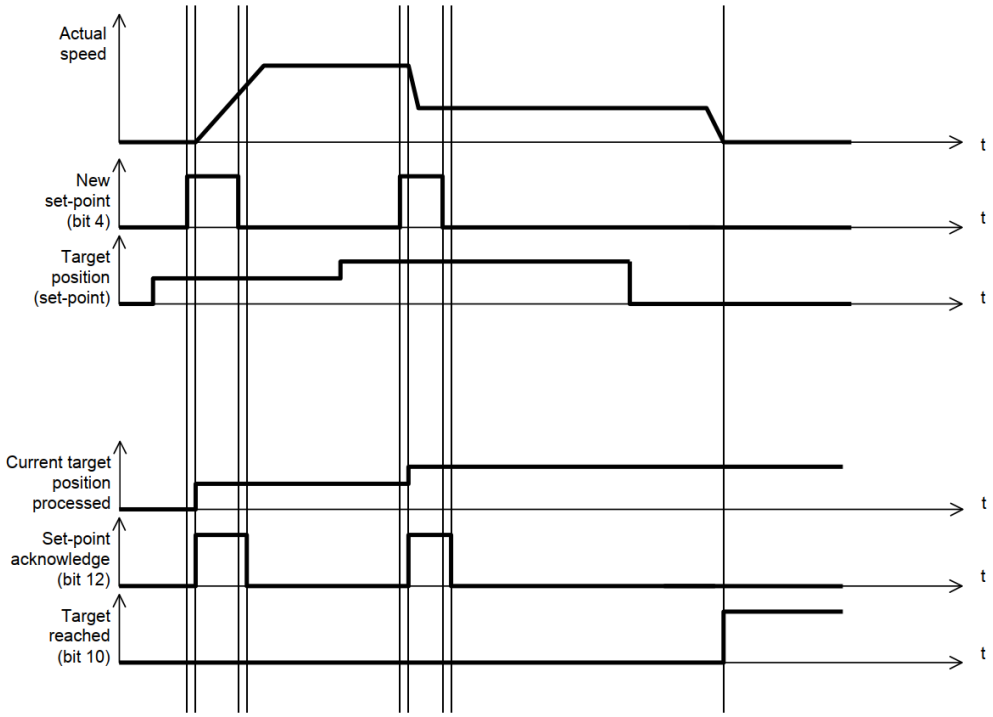


Figure 5-10

2） Control command timing-not immediately updated

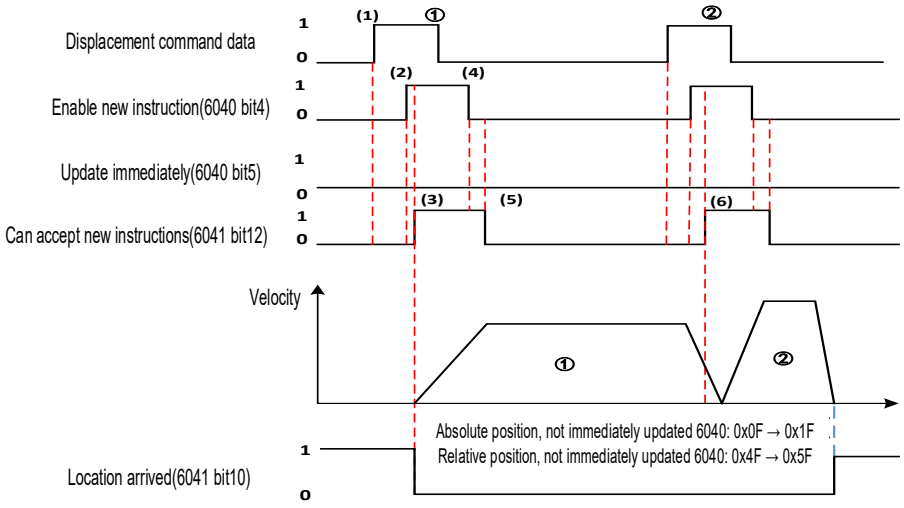


Figure 5-11

In the case of non-immediate update, the annotations (1)-(5) in the figure are similar to the interpretation in the case of immediate update. You can see from the position (6) in the figure. The difference is: in the case of non-immediate update, although the new Target position ②, but need to reach position ① first, decelerate to stop, and then move to position ②.

In the case of non-immediate update, the motor has designed 5 buffers, that is, there can be 5 target positions in the sequence at the same time, and the motor will travel to these positions in sequence. If the cache is full, the new location cannot be accepted, and the new location cannot be received until the cache is empty. These caches are cleared when the motor is stopped.

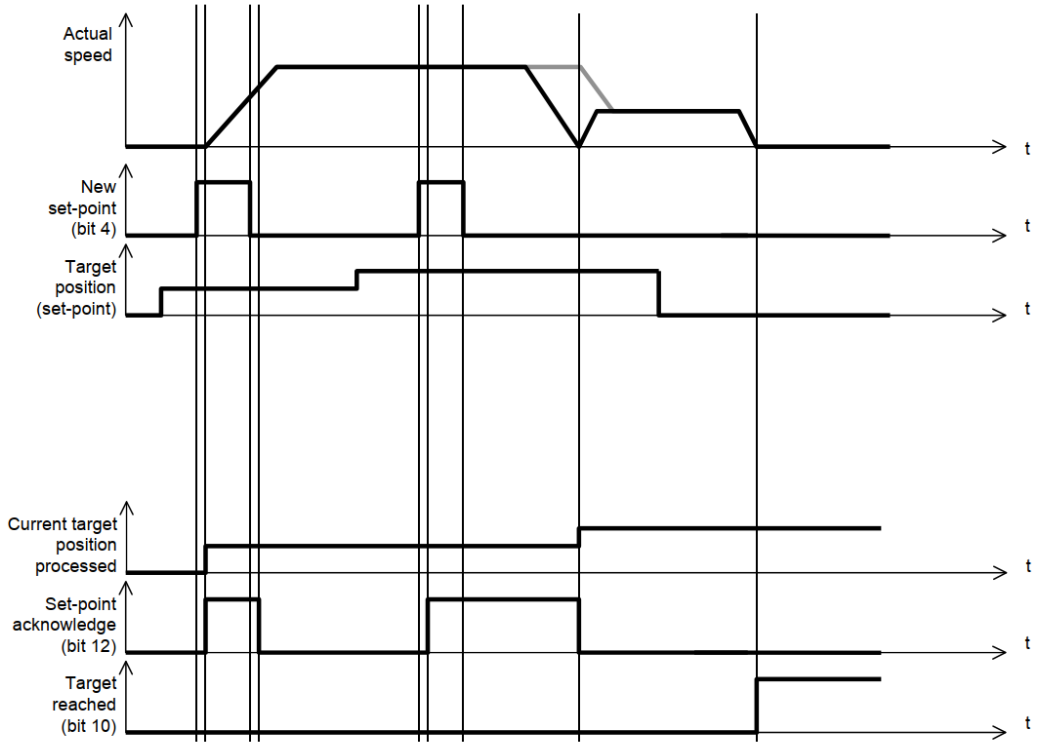


Figure 5-12

5.3.5 Shock Restricted and Unrestricted Modes

In principle, the modes can be divided into "impact limited" and "impact unrestricted" according to impact

- **Shock restricted mode**
  Set object $6086_h(0402_h)$ to "3" to perform impact limited positioning operation. At this time, subject $60A4_h$: 01_h~$04_h(041E_h$~$0424_h)$ items related to impact take effect

- **Unlimited shock mode**
  In this mode, there is no impact limit on each position of the characteristic curve. To run the "shock unrestricted" ramp: Set the item in object $6086_h(0402_h)$ to "0".

5.3.6 Configuration examples

- Configuration mode:
  2002h: 01h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x01, make the device work in profile position mode;
  Send：01 10 00 B1 00 01 02 00 00 (set to CiA402 mode)

  Send: 01 10 03 C2 00 01 02 00 01 (set to contour position mode)

- Parameter configuration:
  Write target location $607A_h$ （$03E7_h$）(user unit);
  Write the current displacement command at a constant speed of 6081_h ($03F8_h$）(user unit/s);
  Set the acceleration of displacement 6083_h （$03FC_h$）(user unit/s2) and deceleration 6084_h （$03FE_h$）(user unit/s2);
  Send: 01 10 03 E7 00 02 04 00 00 27 10 (Set target position $607A_h$ ($03E7_h$)to 10000)

  Send: 01 10 03 F8 00 02 04 00 00 27 10 (Set target speed $6081_h(03F8_h)$ to 10000)

  Send: 01 10 03 FC 00 02 04 00 00 9C 40 (Set acceleration 6083_h ($03FC_h$)to 40000)

  Send: 01 10 03 FE 00 02 04 00 00 9C 40 (Set deceleration 6084_h ($03FE_h$)to 40000)

- Write control word
  6040_h (0380_h)= 0x06→0x07→ 0x(n)F → 0x(n+1)F, the motor is running,
  Send: 01 10 03 80 00 01 02 00 06（Set 6040_h (0380_h)to 0x6, Ready to make motor）

  Send: 01 10 03 80 00 01 02 00 07（Set 6040_h (0380_h)to 0x7, Disable the motor）

  Send: 01 10 03 80 00 01 02 00 0F（Set 6040_h (0380_h)to 0xF, Enable the motor）
  The different instruction types are shown in the table below:

Table 5-11

6040_h（0380_h）-bit6	**6040_h（0380_h）-bit5**	**6040_h（0380_h）Change**	**Description**
0	0	0x0F → 0x1F	Absolute position, not immediately updated
0	1	0x2F → 0x3F	Absolute position, update immediately
1	0	0x4F → 0x5F	Relative position, not updated immediately
1	1	0x6F → 0x7F	Relative position, update immediately

Send: 01 10 03 80 00 01 02 00 1F(Set 6040_h （0380_h）to 0x1F, Absolute position, not immediately updated)
Send: 01 10 03 80 00 01 02 00 3F(Set 6040_h（0380_h） to 0x3F, Absolute position, update immediately)
Send: 01 10 03 80 00 01 02 00 5F(Set 6040_h（0380_h） to 0x5F, Relative position, not immediately updated)
Send: 01 10 03 80 00 01 02 00 7F(Set 6040_h（0380_h） to 0x7F, Relative position, update immediately)

- Monitoring parameters:
  Actual position feedback: 6063_h ($03C6_h$）(encoder unit), 6064_h（$03C8_h$） (user unit)

5.4 Velocity Mode (VM)

5.4.1 Structure Diagram

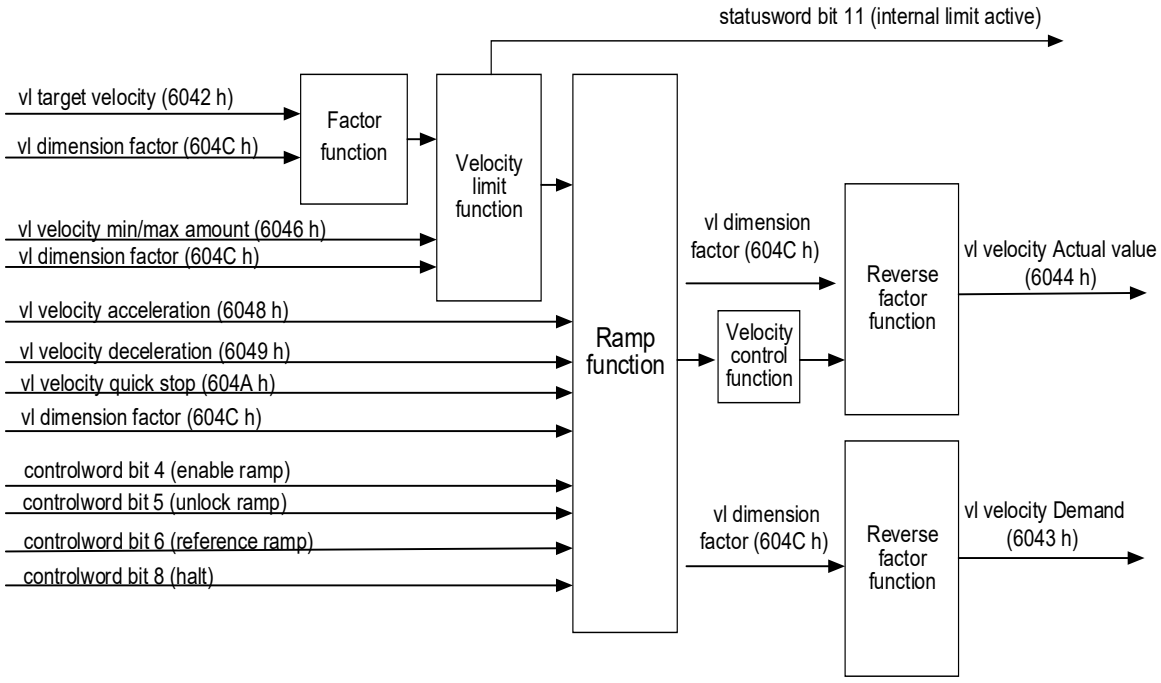


Figure 5-13

5.4.2 Related Objects

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

Table 5-12

Object Index	Description
6042_h（0382_h）	Target speed in VM mode, (default unit: rpm, related to $604C_h$（0394_h和 0396_h）)
6043_h（0383_h）	Target speed command in VM mode (default unit: rpm)
6044_h (0384_h)	Current actual speed feedback value (rpm)
$604C_h$：01_h（0394_h）and $604C_h$：02_h (0396_h)	Scaling factor for the speed unit of the VM mode. The default speed unit is rpm (rev/min). $604C_h$：01_h(0394_h) contains the numerator (multiplier) used to calculate the speed, and $604C_h$：02_h (0396_h) contains the denominator (divisor). Speed unit: $rpm * \frac{604C_h: 01_h(0394_h)}{604C_h: 02_h(0396_h)}$, For example, if $604C_h$：01_h(0394_h) is set to 2 and $604C_h$：02_h (0396_h) is set to 1, the speed unit is 2 rpm. If either of $604C_h$：01_h(0394_h) and $604C_h$：02_h (0396_h) is 0, the speed unit is rpm.
6048_h：01_h（0389_h）and 6048_h：02_h($038B_h$)	Acceleration in VM mode. 6048_h：01_h（0389_h） contains the speed change (in rpm), and 6048_h：02_h($038B_h$) contains the corresponding time (in seconds). The two together calculate the acceleration, the specific formula is as follows: VI velocity acceleration =Delta speed (6048_h：01_h（0389_h）) / Delta time (6048_h：02_h($038B_h$))) For example, if you need to accelerate the motor to 300rpm in 3.5s, configure 6048_h: 01_h（0389_h）=3000, 6048_h: 02_h($038B_h$)=35.
6049_h：01_h（$038C_h$）and 6049_h：02_h($038E_h$)	The deceleration of VM mode. 6049_h：01_h（$038C_h$） contains the speed change (unit: rpm), and 6049_h：02_h($038E_h$) contains the corresponding time (unit: s (second)). The two together calculate the acceleration, the specific formula is as follows: VI velocity deceleration = Delta speed (6049_h：01_h（$038C_h$）) / Delta time (6049_h：02_h($038E_h$))
$604A_h$：01_h（$038F_h$）and $604A_h$：02_h(0391_h)	The deceleration of VM mode during rapid shutdown. $604A_h$：01_h（$038F_h$）contains the speed change (unit: rpm), and $604A_h$：02_h(0391_h) contains the corresponding time (unit: s (second)). The two together calculate the acceleration, the specific formula is as follows: VI Quick stop deceleration = Delta speed ($604A_h$：01_h（$038F_h$）) / Delta time ($604A_h$：02_h(0391_h))
6046_h：01_h（0385_h）and 6046_h：02_h(0387_h)	Speed limit in VM mode (default unit: rpm, related to $604C_h$：01_h（0394_h）and $604C_h$：02_h (0396_h)), The specific description is as follows: 6046h: 01_h（0385_h） sets the minimum speed. If the target speed (6042h（0382_h）) is lower than the minimum speed, its value will be set to the minimum speed of 6046_h: 01_h（0385_h）. 6046_h: 02_h(0387_h) sets the maximum speed. If the target speed (6042_h（0382_h）) is higher than the maximum speed, the value will be set to the maximum speed of 6046h: 02_h(0387_h).

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

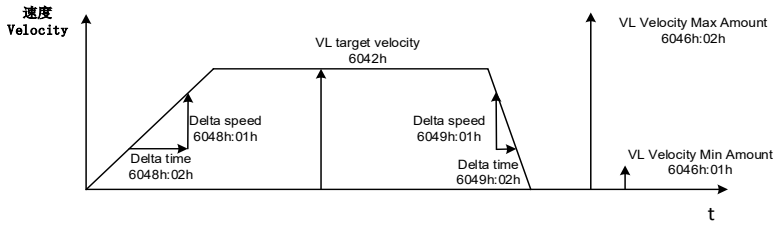


Figure 5-14

5.4.3 Control Commands and Status Information

Enable

To enable this mode, the value must be set to "0" in object 2002_h: 01_h（$00B1_h$） and "2" in object 6060_h ($03C2_h$) (Modes Of Operation).

Control word

15 9	8	7	6	5	4	3 0
(see 3.1.2)	Halt	(see 3.1.2)	Reference ramp	Unlock ramp	Enable ramp	(see 3.1.2)
MSB						LSB

In Velocity mode, the following bits in object 6040_h (0380_h) (Control word) have special functions

Table 5-13

Bit	Value	Description
2	0	When a quick stop is triggered, the motor will perform rapid braking at the rapid stop deceleration set in object $604A_h$. Subsequently, the controller switches to the "Switch on disabled" state
	1	No action
4	0	Without acceleration and deceleration, the output changes to the given speed immediately
	1	Adjust the speed according to the acceleration and deceleration settings
5	0	No longer follow the output of the planner for speed regulation, the speed output value should be locked to the current speed value
	1	Adjust the speed according to the speed output of the planner
6	0	Speed curve planner input is 0
	1	The speed curve planner input is the given target speed
8	0	No action
	1	Motor pause

Status word

15	14	13	12	11	10	9	0

(see 3.1.3)	reserved	(see3.1.3)	reserved	(see 3.1.3)

MSB LSB

In the Velocity mode, the bits in the object 6041_h (0381_h) have no special functions. For the definition of the general bits, see section 3.1.3 Status Sub word.

5.4.4 Functional Description

In this mode, the motor runs at the preset value of the target speed, similar to the inverter. Unlike profile Velocity mode, this mode does not allow the selection of ramps with limited impact.

5.4.5 Configuration Examples

- Configuration mode:
  Write 2002_h：01_h（$00B1_h$）=0、6060_h（$03C2_h$）=0x02, configured as Velocity mode (VM)
  Send: 01 10 00 B1 00 01 02 00 00 (Set to CiA402 mode)
  Send: 01 10 03 C2 00 01 02 00 02 (Set to Contour position mode)
- Parameter configuration:
  Write target speed: 6042_h（0382_h）=1000;
  Write acceleration and deceleration:
  6048_h：01_h（0389_h）= 500，6048_h：02_h（$038B_h$）= 1;
  6049_h：01_h（$038C_h$）= 500，6049_h：02_h（$038E_h$）= 1；
  Send: 01 10 03 89 00 02 04 00 00 01 F4 (set 6048_h:01_h （0389_h） to 500)
  Send: 01 10 03 8B 00 01 02 00 01 (set 6048_h:02_h （$038B_h$） to 1)
  Send: 01 10 03 8C 00 02 04 00 00 01 F4 (set 6049_h:01_h（$038C_h$） to 500)
  Send: 09 10 03 8E 00 01 02 00 01 (set 6049_h:02_h （$038E_h$）to 1)
- Write control word:
  6040_h（0380_h） = 0x06→0x07→0x7F, the motor is running;
  Send: 01 10 03 80 00 01 02 00 06（Set 6040_h （0380_h）to 0x6, Ready to make motor）
  Send: 01 10 03 80 00 01 02 00 07（Set 6040_h （0380_h）to 0x7, Disable the motor）
  Send: 01 10 03 80 00 01 02 00 7F（Set 6040_h （0380_h）to 0x7F, Enable the motor）
- Parameter monitoring:
  Current actual speed $606C_h$（$03D5_h$） (unit rpm)

5.5 Profile Velocity Mode (PV)

In this mode, the host controller sends the target speed, acceleration, and deceleration to the drive, and the speed and torque adjustments are performed internally.

5.5.1 Structure Diagram

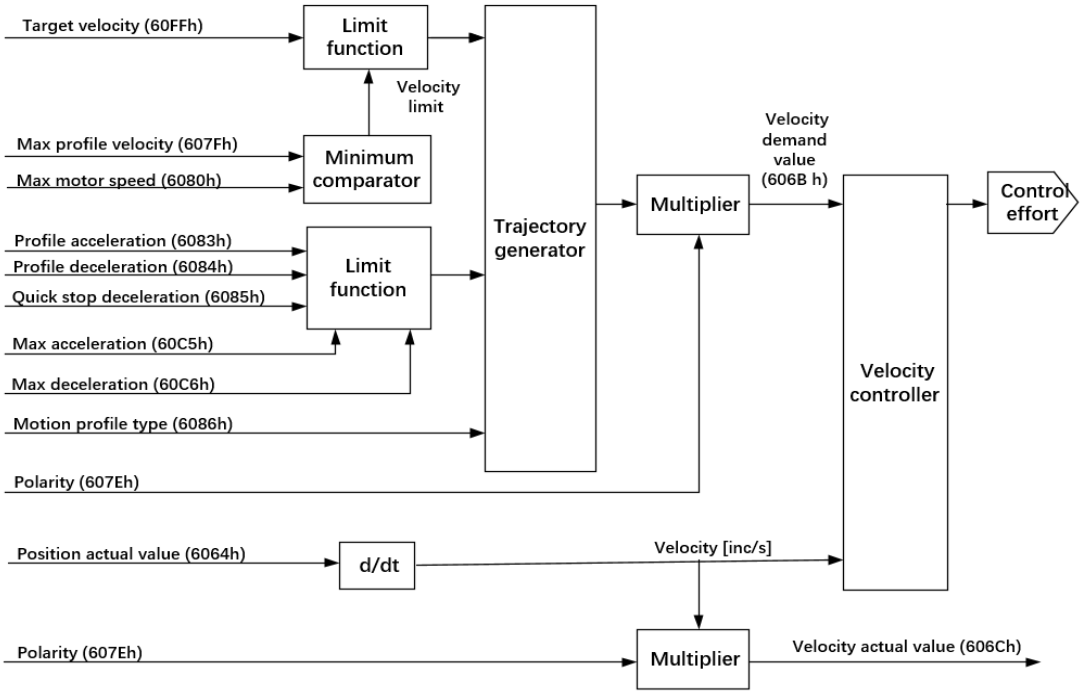


Figure 5-15

5.5.2 Related objects

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

Table 5-14

Object index	Description
6063_h（$03C6_h$）	Motor current absolute position feedback (encoder unit)
6064_h（$03C8_h$）	Motor current user absolute position feedback (user unit)
$606B_h$（$03D3_h$）	Speed command value valid inside the controller
$606C_h$（$03D5_h$）	Current actual speed feedback value (rpm)
$607E_h$（$03F3_h$）	Direction of rotation (polarity), see "4.3 607Eh: Polarity" for details
$607F_h$（$03F4_h$）	The maximum profile speed (user units/s) during operation plays a role in limiting the speed.
6080_h（$03F6_h$）	Maximum speed of motor (rpm)
6083_h（$03FC_h$）	The profile acceleration (user units/s2) during operation is limited by $60C5_h$.
6084_h（$03FE_h$）	The profile deceleration (user units/s2) during operation is limited by $60C6_h$.
6085_h（0400_h）	Stop deceleration when executing "fast stop" (user unit/s2)
6086_h（0402_h）	Select the type of curve or slope, If the value is "0", the impact (acceleration) will not be limited, that is, the trapezoid curve;

Object index	Description
	If the value is "3", the value in 60A4h: 01h-02h will be used to limit the impact (acceleration), that is, the S-curve, the device only uses the 01h and 02h indexes.
$60C5_h$（$043B_h$）	Maximum acceleration limit (user units/s^2)
$60C6_h$（$043D_h$）	Maximum deceleration limit (user units/s^2)
$60FF_h$（0448_h）	Target speed (user units/s)

5.5.3 Control Commands and Status Messages

Enable

To enable this mode, you must set the value to "0" in object 2002h: 01_h($00B1_h$) and the value to "3" in object 6060_h ($03C2_h$).

Control word

15 9	8	7	6 4	3 0
(see 3.1.2)	Halt	(see 3.1.2)	reserved	(see 3.1.2)
MSB				LSB

In profile Velocity mode, the following bits in object 6040_h (0380_h) have special functions

Table 5-15

Bit	Value	Meaning
8	0	The motor continues to run
	1	suspend

Status word

15	14 13	12	11	10	9 0
(see 3.1.3)	-	Speed	(see 3.1.3)	Target reached	(see 3.1.3)
MSB					LSB

In profile Velocity mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-16

Bit	Value	Meaning
10	0	In non-paused state (bit8 = 6040_h(0380_h)): The target speed has not been reached In the pause state (bit8 = 1 at 6040_h(0380_h)): the motor decelerates
	1	In non-pause state (bit8 = 6040_h(0380_h)): target speed reached In the pause state (bit8 = 1 in 6040_h(0380_h)): the motor speed is 0
12	0	Speed is not 0
	1	Speed is 0

5.5.4 Functional description

In this mode, the host controller sends the target speed, acceleration, and deceleration to the drive, and the speed and torque adjustments are performed internally.

5.5.5 Configuration examples

- Configuration mode
  Write 2002_h：01_h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x03, make it work in profile Velocity mode;
  Send: 01 10 00 B1 00 01 02 00 00(Set to CiA402 mode)
  Send: 01 10 03 C2 00 01 02 00 03(Set to Contour speed mode)

- Configuration parameters:
  Write target speed: $60FF_h$（0448_h）= 50000;
  Write profile acceleration: 6083_h（$03FC_h$）= 40000;
  Write profile deceleration: 6084_h（$03FE_h$）= 40000;
  Send: 01 10 04 48 00 02 04 00 00 C3 50(Set the target speed of $60FF_h$（0448_h） to 50000. When enabled, the motor runs at 300rpm)
  Send: 01 10 04 48 00 02 04 FF FF 3C B0 (Set the target speed of $60FF_h$（0448_h） to -50000. When enabled, the motor runs at -300rpm)
  Send: 01 10 03 FC 00 02 04 00 00 9C 40(Set acceleration 6083_h （$03FC_h$）to 40000)
  Send: 01 10 03 FE 00 02 04 00 00 9C 40(Set deceleration 6084_h （$03FE_h$）to 40000)

- Write control word:
  6040_h（0380_h）=0x06→0x07→0x0F, the motor is running;
  Send: 01 10 03 80 00 01 02 00 06（Set 6040_h to 0x6, Ready to make motor）
  Send: 01 10 03 80 00 01 02 00 07（Set 6040_h to 0x7, Disable the motor）
  Send: 01 10 03 80 00 01 02 00 0F（Set 6040_h to 0xF, Enable the motor）
- Parameter monitoring:
  Current actual speed $606C_h$（$03D5_h$） (unit: rpm)

5.6 Profile Torque Mode (PT)

In this mode, the host controller sends the target torque 6071_h （$03DB_h$）and the torque ramp constant 6087_h （0403_h） to the drive. The torque adjustment is performed internally. When the speed reaches the limit value, it will enter the speed regulation phase.

5.6.1 Structure diagram

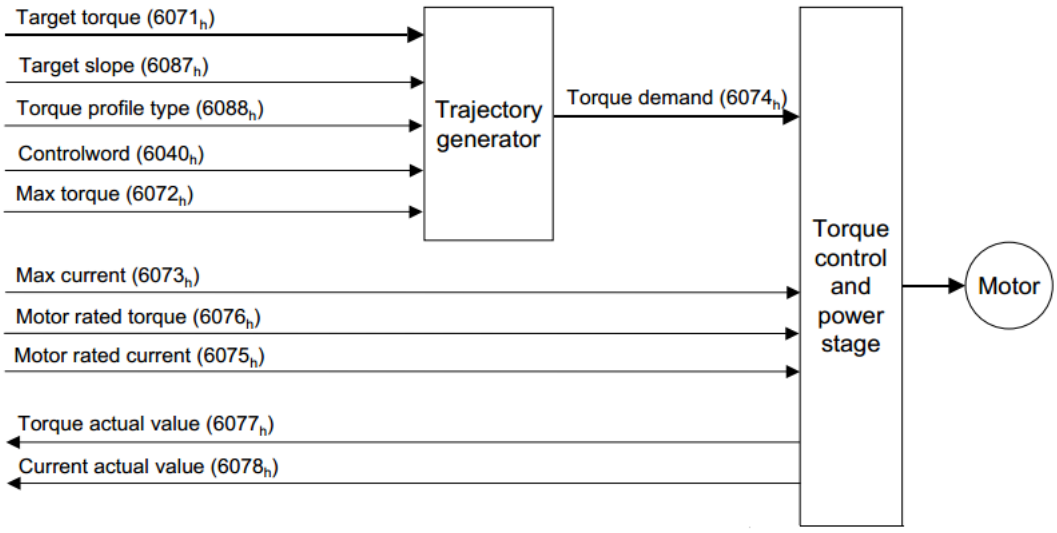


Figure 5-16 Control block diagram

5.6.2 Related objects

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

Table 5-17

Object Index	Description
$605D_h$ ($03C0_h$)	Pause mode selection
$605A_h$ ($03BF_h$)	Quick stop mode selection
6071_h ($03DB_h$)	Set target torque (unit: 0.1%)
6072_h ($03DC_h$)	Maximum torque (unit: 0.1%) on the entire ramp (acceleration, saved torque, braking)
2007_h：10_h($016D_h$)	Torque control forward speed limit (rpm)
2007_h：11_h($016E_h$)	Torque control reverse speed limit (rpm)
6087_h (0403_h)	Maximum torque change per second (unit: 0.1%)
6088_h (0405_h)	Torque ramp type (0-ramp, 2-none)
6077_h ($03E3_h$)	Actual torque value (unit: 0.001N.m)

5.6.3 Control Commands and Status Messages

Enable

To enable this mode, you must set the value to "0" in object 2002_h: 01_h （$00B1_h$）and the value to "4" in object 6060_h ($03C2_h$).

Control word

15 9	8	7	6 4	3 0
(see 3.1.2)	Halt	(see 3.1.2)	reserved	(see 3.1.2)

MSB LSB

In profile torque mode, the following bits in object 6040_h (0380_h) have special functions

Table 5-18

Bit	Value	Meaning
8	0	The motor continues to run
	1	suspend

Status word

15 14	13 12	11	10	9 0
(see 3.1.3)	reserved	(see 3.1.3)	Target reached	(see 3.1.3)

MSB LSB

In profile torque mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-19

Bit	Value	Meaning
10	0	In non-pause state (bit8 = 6040_h(0380_h)): target torque is not reached In the pause state (bit8 = 1 at 6040_h(0380_h)): the motor decelerates
	1	In non-pause state (bit8 = 6040_h(0380_h)): target torque reached In the pause state (bit8 = 1 in 6040_h(0380_h)): the motor speed is 0

5.6.4 Functional Description

Speed limit function

When running the integrated low-voltage brushless motor in profile torque mode, when the motor's operating speed exceeds the value set by the speed limit (forward speed limit value 2007_h: 10_h($016D_h$) and reverse speed limit value 2007_h: 11_h($016E_h$)), it automatically turns Control the Velocity mode and control the current speed within the speed limit. When it is detected that the given target torque is less than the average torque of the current speed, the motor exits the speed control and returns to the torque mode control.

5.6.5 Configuration Examples

- Configuration mode
  Write 2002_h：01_h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x04, make it work in profile torque mode;
  Send: 01 10 00 B1 00 01 02 00 00(Set to CiA402 mode)
  Send: 01 10 03 C2 00 01 02 00 04(Set to contour torque mode)
- Configuration parameters:
  Write target torque: 6071_h($03DB_h$) = 500 (unit: 0.1%);
  Write torque ramp type 6088_h(0405_h)=0;
  Write torque ramp 6087_h(0403_h)=10;

Send: 01 10 03 DB 00 01 02 01 F4(Set 6071_h($03DB_h$) to 500 and the motor runs at 50% rated torque)；

Send: 01 10 04 05 00 01 02 00 00(Set slope type 6088_h(0405_h) to slope)

Send: 01 10 04 03 00 02 04 00 00 00 0A(Set the torque ramp 6087_h (0403_h)to 10)

- Write control word:

  6040_h（0380_h）=0x06→0x07→0x0F，motor running;

  6040_h（0380_h）=0x06→0x07→0x0F, the motor is running;

  Send:01 10 03 80 00 01 02 00 06（set 6040_h (0380_h）to 0x6 to make motor ready）

  Send:01 10 03 80 00 01 02 00 07（set 6040_h （0380_h）to 0x7, Disable the motor）

  Send:01 10 03 80 00 01 02 00 0F（set 6040_h（0380_h） to 0xF, Enable the motor）

- Parameter monitoring:

  Actual torque 6077_h（$03E3_h$）(unit: 0.001N.m)

5.7 Home Mode (HM)

The homing mode is used to move from the current position to the origin position of the device. During the

movement, the maximum acceleration, minimum deceleration, maximum speed, minimum speed, etc. are taken into account.

5.7.1 Structure diagram

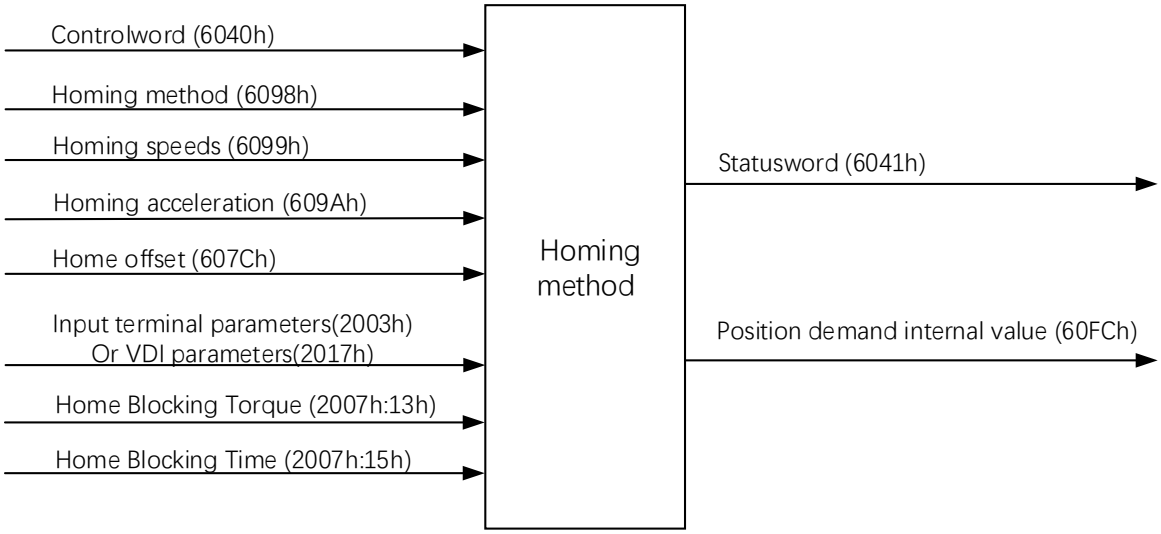


Figure 5-17

5.7.2 Related Objects

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

Table 5-20

Object Index	Description
$607C_h$ ($03ED_h$)	Specific description: Set the physical position where the mechanical zero point deviates from the motor origin in the homing mode; when the homing operation is completed, bit 15=1 of status word 6041_h(0381_h) becomes effective
6098_h (0416_h)	Origin return method (see "5.7.4 Function description")
6099_h:01_h (0417_h) 6099_h:02_h (0419_h)	Operating speed (user units/s) Sub-index: 6099_h: 01_h(0417_h): Find the speed of the switch; 6099_h: 02_h(0419_h): the speed of finding the origin
$609A_h$ ($041B_h$)	Acceleration of homing operation (user unit/s^2)
$60FC_h$（0446_h）	The output of the trajectory generator is the internally planned real-time position command (encoder unit)
6063_h ($03C6_h$)	Motor current absolute position feedback (encoder unit)
2003_h ($00D5_h$~$00E6_h$)	Physical input terminal function configuration (see "6.1.1 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 fault will be reported
2017_h (0326_h~0345_h)	Virtual terminal input function configuration (see: "6.1.1 Digital Input" for details)
2007h:13_h (0170_h)	The detected torque when the locked position is found
2007h:15_h (0172_h)	Detection time when blocking to find the origin

Examples of speed meaning

The following figure shows the speed change when using the homing method 18:

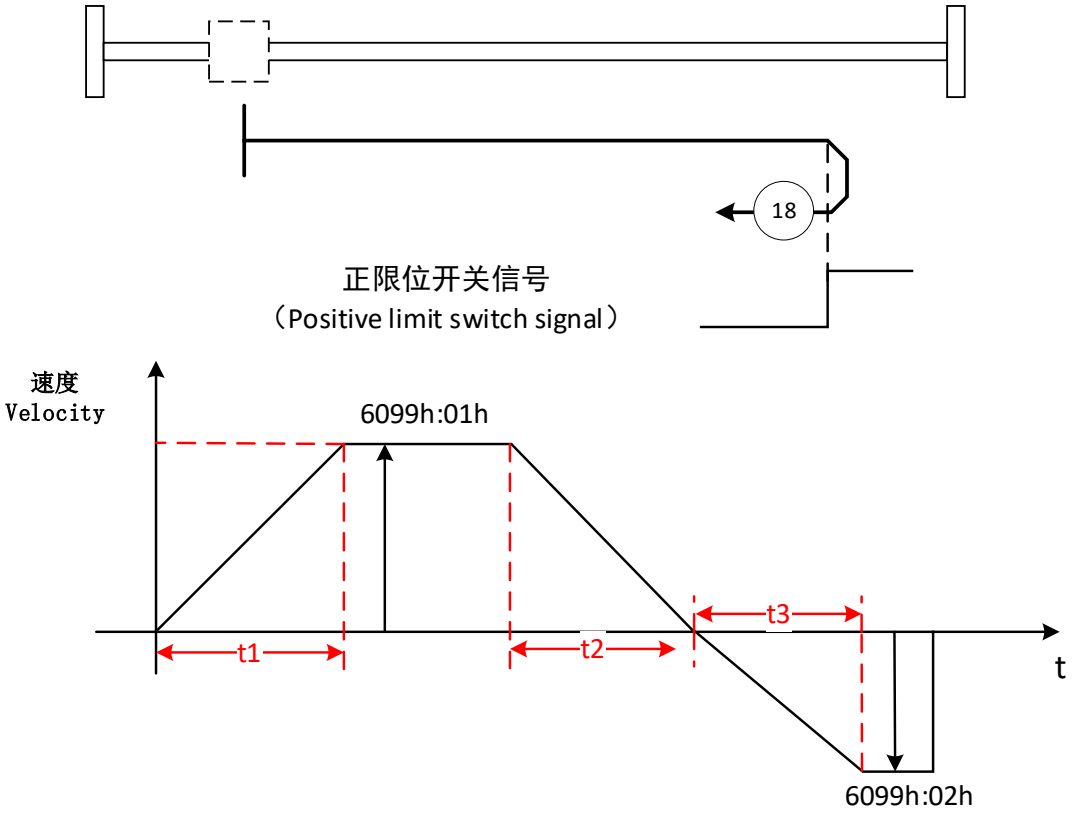


Figure 5-18

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

5.7.3 Control Commands and Status Messages

Enable

To enable this mode, you must set the value to "0" in object 2002_h:01_h（$00B1_h$） and the value to "6" in object 6060_h ($03C2_h$), and configure the function of the origin switch or limit switch for the multi-function terminals. For multi-function terminal configuration, please refer to Chapter 6.1.

Control word

15 9	8	7	6 5	4	3 0
(see 3.1.2)	Halt	(see 3.1.2)	reserved (0)	Homing operation start	(see 3.1.2)
MSB					LSB

In the homing mode, the following bits in object 6040_h (0380_h) have special functions

Table 5-21

Bit	Value	Meaning
4	0	The homing mode is not activated
	1	Start or continue homing mode

8	0	The motor decides whether to start homing according to bit4 setting
	1	The motor is suspended according to the setting of $605D_h$（$03C0_h$）

Status word

15	14	13	12	11	10	9	0
(see 3.1.3)		Homing error	Homing attained	(see 3.1.3)	Target reached	(see 3.1.3)	
MSB							LSB

In the homing mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-22

Bit	Value	Meaning
10	0	Target position not reached
	1	Target position reached
12	0	Return to origin is not complete
	1	Return to origin completed

5.7.4 Functional Description

5.7.4.1 Switch find origin method

The way of using the switch to return to the origin, currently supports 17~30 in CiA402. After returning 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.

DI function number configuration of input terminal related to HM mode:

Table 5-23

Function Number	Function Definition
14	Positive limit switch
15	Negative limit switch
31	Home switch

The directions and switch states that require homing are shown in the table below:

Table 5-24

Homing Method6098_h(0416_h)	Switch Related
17	Negative limit switch
18	Positive limit switch
19	Home switch
20	Home switch
21	Home switch
22	Home switch
23	Home switch, positive limit switch

Homing Method6098_h(0416_h)	Switch Related
24	Home switch, positive limit switch
25	Home switch, positive limit switch
26	Home switch, positive limit switch
27	Home switch, negative limit switch
28	Home switch, negative limit switch
29	Home switch, negative limit switch
30	Home switch, negative limit switch

- Homing method 17:

Mechanical origin: negative limit switch

When the negative limit switch is invalid, it starts to return to zero at the high speed in the reverse direction. When the negative limit switch is valid (changes along the edge), it decelerates, reverses, and runs at a low speed in the positive direction. When the negative limit switch is invalid (change along the edge) to stop.

When the negative limit switch is valid, it starts to return to zero at low speed directly, and stops when the negative limit switch is invalid (change along the edge).

The motion trajectory is as follows:

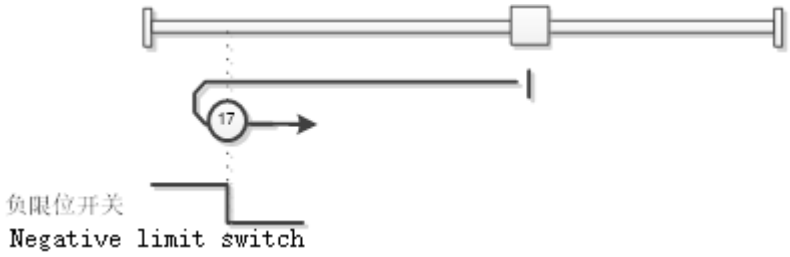


Figure 5-19

- Homing method 18:

Mechanical origin: positive limit switch

When the positive limit switch is invalid, it starts to return to zero at the high speed of the positive direction. When the positive limit switch is valid (changes along the edge), it decelerates, reverses, and runs at a low speed in the reverse direction. When the positive limit switch is invalid (change along the edge) to stop.

When the positive limit switch is valid, it starts to return to zero at low speed directly, and stops when the positive limit switch is invalid (change along the edge).

The motion trajectory is as follows:

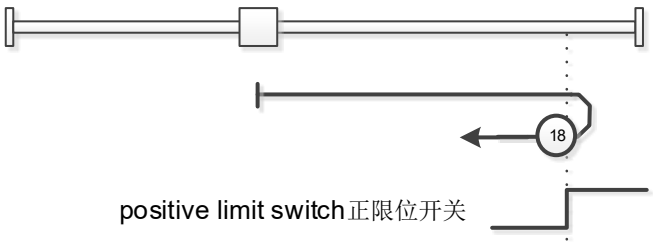


Figure 5-20

- Homing method 19 and 20:

Mechanical origin: origin switch

Homing method 19:

When the home switch is invalid, it starts to return to zero at the high speed in the forward direction.

When the home switch is encountered when it is valid (change along the edge), it decelerates, reverses, and runs at a low speed in the reverse direction, when the home switch is invalid (change along the edge) Stop; when the home switch is valid, it will directly return to zero at low speed, and stop when the home switch is invalid (change along the edge).

Homing method 20:

When the origin switch is invalid, it starts to return to zero at the positive low speed directly, and stops when the origin switch is valid (change along the edge); when the origin switch is valid, it returns to zero at a high speed in the reverse direction, and the origin switch is encountered as When it is invalid (change along the edge), it decelerates, reverses, and runs at a low speed in the forward direction. It stops when it encounters the origin switch is effective (change along the edge).

The motion trajectory is as follows:

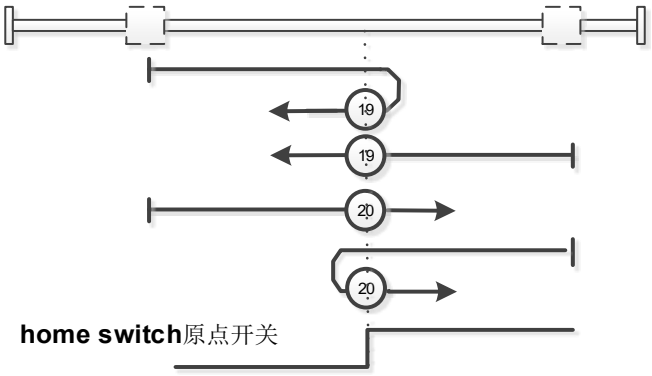


Figure 5-21

- Homing method 21 and 22:

Mechanical origin: origin switch

Homing method 21:

When the home switch is invalid, it starts to return to zero at a high speed in the reverse direction. When the home switch is encountered, it is valid (change along the edge), deceleration, reverse, forward low speed operation, when the home switch is invalid (change along the edge) Stop; when the home switch is valid, it will directly return to zero at low speed in the positive direction, and stop when the home switch is invalid (change along the edge).

Homing method 22:

When the origin switch is invalid, it starts to return to zero at low speed directly, and stops when the origin switch is valid (along the change); when the origin switch is valid, it starts to return to zero at a high speed in the forward direction, and the origin switch is encountered as When it is invalid (change along the edge), it decelerates, reverses, and runs at a low speed in the reverse direction, and stops when it encounters the origin switch is effective (change along the edge).

The motion trajectory is as follows:

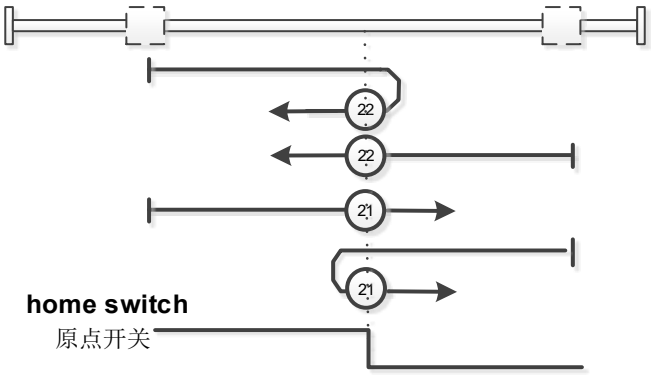


Figure 5-22

- Homing method 23 - 26:

  Mechanical origin: origin switch

  In these ways, the motor is first moved in one direction to scan the origin switch. Only when the origin switch is in the active state is the search trajectory shorter.

  Homing method 23:

  a. When the origin switch is invalid, when the positive limit switch is not encountered, return to zero at the high speed in the forward direction, when the origin switch is valid (change along the edge), decelerate, reverse, reverse low speed operation, encounter the origin Stop when the switch is invalid (change along the edge);

  b. When the origin switch is invalid, it starts to return to zero at a high speed in the forward direction. When it encounters a positive limit switch, reverse, reverse high speed operation. when the origin switch is valid (change along the edge), decelerate, continue to reverse low speed operation, encounter the origin Stop when the switch is invalid (change along the edge);

  c. When the origin switch is valid, it will directly return to zero at low speed, and stop when it is invalid (change along the edge).

  Homing method 24:

  a. When the origin switch is invalid, when the positive limit switch is not encountered, return to zero at the high speed in the forward direction, when the origin switch is valid (change along the edge), decelerate, reverse, reverse low speed operation, encounter the origin When the switch is invalid (change along the edge), reverse, forward low-speed operation, stop when the origin switch is valid (change along the edge);

  b. When the origin switch is invalid, it starts to return to zero at a high speed in the forward direction. When it encounters the positive limit switch, it runs in reverse and reverse high speeds. When the origin switch is valid (changes along the edge), it decelerates and runs in reverse low speed. When the origin switch is invalid (change along the edge), it runs in reverse and forward low speed, and stops when the origin switch is valid (change along the edge);

  c. When the home switch is valid, the reverse low speed starts to return to zero directly. When the home switch is invalid (change along the edge), the reverse and forward low speed run, and stops when the origin switch is valid (change along the edge).

Homing method 25:

a. When the origin switch is invalid, when the positive limit switch is not encountered, return to zero at the high speed in the positive direction, when the origin switch is valid (changes along the edge), decelerate, run at low speed in the forward direction, when the origin switch is invalid (Edge change), reverse, reverse low speed operation, stop when the origin switch is valid (edge change);

b. When the origin switch is invalid, it starts to return to zero at a high speed in the forward direction. When the positive limit switch is encountered, it runs in reverse and reverse directions. When the origin switch is valid (changes along the edge), it decelerates. When the origin switch is invalid (change along the edge), it runs in reverse and reverse low speed, and stops when the origin switch is valid (change along the edge);

c. When the origin switch is valid, it will directly return to zero at the low speed in the forward direction. When the origin switch is invalid (change along the edge), it runs in reverse and reverse speed, and stops when the origin switch is valid (change along the edge).

Homing method 26:

a. When the origin switch is invalid, when the positive limit switch is not encountered, return to zero at the high speed in the positive direction, when the origin switch is valid (changes along the edge), decelerate, run at low speed in the forward direction, when the origin switch is invalid (Change along) stop;

b. When the origin switch is invalid, it starts to return to zero at a high speed in the forward direction. When the positive limit switch is encountered, it runs in reverse and reverse directions. When the origin switch is valid (changes along the edge), it decelerates and runs at low speed in the forward direction. Stop when the origin switch is invalid (change along the edge);

c. When the home switch is valid, it will directly return to zero at low speed in the positive direction, and stop when the home switch is invalid (change along the edge).

The motion trajectory is as follows:

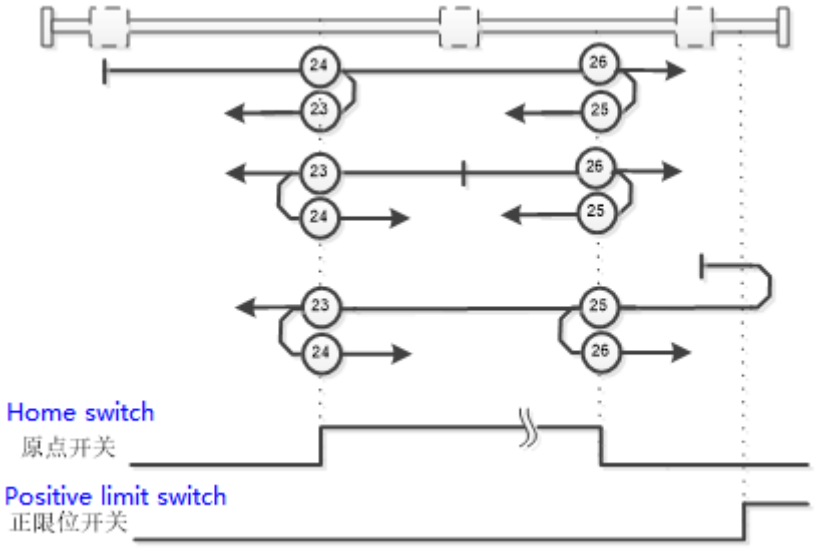


Figure 5-23

- Homing method 27 - 30:
  Mechanical origin: origin switch

  In these methods, the motor is first moved in one direction to scan the origin switch. Only when the origin switch is in the active state is the search trajectory shorter.

  Homing method 27:

  a. When the origin switch is invalid, when the negative limit switch is not encountered, return to zero at high speed in the reverse direction, when the origin switch is valid (change along the edge), decelerate, reverse, run at low speed in the forward direction, encounter the origin Stop when the switch is invalid (change along the edge);

  b. When the origin switch is invalid, it starts to return to zero at the reverse high speed. When it encounters the negative limit switch, it runs in the reverse direction and the forward direction. When the origin switch is valid (changes along the edge), it decelerates and continues the forward low speed operation. Stop when the origin switch is invalid (change along the edge);

  c. When the home switch is valid, it will directly return to zero at low speed in the positive direction, and stop when the home switch is invalid (edge change).

  Homing method 28:

  a. When the origin switch is invalid, when the negative limit switch is not encountered, return to zero at high speed in the reverse direction, when the origin switch is valid (change along the edge), decelerate, reverse, run at low speed in the forward direction, encounter the origin When the switch is invalid (change along the edge), reverse, reverse low speed operation, stop when the origin switch is valid (change along the edge);

  b. When the origin switch is invalid, it starts to return to zero at a high speed in the reverse direction. When the negative limit switch is encountered, it runs in the reverse direction and the forward direction. When the origin switch is valid (changes along the edge), it decelerates and the low speed in the forward direction. When the origin switch is invalid (change along the edge), it runs in reverse and reverse low speed, and stops when the origin switch is valid (change along the edge);

  c. When the origin switch is valid, it will directly return to zero at the low speed in the forward direction. When the origin switch is invalid (change along the edge), it runs in reverse and reverse speed, and stops when the origin switch is valid (change along the edge).

  Homing method 29:

  a. When the origin switch is invalid, when the negative limit switch is not encountered, return to zero at a high speed in the reverse direction, when the origin switch is valid (changes along the edge), deceleration, reverse low speed operation, when the origin switch is invalid (Change along the edge), reverse, forward low speed operation, stop when the origin switch is valid (change along the edge);

  b. When the origin switch is invalid, it starts to return to zero at the reverse high speed. When it encounters the negative limit switch, it runs in the reverse direction and the forward direction. When the origin switch is valid (changes along the edge), it decelerates and runs in the reverse low speed. When the origin switch is invalid (change along the edge), it runs in reverse and forward low speed, and stops when the origin switch is effective (change along the edge);

c. When the home switch is valid, the reverse low speed starts to return to zero directly. When the home switch is invalid (edge change), the reverse and forward low speed run, when the home switch is valid (edge change), stop.

Homing method 30:

a. When the origin switch is invalid, when the negative limit switch is not encountered, return to zero at a high speed in the reverse direction, when the origin switch is valid (changes along the edge), deceleration, reverse low speed operation, when the origin switch is invalid (Change along) stop;

b. When the origin switch is invalid, it starts to return to zero at the reverse high speed. When it encounters the negative limit switch, it runs in the reverse direction and the forward direction. Stop when the origin switch is invalid (change along the edge);

a. When the origin switch is valid, it will directly return to zero at low speed, and stop when it is invalid (edge change).

The motion trajectory is as follows:

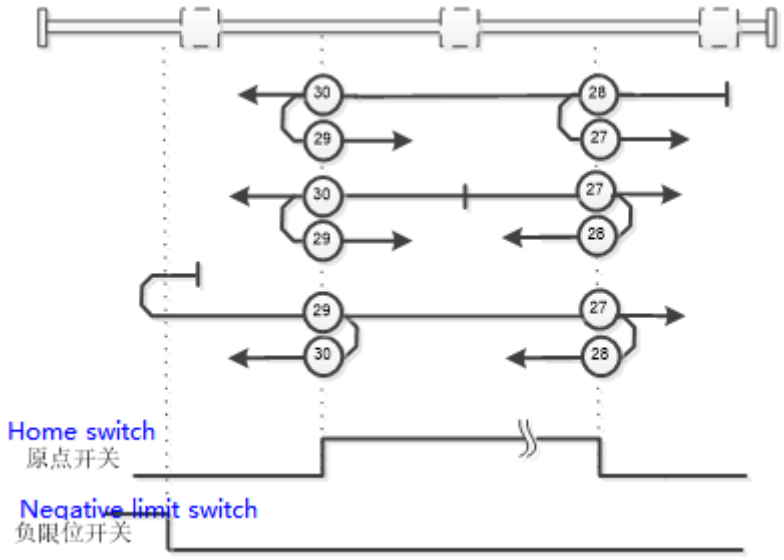


Figure 5-24

5.7.4.2 Finding the origin by blocking rotation

The locked-rotation origin return 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 limit torque is replaced by the value of the detected torque of the locked rotor.

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

Table 5-25

Homing Method 6098_h (0416_h)	Running Direction
37	Run forward at 6099_h:02_h (0419_h) speed
38	Reverse running at 6099_h:02_h (0419_h) speed

In addition, this 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（0170_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（0172_h）

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

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

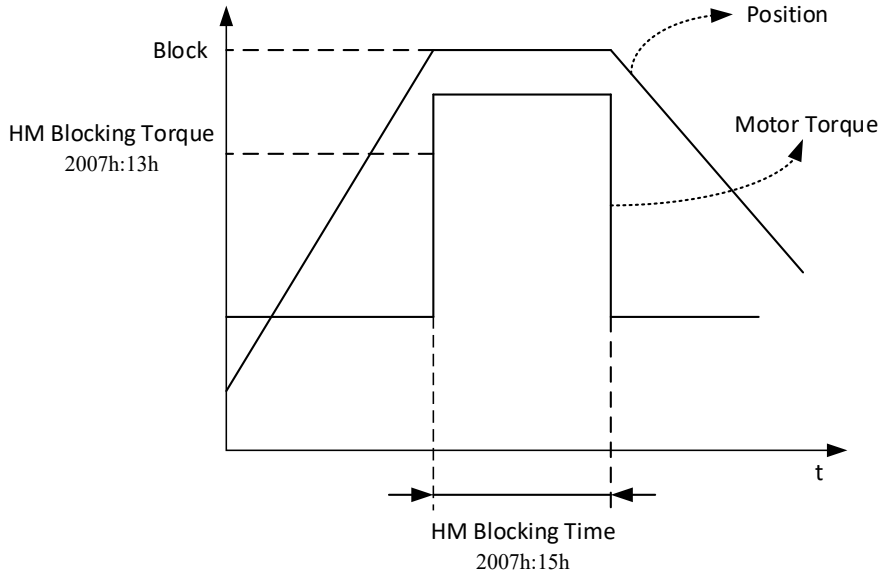


Figure 5-25

5.7.5 Configuration Examples

1. Negative limit switch to find the origin

- Set terminal
  2003_h: 03_h（$00D5_h$）=15 (negative limit switch), 2003_h: 04_h（$00D6_h$）=0 (active low level);
  Send:01 10 00 D5 00 01 02 00 0F（Set DI1 function to negative limit switch）
  Send:01 10 00 D6 00 01 02 00 00（Set DI1 logic to active low level）

- Setting mode
  Write 2002_h: 01_h=0（$00B1_h$）, running mode 6060_h（$03C2_h$）=0x06, make it work in the homing mode;
  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）
  Send:01 10 03 C2 00 01 02 00 06（Set to Home mode）

- Set the home method
  6098_h （0416_h）=17;
  Send:01 10 04 16 00 01 02 00 11（Set the home method to 17）

- Write the speed of searching for limit switch and the speed of searching for origin signal
  6099_h：01_h（0417_h）= 10000(user unit / s)　6099_h：02_h(0419_h)=1000(user unit / s);
  Send:01 10 04 17 00 02 04 00 00 27 10（Set search limit switch speed to 10000）
  Send:01 10 04 19 00 02 04 00 00 03 E8（Set origin signal speed to 1000）

- Set back to zero acceleration
  $609A_h$（$041B_h$）=200000(user unit / s2);

Send:01 10 04 1B 00 02 04 00 03 0D 40（Set the acceleration back to zero to 200000)）

- Write control word

  6040_h（0380_h）= 0x06→0x07→0x0F→0x1F, the motor is running.

  Send:01 10 03 80 00 01 02 00 06（Set 6040_h（0380_h） to 0x6, Ready to make motor）

  Send:01 10 03 80 00 01 02 00 07（Set 6040_h （0380_h）to 0x7, Disable the motor）

  Send:01 10 03 80 00 01 02 00 0F（Set 6040_h （0380_h）to 0xF, Enable the motor）

  Send:01 10 03 80 00 01 02 00 1F（Set 6040_h （0380_h）to 0x1F, the motor runs)

2. Blocking rotation to find the origin

- Setting the blocking detection torque and blocking detection time

  2007_h:13_h（0170_h）=300，2007_h:15_h（0172_h）=50

  Send:01 10 01 70 00 01 02 01 2C（Set the lock-up detection torque to 30%）

  Send:01 10 01 72 00 01 02 00 32（Set the gridlock detection time to 50ms）

- Setting mode

  Write 2002_h: 01_h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x06, make it work in the homing mode;

  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）

  Send:01 10 03 C2 00 01 02 00 06（Set to Home mode）

- Set the origin return method

  6098_h （0416_h）=37;

  Send:01 10 04 16 00 01 02 00 25（Set the origin regression mode to 37）

- Set the speed of finding the origin signal

  6099_h：02_h(0419_h)=1000;

  Send:01 10 04 19 00 02 04 00 00 03 E8（Set the find origin signal speed to 1000）

- Set back to zero acceleration

  $609A_h$（$041B_h$）=200000;

  Send:01 10 04 1B 00 02 04 00 03 0D 40（Set acceleration back to zero to 200000）

- Write control word

  6040_h(0380_h)= 0x06→0x07→0x0F→0x1F, the motor is running

  Send:01 10 03 80 00 01 02 00 06（Set 6040_h (0380_h)to 0x6 to make motor ready）

  Send:01 10 03 80 00 01 02 00 07（Set 6040h (0380h)to 0x7 to disable motor）

  Send:01 10 03 80 00 01 02 00 0F（Set 6040h(0380h) to 0xF to enable motor）

  Send:01 10 03 80 00 01 02 00 1F（Set 6040h (0380h)to 0x1F to run the motor）

5.8 Interpolation Mode (IP)

Interpolated position mode is used to synchronize multiple axes. To this end, the advanced controller performs ramp and path calculations and transfers the corresponding demand position of the axis at a specific time to the

controller. The controller is inserted between these intermediate positions.

5.8.1 Structure Diagram

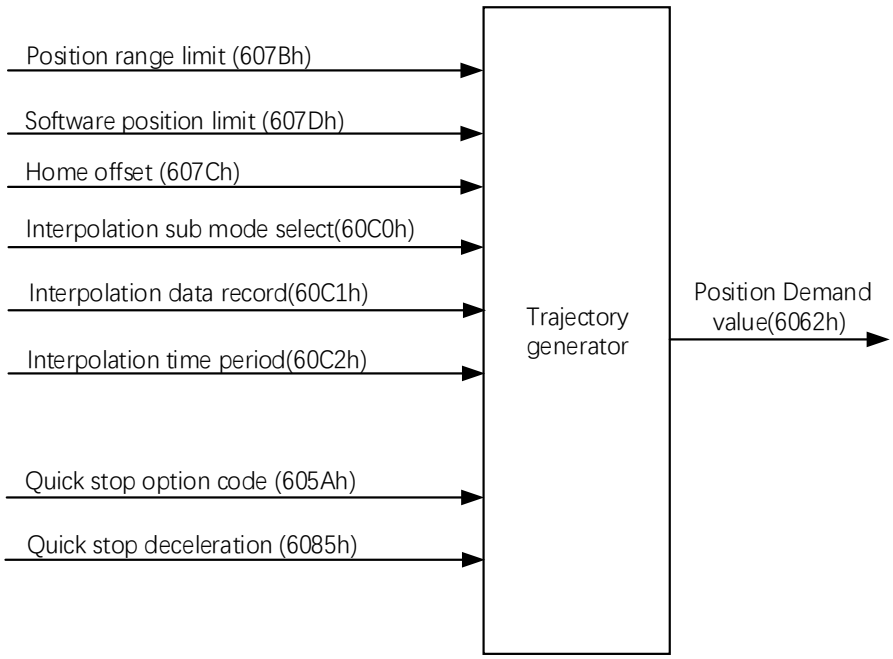


Figure 5-26

5.8.2 Related Objects

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

Table 5-26

Object Index	Description
607B$_h$：01$_h$ (03E9$_h$) 607B$_h$：02$_h$ (03EB$_h$)	Position range limit (user units).
607D$_h$：01$_h$ (03EF$_h$) 607D$_h$：02$_h$ (03F1$_h$)	The limit of the target position (user units).
60C0$_h$(042C$_h$)	Selection of interpolation type, the device only supports linear interpolation, the default value is 0
60C1$_h$：01$_h$ (042D$_h$)	Set the interpolation target position (user unit)
60C2$_h$：01$_h$ (042F$_h$) 60C2$_h$：02$_h$ (0430$_h$)	Interpolation time (recommended 1-20ms) Defined by 60C2$_h$：02$_h$ (0430$_h$) and 60C2$_h$：02$_h$ (0430$_h$), interpolation period = t*10n seconds 60C2$_h$: 01$_h$ (042F$_h$): Interpolation time constant t. 60C2$_h$: 02$_h$ (0430$_h$): Interpolation time index n
605A$_h$ (03BF$_h$)	Quick stop mode selection
605D$_h$ (03C0$_h$)	Pause mode selection
6085$_h$ (0400$_h$)	Stop deceleration when executing "fast stop" (user unit/s^2)
6062$_h$ (03C4$_h$)	The currently valid target position command value in the drive (user unit)
6063$_h$ (03C6$_h$)	Motor current absolute position feedback (encoder unit)

5.8.3 Control Commands and Status Messages

Enable

To enable this mode, you must set object 2002_h: 01_h（$00B1_h$） to "0" and object 6060_h ($03C2_h$)to "7".

Control word

15	9	8 7	6 5	4	3 0
(see 3.1.2)	Halt	(see 3.1.2)	reserved	Enable interpolation	(see 3.1.2)
MSB					LSB

In interpolation mode, the following bits in object 6040_h (0380_h) have special functions:

Table 5-27

Bit	Value	Meaning
4	0	Disability interpolation
	1	Enable interpolation
8	0	Execute bit 4 instruction
	1	Pause according to the configuration of object $605D_h$($03C0_h$)

Status word

15 14	13	12	11	10	9 0
(see 3.1.3)	following error	ip mode active	(see 3.1.3)	Target reached	(see 3.1.3)
MSB					LSB

In interpolation mode, the following bits in object 6041h (Status word) have special functions:

Table 5-28

Bit	Value	Meaning
10	0	Did not reach the target position
	1	Reach the target position
12	0	Interpolation is not activated
	1	Interpolation activation
13	0	No following error
	1	Follow error

5.8.4 Functional Description

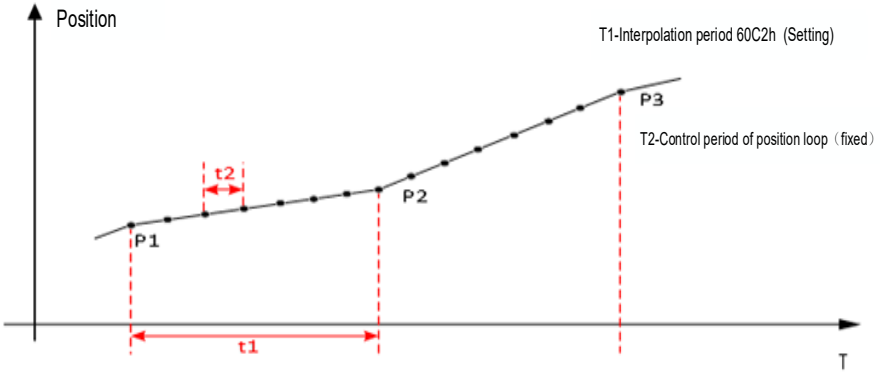


Figure 5-27 Interpolation diagram

Note: t1: Interpolation cycle can be set through the object dictionary $60C2_h$：01_h ($042F_h$)和 $60C2_h$：02_h (0430_h) (recommended 1-20ms).

t2: Position loop control period, determined internally by the drive.

P1/P2/P3: Absolute position. Absolute position commands are sent through the object dictionary $60C1_h$: 01_h($042D_h$).

Interpolation mode only supports absolute position commands.

5.8.5 Configuration Examples

- Configuration works in interpolation mode:
  Write 2002_h: 01_h（$00B1_h$）=0, operation mode 6060_h（$03C2_h$）=0x07;
  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）
  Send:01 10 03 C2 00 01 02 00 07（Set to interpolation mode）
- Configure interpolation time:
  Time constant $60C2_h$: 01_h ($042F_h$)= 20 and time index $60C2_h$: 02_h(0430_h) = -3, the time is 20ms;
  Send:01 10 04 2F 00 01 02 00 14（set $60C2_h$:01_h($042F_h$) to 20）
  Send:01 10 04 30 00 01 02 00 FD（set $60C2_h$:02_h(0430_h) to -3）
- Motor operation:
  Write control word 6040_h（0380_h） = 0x06→0x07→0x0F→0x1F
  Send:01 10 03 80 00 01 02 00 06（Set 6040_h to 0x6, Ready to make motor）
  Send:01 10 03 80 00 01 02 00 07（Set 6040_h to 0x7, Disable the motor）
  Send:01 10 03 80 00 01 02 00 0F（Set 6040_h to 0xF, Enable the motor）
  Send:01 10 03 80 00 01 02 00 1F（Set 6040_h to 1F,the motor running）
- Host computer writes interpolation position $60C1_h$: 01_h according to interpolation cycle (only supports absolute position commands).

5.9 Cyclic Synchronized Position Mode (CSP)

In this mode, absolute position presets will be sent to the controller via the field bus at fixed time intervals (hereinafter referred to as "cycles"). At this time, the controller no longer calculates the slope, it only follows the preset value. The target position is transmitted via PDO and the controller will react to it immediately. Bit 4 in Control word does not need to be set (unlike profile position mode). The target preset value is an absolute value, so it has nothing to do with the number of times each cycle is sent.

5.9.1 Structure diagram

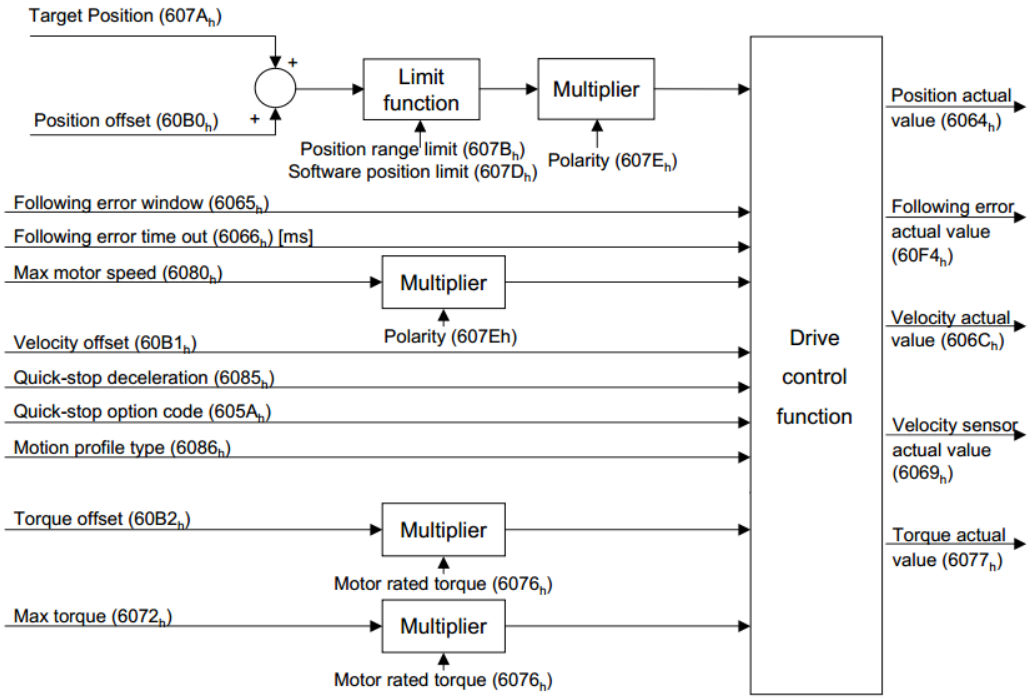


Figure 5-28

5.9.2 Related objects

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

Table 5-29

Object Index	Description
607Ah (03E7h)	Preset target location (user unit)
607Bh：01h (03E9h) 607Bh：02h (03EBh)	Location range limitation (user unit)
607Dh：01h (03EFh) 607Dh：02h (03F1h)	Target position limit (user units)
607Eh (03F3h)	Direction of rotation (polarity), see "4.3 607Eh: Polarity" for details
6065h (03CAh)	Following deviation threshold window (user unit)
6066h (03CCh)：	Time range of following error, unit: millisecond. If the actual position exceeds the error range 6065h(03CAh) longer than the specified time 6066h(03CCh), the following error will be triggered and bit 13 of the control word will be set to 1.
6080h (03F6h)	Maximum speed of motor (rpm)

6085_h (0400_h)	Stop deceleration when executing "fast stop" (user unit/s^2)
$605A_h$ ($03BF_h$)	Quick stop mode selection
$605D_h$ ($03C0_h$)	Pause mode selection
6086_h (0402_h)	The type of slope to be traveled. If the value is "0", the impact will not be limited. If the value is "3", the value in $60A4_h$: 01_h-04_h ($041E_h$~0424_h)will be used to limit the impact. This device only uses 01_h-02_h($041E_h$~0424_h) .
$60B0_h$ (0426_h)	Position offset (user unit)
$60B1_h$ (0428_h)	Speed offset (user unit)
6064_h ($03C8_h$)	Motor current user absolute position feedback (user unit)
$606C_h$ ($03D5_h$)	Current actual speed feedback value (rpm)
60F4h (0440_h)	Real-time following deviation (user unit)

5.9.3 Control Commands and Status Messages

Enable

To enable this mode, the value must be set to "0" in object 2002_h: 01_h（$00B1_h$） and "8" in object 6060h ($03C2_h$).

Control word

In the cyclic synchronous position mode, the bits in the object 6040_h (0380_h) have no special functions. For the definition of general bits, see section 3.1.2 Control Word.

Note: CSP mode only supports absolute position commands

Status word

15 14	13	12	11	10	9 0
(see 3.1.3)	Following error	Target position ignored	(see 3.1.3)	reserved	(see 3.1.3)

MSB LSB

In cyclic synchronized position mode, the following bits in object 6041h (0381_h) have special functions:

Table 5-30

Bit	Value	Meaning
12	0	The controller does not follow the target preset value, the preset value of $607A_h$ ($03E7_h$) will be ignored
	1	The controller follows the target preset value and the object $607A_h$ ($03E7_h$) is used as the position control input
13	0	No following error
	1	Follow error

5.9.4 Functional Description

In this mode, absolute position presets will be sent to the controller via the fieldbus at fixed time intervals (hereinafter referred to as "cycles").

5.9.5 Configuration Examples

- Configuration mode:
  Working in cyclic synchronized position mode: write 2002_h: 01_h（$00B1_h$） = 0, running mode 6060_h（$03C2_h$） = 0x08;
  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）
  Send:01 10 03 C2 00 01 02 00 08（Set to cyclic synchronization position mode）
- Motor operation:
  Write control word 6040_h（0380_h） = 0x06→0x07→0x0F;
  Send:01 10 03 80 00 01 02 00 06（Set 6040_h（0380_h） to 0x6, Ready to make motor）
  Send:01 10 03 80 00 01 02 00 07（Set 6040_h （0380_h）to 0x7, Disable the motor）
  Send:01 10 03 80 00 01 02 00 0F（Set 6040_h （0380_h）to 0xF, Enable the motor）
- The host computer sends the target position 607A($03E7_h$) according to the synchronization cycle .

5.10 Cyclic Synchronous Velocity Mode (CSV)

In the periodic synchronous Velocity mode, the host controller sends the calculated target speed $60FF_h$（0448_h）to the drive periodically and synchronously, and the speed and torque adjustment are performed internally.

5.10.1 Structure Drawing

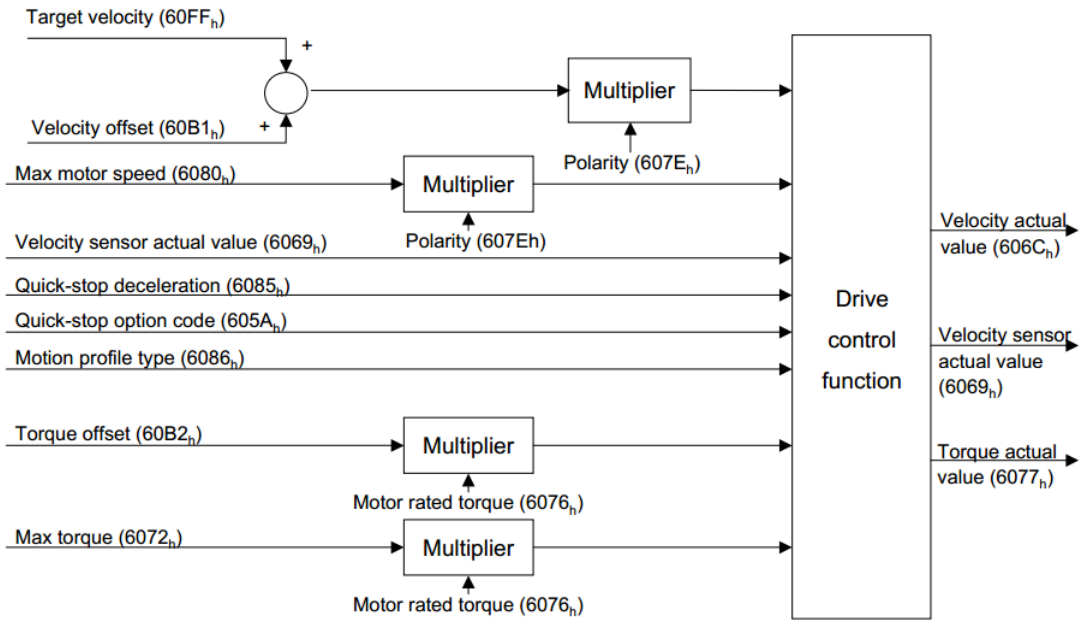


Figure 5-29

5.10.2 Related Objects

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

Table 5-31

Object Index	Description
$605A_h$ ($03BF_h$)	Quick stop mode selection
$605D_h$ ($03BE_h$)	Pause mode selection
$606C_h$（$03D5_h$）	Current actual speed feedback value (unit: rpm)
$607E_h$（$03F3_h$）	Direction of rotation (polarity), see "4.3 607Eh: Polarity" for details
$60FF_h$（0448_h）	Set the target speed (user units/s)
6080_h（$03F6_h$）	Maximum speed of motor (rpm)
6085_h（0400_h）	Stop deceleration when executing "fast stop" (user unit/s^2)
6086_h（0402_h）	Select the type of curve or 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 $60A4_h$: 01_h-04_h ($041E_h$~0424_h)will be used to limit the impact (acceleration), that is, the S-curve, the device only uses the 01_h - 02_h ($041E_h$~0420_h)indexes.
$60B1_h$（0428_h）	Position offset (user unit)
$60B2_h$（$042A_h$）	Torque bias (0.1%)
6072_h（$03DC_h$）	Maximum torque (0.1%) on the entire ramp (acceleration, torque saving, braking)

5.10.3 Control Commands and Status Messages

Enable

To enable this mode, the value must be set to "0" in object 2002_h: 01_h（$00B1_h$） and "9" in object 6060_h ($03C2_h$).

Control word

In the cyclic synchronous Velocity mode, the bits in the object 6040_h (0380_h) have no special functions. For the definition of general bits, refer to chapter 3.1.2 Control Word.

Status word

15 14	13	12	11	10	9 0
(see 3.1.3)	reserved	Target velocity ignored	(see 3.1.3)	reserved	(see 3.1.3)

MSB LSB

In cyclic synchronous Velocity mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-32

Bit	Value	Meaning
12	0	Target speed will be discarded
	1	Target speed as planner input

5.10.4 Functional Description

In the periodic synchronous Velocity mode, the host controller sends the calculated target speed $60FF_h$(0448_h) to the motor driver periodically and synchronously, and the speed adjustment is performed internally by the motor.

5.10.5 Configuration Examples

- Setting mode:
  Write 2002_h: 01_h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x09, make it work in cyclic Velocity mode;
  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）
  Send:01 10 03 C2 00 01 02 00 09（Set to cyclic synchronous speed mode）
- Write control word
  6040_h（0380_h）=0x06→0x07→0x0F, the motor is running, the host computer needs to write the speed object $60FF_h$ according to the synchronization cycle
  Send:01 10 03 80 00 01 02 00 06（Set 6040_h （0380_h）to 0x6, Ready to make motor）
  Send:01 10 03 80 00 01 02 00 07（Set 6040_h （0380_h）to 0x7, Disable the motor）
  Send:01 10 03 80 00 01 02 00 0F（Set 6040_h （0380_h）to 0xF, Enable the motor）

5.11 Cyclic Synchronous Torque Mode (CST)

In this mode, the host controller sends the calculated target torque 6071_h ($03DB_h$)to the drive periodically and synchronously. The torque adjustment is performed internally. When the speed reaches the limit value, it will enter

the speed regulation stage.

5.11.1 Structure Diagram

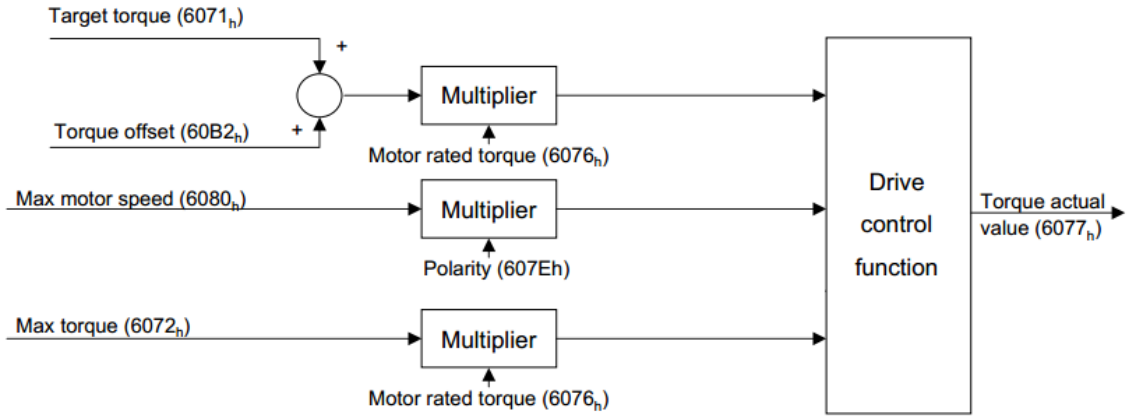


Figure 5-30

5.11.2 Related Objects

In this mode, you need to pay attention to the following items:

Table 5-33

Object Index	Description
605Ah (03BFh)	Quick stop mode selection
605Dh (03BEh)	Pause mode selection
6071h (03DBh)	Set target torque (0.1%)
6072h (03DCh)	Maximum torque (0.1%) on the entire ramp (acceleration, torque saving, braking)
6073h (03DDh)	Maximum allowable current (0.01A)
6080h (03F6h)	Maximum speed of motor (rpm)
60B2h (042Ah)	Torque bias (0.1%)
6088h (0405h)	Torque ramp type (0-ramp, 2-none)
6077h (03E3h)	Actual torque value (0.001N.m)

5.11.3 Control Commands and Status Messages

Enable

To enable this mode, the value must be set to "0" in object 2002h: 01h（00B1h） and "8" in object 6060h (03C2h).

Control word

In the cyclic synchronous torque mode, the bits in the object 6040h (0380h) have no special functions. For the definition of general bits, see section 3.1.2 Control Word.

Status word

15 14	13	12	11	10	9 0
(see 3.1.3)	reserved	Target torque ignored	(see 3.1.3)	reserved	(see 3.1.3)

MSB LSB

In cyclic synchronous torque mode, the following bits in object 6041_h (0381_h) have special functions:

Table 5-34

Bit	Value	Meaning
12	0	The target torque will be discarded
	1	Target torque as planner input

5.11.4 Functional Description

In this mode, the host controller sends the calculated target torque 6071_h($03DB_h$) to the drive periodically and synchronously.

5.11.5 Configuration Examples

- Setting mode:
  Write 2002_h: 01_h（$00B1_h$）=0, running mode 6060_h（$03C2_h$）=0x0A, make it work in cyclic synchronous torque mode;
  Send:01 10 00 B1 00 01 02 00 00 （Set to CiA402 mode）
  Send:01 10 03 C2 00 01 02 00 0A（Set to cyclic synchronous torque mode）
- Write control word
  Write control word 6040_h（0380_h）=0x06→0x07→0x0F, the motor is running, the host computer needs to write the target torque 6071_h($03DB_h$)
  Send: 01 10 03 80 00 01 02 00 06 (set 6040h (0380h) to 0x6 to make motor ready)
  Send: 01 10 03 80 00 01 02 00 07 (set 6040h (0380h) to 0x7 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040h (0380h) to 0xF to enable motor)

5.12 NiMotion Mode

5.12.1 Overview

The NiMotion mode is different from the CIA402 mode. There is no state machine management, and the position, speed, and torque control are carried out through the parameters configured by the multi-function terminals,

including NiMotion position mode, NiMotion Velocity mode, and NiMotion torque mode. 2002_h: 01_h （$00B1_h$）(control mode selection) is used to determine which working mode is in, the specific correspondence is shown in 5.1 Control Mode Overview.

5.13 NiMotion Position Mode

5.13.1 Structure diagram

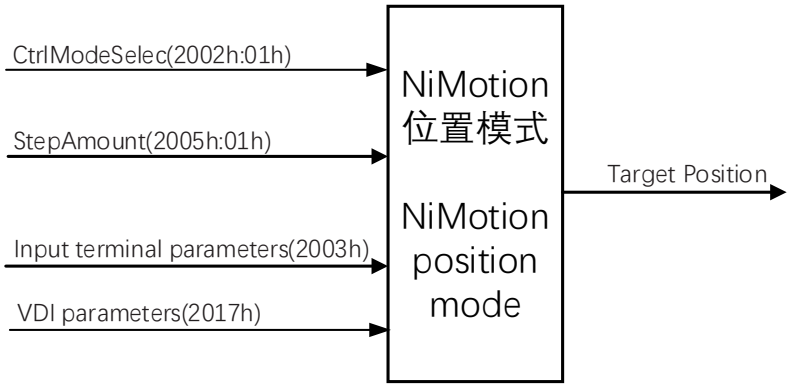


Figure 5-31

5.13.2 Related Objects

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

Table 5-35

Object Index	Description
2002_h：01_h ($00B1_h$)	Operating mode selection
2005_h:01_h ($010D_h$)	Position command source, 0-Pulse, 1-Step, 2-Multistage position
2005_h:05_h (0112_h)	Step setting (encoder unit)
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input" for details)
2003_h ($00D5_h$ ~$00E6_h$)	Physical input terminal function configuration (see: "6.1.1 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 fault will be reported

5.13.3 Control Commands and Status Messages

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

Enable

The operation premise of this mode: the integrated low-voltage motor is switched to NiMotion position mode (2002_h: 01_h ($00B1_h$)write 1).

1 Location source selection:

The object (2005_h: 01_h($010D_h$)) is the source of the position command, meaning: 0-pulse, 1-step ;

2 Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 Digital Input and Output" for terminal setting details)
If the set position command source is 1, configure one input terminal as the motor enable, and configure its effective logic level according to the actual situation, the default low level is effective; set the other input terminal as the step enable, according to the actual configuration Its effective logic level, the default low level is effective;
If the set position command source is 0 (external pulse given), in addition to setting the motor enable terminal, it is also necessary to set the input terminals DI1 and DI2 as the corresponding pulse input function.

3 Location setting
If the source of the position command is 1, you need to set the given value of the step amount and write it to the object 2005_h: 05_h(0112_h);
If the source of the position command is 0, you need to input pulse signals from the input terminals DI1 and DI2.

4 Run
If the source of the position command is 1, the motor is enabled through the terminal, and the step signal is enabled through the terminal, the distance (unit: encoder unit) set by the motor travel 2005_h:05_h (0112_h), each step is enabled , The motor travels once;
If the source of the position command is 0, the motor is enabled according to the set motor enable terminal, and the motor runs according to the pulse signal of the input terminal.

5.13.4 Functional Description

In this mode, the motor will travel a controllable distance according to the input signal of the terminal.

5.13.5 Examples

Step position control

parameter settings:

- Set to NiMotion position mode 2002_h: 01_h($00B1_h$)=1

Send:01 10 00 B1 00 01 02 00 01 （Set to NiMotion position mode）

- Set the position command source as step amount: 2005_h: 01_h($010D_h$)=1

Send:01 10 01 0D 00 01 02 00 01（Set the source of position command as step amount）

- Set step size: 2005_h: 05_h (0112_h) = 50

Send:01 10 01 12 00 01 02 00 32（Set the step amount to 50）

- Set the motor enable: 2003_h: 03_h($00D5_h$)=1, the logic is low level effective: 2003h: 04_h($00D6_h$)=0

Send:01 10 00 D7 00 01 02 00 14（Set the DI1 function to servo enable）
Send:01 10 00 D8 00 01 02 00 03（Set the DI1 logic to low level valid）

- Set step enable: 2003_h: 05_h($00D7_h$)=20, logic is valid for falling edge: 2003_h: 06_h($00D8_h$)=3

Send:01 10 00 D7 00 01 02 00 14（Set the DI2 function to step volume enable）
Send:01 10 00 D8 00 01 02 00 03（Set the DI2 logic to low level valid）

Running:

- Motor enable: DI1 input low level
- 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 settings:

- Set to NiMotion position mode 2002_h: $01_h(00B1_h)$=1

Send:01 10 00 B1 00 01 02 00 00（Set to NiMotion position mode）

- Set the position command source as pulse: 2005_h: $01_h(010D_h)$=0

Send:01 10 01 0D 00 01 02 00 00（Set the position command source to pulse）

- Set input terminal:

DI1 orthogonal pulse input A: 2003_h: 03_h=40
Send:01 10 00 D5 00 01 02 00 28（Set the DI1 function to quadrature pulse input A）
DI2 orthogonal pulse input B: 2003_h: 05_h=41
Send:01 10 00 D7 00 01 02 00 29（Set the DI2 function to quadrature pulse input B）
DI3 function number is set to motor enable:
The function selection is stepping enable: 2003_h: 07_h =1
Send:01 10 00 D9 00 01 02 00 01（Set the DI3 function to servo-enabled）
Logic selection low level effective: 2003_h: 08_h =0
Send:01 10 00 DA 00 01 02 00 00（Set the DI3 logic to low level valid）

Running:

- Motor enable: DI3 input low level
- Pulse input control: The terminals DI1 and DI2 input orthogonal pulses, and the motor performs position control according to the input pulses

5.14 NiMotion Velocity mode

5.14.1 Structure Diagram

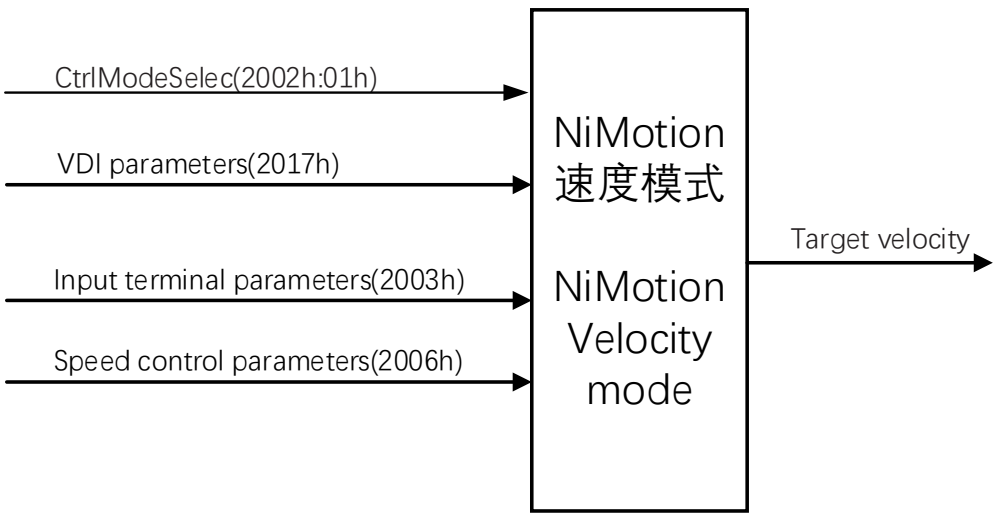


Figure 5-32

5.14.2 Related Objects

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

Table 5-36

Object Index	Description
2000_h：$0F_h(006B_h)$	Maximum speed, rpm
2002_h：01_h ($00B1_h$)	Operating mode selection
2006_h：01_h（0148_h）	NiMotion velocity mode main velocity reference source, 0-digital reference, 3-duty cycle reference (20Hz-20kHz, 1kHz recommended)
2006_h：02_h（0149_h）	2006_h: 02_h: NiMotion velocity mode auxiliary velocity reference source, 0-digital reference, 3-multi-velocity
2006_h：03_h（$014A_h$）	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（$014B_h$）	Velocity command digital setting value, unit rpm
2006_h：07_h（$014E_h$）	Velocity command acceleration ramp time constant, unit ms, indicating the time it takes to accelerate from 0rpm to 1000rpm.
2006_h：08_h（$014F_h$）	Velocity command deceleration ramp time constant, unit ms
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input" for details)
2003_h ($00D5_h$ ~$00E6_h$)	Physical input terminal function configuration (see: "6.1.1 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 fault will be reported

5.14.3 Control Commands and Status Information

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

Enable

The premise of this mode of operation: the integrated low-voltage motor is switched to NiMotion Velocity mode (2002_h: 01_h($00B1_h$) is set to 2).

1 Speed source selection:

The object is the source of the speed command 2006_h: 01_h(0148_h), meaning: 0-number given, 3-duty ratio

given;

2 Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 Digital Input and Output" for terminal setting details):
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. If the speed command source set in the previous step is 3 (duty cycle given), you need to set the corresponding input terminals DI1 and DI2 as the duty cycle input function

3 Speed, acceleration and maximum speed settings
If the source of the speed command is 0-digital reference, you need to set the speed reference value and write it to the object 2006_h: $04_h(014B_h)$;
If the selected speed command source is 3-duty ratio setting, you need to input pulse signals from DI1 (duty ratio) and DI2 (direction).when the direction input signal is invalid, 0% duty ratio corresponds to the negative maximum speed 2000_h: $0F_h$ ($006B_h$), 100% duty ratio corresponds to speed 0; When the direction input signal is valid, 0% duty ratio corresponds to the positive maximum speed 2000_h: $0F_h(006B_h)$, 100% duty ratio corresponds to speed 0;
If the main speed command source is selected as 4~50% duty cycle speed control setting, you need to input pulse signal from DI1 (duty cycle), 0~50% corresponds to negative limited speed 2006_h: $0B_h$ (0152_h)~0,50% -100% corresponds to 0~ positive limit speed 2006_h:$0A_h(0151_h)$.
Configure acceleration and deceleration 2006_h: $07_h(014E_h)$ and 2006_h: $08_h(014F_h)$
Configure the maximum speed 2000_h: $0F_h(006B_h)$, the duty cycle speed is 100% corresponding to the maximum speed

4 Run
According to the set motor enable terminal, the motor is enabled, and the motor runs at the set speed.

5.14.4 Functional Description

In this mode, the motor will run to the corresponding speed according to the terminal input or digital setting.

5.14.5 Examples

Realize digital given speed regulation:

1 Parameter setting:

Set the Velocity mode to 2 (NiMotion Velocity mode): 2002_h: $01_h(00B1_h)$=2
send：01 10 00 B1 00 01 02 00 02(Set to NiMotion speed mode)

Set the source of the main speed command to 0 (number given): 2006_h: $01_h(0148_h)$=0;
Send：01 10 01 48 00 01 02 00 00(Set the speed command source to 0)
Set the source of the auxiliary speed command to 0 (number given): 2006_h: $03_h(014A_h)$=0;
Send：01 10 01 4A 00 01 02 00 00(Set the speed command as the main speed given)
Set the acceleration ramp time constant to 100ms: 2006_h: $07_h(014E_h)$=100
Send：01 10 01 4E 00 01 02 00 64(Set the acceleration ramp time constant to 100ms)
Set the deceleration ramp time constant to 100ms: 2006_h: $08_h(014F_h)$=100
Send：01 10 01 4F 00 01 02 00 64(Set the deceleration ramp time constant to 100ms)
Set the digital setting to 60rpm, 2006_h: $04_h(014B_h)$=60

Send：01 10 01 4B 00 01 02 00 3C(Set the digital reference to 60rpm)

2 Terminal settings

Set DI1 terminal:
The function selection is step enable: 2003_h: 03_h ($00D5_h$)=1
Send：01 10 00 D5 00 01 02 00 01(Set the DI1 function to servo enable)
Logic selection low level effective: 2003_h: 04_h ($00D6_h$) =0
Send：01 10 00 D6 00 01 02 00 00(Set the DI1 logic to low level valid)

3 Running:

DI1 input low level, the motor speed regulation according to the digital setting 2006_h: 02_h($014B_h$).

To achieve duty cycle input speed regulation:

1 Parameter setting:

Set the Velocity mode to 2 (NiMotion speed): 2002_h: 01_h($00B1_h$)=2
Send：01 10 00 B1 00 01 02 00 02(Set to NiMotion speed mode)

Set the speed command source to 3 (duty cycle): 2006_h: 01_h(0148_h)=3
Send：01 10 01 48 00 01 02 00 03(Set the speed command source to 3)

Set the acceleration ramp time constant to 100ms: 2006_h: 07_h($014E_h$)=100
Send：01 10 01 4E 00 01 02 00 64(Set the deceleration ramp time constant to 100ms)
Set the deceleration ramp time constant to 100ms: 2006_h: 08_h($014F_h$)=100
Send：01 10 01 4F 00 01 02 00 64(Set the deceleration ramp time constant to 100ms)
Set the maximum speed to 3000rpm: 2000h: 0Fh=3000
Send：01 10 00 6B 00 01 02 0B B8(Set the maximum speed to 3000rpm)

2 Terminal settings

DI1 duty ratio input: 2003h: 03h=44
Send：01 10 00 D5 00 01 02 00 2C(Set DI1 function to duty cycle input)
DI2 duty direction input: 2003h: 05h=45
Send：01 10 00 D7 00 01 02 00 2D(Set DI2 to the duty cycle direction input)
DI3 function number is set to step enable:
The function selection is stepping enable: 2003h: 07h =1
Send：01 10 00 D9 00 01 02 00 01(Set the DI3 function to servo-enabled)
Logic selection is active low: 2003h: 08h =0
Send：01 10 00 DA 00 01 02 00 00(Set the DI3 logic to low level valid)

3 Running:

DI1 input square wave with certain duty ratio, DI3 input low level to enable the motor, the motor speed regulation according to the input duty ratio

5.15 NiMotion Torque Mode

5.15.1 Structure Diagram

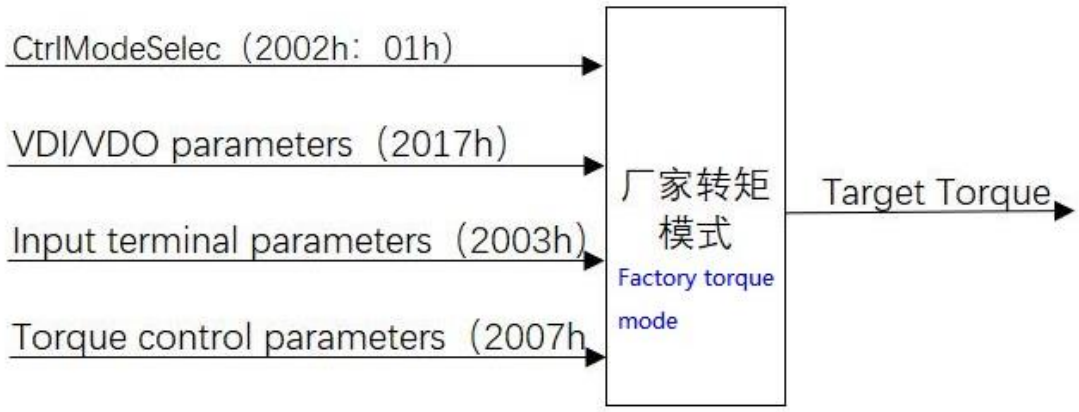


Figure 5-33

5.15.2 Related Objects

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

Table 5-37

Object Index	Description
2002h：01h(00B1h)	Operating mode selection
2007h：01h(015Eh)	NiMotion torque mode main torque reference source, 0-digital reference
2007h：02h(015Fh)	NiMotion torque mode auxiliary torque reference source, 0-digital reference
2007h：03h(0160h)	NiMotion torque mode torque command selection, 0- main torque reference, 1-auxiliary torque reference, 2- main torque and auxiliary torque reference (default 0)
2007h：04h(0161h)	2007h: 04h: Digital setting value of torque command (0.1%)
2007h：0Ah(0167h)	positive internal torque limit (0.1%)
2007h：0Bh(0168h)	anti-internal torque limit (0.1%)
2007h：10h(016Dh)	Torque control forward speed limit value (rpm)
2007h：11h(016Eh)	Reverse speed limit value of torque control (rpm)

Object Index	Description
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input" for details)
2003_h ($00D5_h$ ~$00E6_h$)	Physical input terminal function configuration (see "6.1.1 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 fault will be reported

5.15.3 Control commands and status messages

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

Enable

The premise of this mode of operation: the integrated low-voltage brushless motor is switched to the NiMotion torque mode (2002 $_h$: 01_h($00B1_h$) is set to 3).

1 Torque source selection:
The object 2007_h: 01_h($015E_h$) is the source of torque command, meaning: 0-number given;

2 Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 Digital Input and Output" for terminal setting details):
Configure one input terminal to enable the motor, and configure its effective logic level according to the actual situation.

3 Torque setting and torque and speed limit settings
If the source of the speed command is digital reference, you need to set the torque reference value and write it to the object 2007_h: 04_h(0161_h);
Configure positive and negative internal torque limits 2007h:$0A_h$(0168_h) and 2007h:$0B_h$(0168_h)
Configure forward and reverse speed limit values 2007_h: 10_h($016D_h$) and 2007_h: 11_h(0168_h).

4 run
According to the set motor enable terminal, the motor is enabled, and the motor runs at the set torque.

5.15.4 Functional description

In this mode, the motor will run to the corresponding torque through the enable control of the physical or virtual terminals and the torque digital setting. When the running speed of the motor exceeds the speed limit (forward speed limit value 2007_h: 10_h （$016D_h$）and reverse speed limit value 2007_h: 11_h（$016E_h$））, it will automatically switch to Velocity mode control and control the current speed within the speed limit . When it is detected that the given target torque is less than the average torque of the current speed, the motor exits the speed control and returns to the torque mode control.

5.15.5 Examples

1. Parameter setting:
Set the Velocity mode to 3 (NiMotion torque mode): 2002h: 01h(00B1h)=3;
Send：01 10 00 B1 00 01 02 00 03(Set to NiMotion torque mode)
Set the speed command source to 0 (digital reference): 2007h: 01h(015Eh)=0
Send：01 10 01 5E 00 01 02 00 00(Set the speed command source to 0)

Set the speed command to select the main torque command: 2007h: 03h(0160h)=0

Send：01 10 01 60 00 01 02 00 00(Set the speed command to select the main torque command)

Set positive and negative internal torque limit to 100%: 2007h: 0Ah(0167h)=1000 and 2007h: 0Bh(0168h)=1000

Send：01 10 01 67 00 01 02 03 E8（Set forward internal torque limit to 100%）

Send：01 10 01 68 00 01 02 03 E8（Set reverse internal torque limit to 100%）

Set the forward and reverse speed limit value to 3000rpm: 2007h: 10h(016Dh)=3000 and 2007h: 11h(016Eh)=3000

Send：01 10 01 6D 00 01 02 0B B8（Set forward speed limit values to 3000 rpm）

Send：01 10 01 6E 00 01 02 0B B8（Set reverse speed limit values to 3000 rpm）

Set the number given as 100%, 2007h: 04h(0161h)=1000

Send：01 10 01 61 00 01 02 03 E8(Set the digital reference to 100%)

2. Terminal settings

Set DI1 terminal:

Function selection for step enable: 2003h: 03h (00D5h)=1

Send：01 10 00 D5 00 01 02 00 01(Set the DI1 function to step enable)

Logic selection is active low: 2003h: 04h (00D6h)=0

Send：01 10 00 D6 00 01 02 00 00(Set DI1 logic to low level valid)

3. Running:

DI1 input low level, the motor runs according to the torque of digital setting 2007_h:04_h (0161_h)

5.16 Multi-Segment Position Mode

5.16.1 Structure Diagram

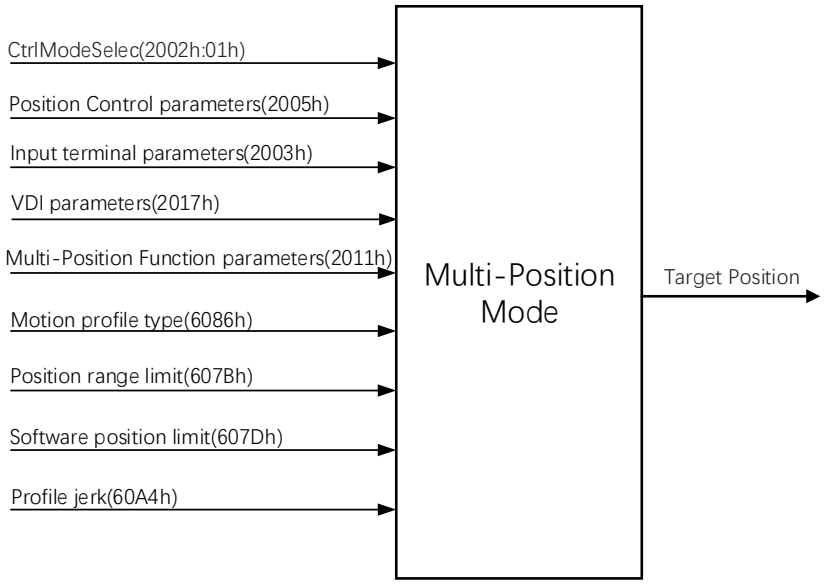


图 Figure 5-34

5.16.2 Related Objects

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

Table 5-38

Object Index	Description
2002h:01h (00B1h)	Operating mode selection
2005h：01h (010Dh)	source of position command, 0-pulse, 1-step, 2-multi-step position
2005h：05h (0112h)	Step amount setting (encoder unit)
2017h (0326h ~0346h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input" for details)
2003h (00D5h ~00E6h)	Physical input terminal function configuration (see: "6.1.1 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 fault will be reported
2011h (0291h ~02E6h)	Multi-section position function parameter group
6086h (0402h)	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(041Eh~0420h) will be used to limit the impact (acceleration), that is, the S-curve, the device only uses the 01h and 02h（041Eh and 0420h） indexes.
607Bh（03E9h~03EBh）	Location range limitation (user unit)
607Dh（03EFh~03F1h）	Target position limit (user units)
60A4h（041Eh~0424h）	profile acceleration, please refer to 6.3.2 for details

5.16.3 Control Commands and Status Messages

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

Enable

The premise of this mode of operation: the motor operation mode is switched to NiMotion position mode (2002h:01h(00B1h) write 1).

1. Location source selection:
   The object 2005h:01h (010Dh)is the source of the position command 2005h:01h(010Dh), meaning: 0-pulse, 1-step, 2-multi-step position;

2. Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 Digital Input and Output" for terminal setting details)
   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-position position/speed enable (function number 28), according to the actual configuration of its effective logic level, the default low level is effective;

3. Location setting
   Through the multi-segment position function parameter group 2011h (0291h ~02E6h), set the corresponding operation mode and operation parameters of each stage.

4. Run

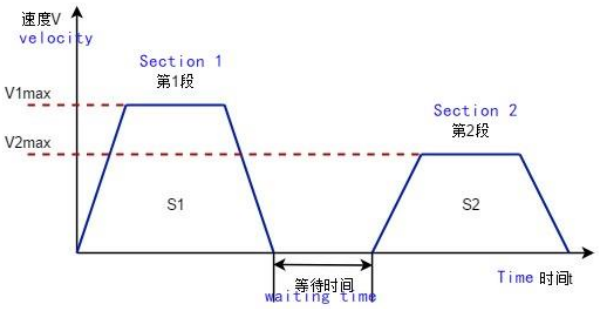
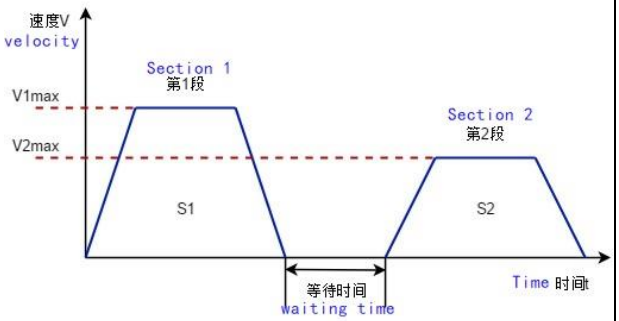
First enable the motor through one terminal and then enable the multi-segment position command through the other terminal. The motor will run according to the operation mode and operation parameters set by the multi-segment position function parameter group $2011_h(0291_h \sim 02E6_h)$.

5.16.4 Functional Description

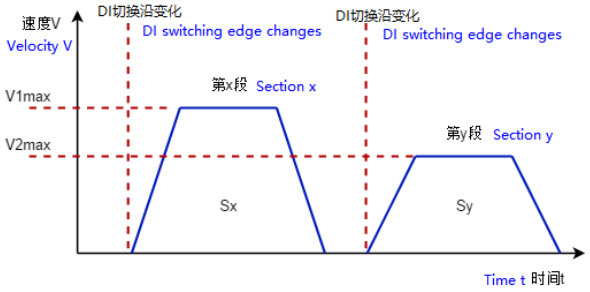
There are three operation modes for the multi-stage position mode, which are single operation, cyclic operation and DI switching operation. When using the multi-segment position function, the DI port must be set to enable (function number 1) and multi-segment position/speed enable (function number 28); during each segment operation, the enable must be valid; in non-DI switching operation mode, In addition to the enabling effect, it must be ensured that the multi-segment position/speed enable is effective. If the multi-segment position/speed enable is turned off, the displacement command not sent by this segment will be abandoned and the machine will be stopped.

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

Table 5-39

Operation mode	Explanation	Running curve
Single run $2011_h:01_h(0291_h) = 0$	Run once from paragraph 1 to paragraph 16; The segment number is automatically incremented and switched; Waiting time can be set between segments; Multi-level position/speed enable is level effective.	
cycle operation $2011_h:01_h(0291_h) = 1$	The number of times set from $2011_h:06_h(0296_h)$ to the end point set by $2011_h:02(0292_h)$= to the end point set by $2011_h:04_h(0294_h)$; The segment number is automatically incremented and switched; Waiting time can be set between segments; Multi-level position/speed enable is level effective; Note: If the setting value of the end segment is less than the start segment, the first segment to the 16th segment will cycle once.	

<table>
<tr><th>Operation mode</th><th>Explanation</th><th>Running curve</th></tr>
<tr>
<td>DI switch operation 2011_h:01_h (0291_h)= 2

Note: The physical terminal and the virtual terminal can be used in combination, and the combination must be 4 bits; for example: Physical terminal selection DI1=6, DI2=7, VI1=8, VI2=9.</td>
<td>The position segment number switching is determined by the decimal 0~15 corresponding to the 4-bit binary number composed of the multi-function numbers 6, 7, 8, 9;
The waiting time set between segments is invalid;
Multi-level position/speed enable is level effective;
The relationship between the section number and the multi-function number is as follows:
<table>
<tr><th>Fun9</th><th>Fun8</th><th>Fun7</th><th>Fun6</th><th>Segment 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>
</table>

5.16.5 Configuration Examples

Multi-section position-single operation control

- Parameter setting

Set to NiMotion position mode: 2002_h:01_h($00B1_h$)=1;
Set the source of the position command as a multi-segment position: 2005_h:01_h($010D_h$)=2;
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 the source of the position command as a multi-segment position)
Set multi-section position function parameter group:
Section position operation mode: 2011_h:01_h (0291_h)= 0 (single operation);
Send：01 10 02 91 00 01 02 00 00(Set the multi-segment position running mode as single run)
Position command type selection: 2011_h:05_h (0295_h)= 0 (relative displacement);
Send：01 10 02 95 00 01 02 00 00(Set the position instruction type to relative displacement)
Operating parameters of the first stage displacement:
Movement displacement 2011_h:07_h (0297_h)= 10000 (user unit), the maximum allowable speed 2011_h:08_h (0299_h)=500 (unit rpm),
Send：01 10 02 97 00 02 04 00 00 27 10（Set the first segment moving displacement to 10000）
Send：01 10 02 99 00 01 02 01 F4（Set the maximum running speed of the first segment displacement to 500rpm）
Acceleration and deceleration time 2011h:09h ($029A_h$)= 500 (unit ms), waiting time after

displacement is completed 2011_h:$0A_h$ ($029B_h$)=1000 (unit ms)

Send：01 10 02 9A 00 01 02 01 F4（Set the acceleration and deceleration time of the first stage displacement as 500ms）

Send：01 10 02 9B 00 01 02 03 E8（Set the wait time for the completion of the first displacement to 1000ms）

Paragraphs 2 to 16 refer to the operating parameter settings for the displacement of paragraph 1;

Set motor enable: 2003h:03h(00D5h)=1、Logic is active low:2003h:04h(00D6h)=0；

Set the motor enable: 2003h:03h(00D5h)=1, the logic is low level effective: 2003h:04h(00D6h)=0;

Send：01 10 00 D5 00 01 02 00 01（Set the motor enable）

Send：01 10 00 D6 00 01 02 00 00（Set DI1 logic to active low level）

Setting Multi-Segment Position/Speed Enable: 2003h:05h(00D7h)=28 、 Logic is active low:2003h:06h(00D8h)=0；

Set multi-stage position/speed enable: 2003_h:05_h($00D7_h$)=28, logic is low level effective: 2003_h:06_h($00D8_h$)=0;

Send：01 10 00 D7 00 01 02 00 1C（Set multi-stage position/speed enable）

Send：01 10 00 D8 00 01 02 00 00（Set DI2 logic to active low level）

- Run

Motor enable: DI1 input low level;

Multi-stage position/speed enable, DI2 input low level, the motor operates according to the operating mode and operating parameters set by the 2011_h (0291_h ~$02E6_h$)multi-stage position function parameter group.

5.17 Multi-Speed mode

5.17.1 Structure diagram

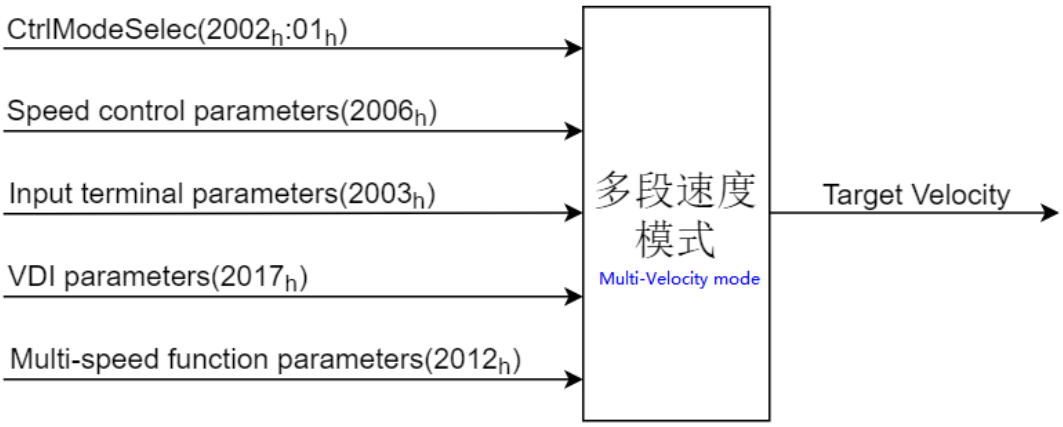


Figure 5-35

5.17.2 Related Objects

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

Table 5-40

Object Index	Description
2002h:01h (00B1h)	Operating mode selection
2006h：01h (0148h)	NiMotion Velocity mode main speed reference source, 0-digital reference, 3-duty cycle reference (20Hz-20kHz, 1kHz recommended)
2006h：02h (0149h)	NiMotion Velocity mode auxiliary speed reference source, 0-number reference, 3-multi-step speed
2006h：03h (014Ah)	NiMotion Velocity mode speed command selection, 0-main speed reference, 1-auxiliary speed reference, 2-main speed and auxiliary speed reference (default 0)
2006h：04h (014Bh)	Speed command digital setting value, unit rpm
2006h：07h (014Eh)	Speed command acceleration ramp time constant, unit ms, indicating the time it takes to accelerate from 0 rpm to 1000 rpm.
2006h：08h (014Fh)	Speed command deceleration ramp time constant, unit ms
2017h (0326h ~0346h))	Virtual input terminal function configuration (see: "6.1.1 Digital Input" for details)
2003h (00D5h ~00E6h)	Physical input terminal function configuration (see: "6.1.1 Digital Input" for details)

Object Index	Description
	It should be noted that multiple input terminals cannot be configured as the same switch at the same time, otherwise the terminal setting fault will be reported
2012_h ($02E9_h$ ~0323_h)	Multi-stage speed function parameter group

5.17.3 Control Commands and Status Messages

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

Enable

The premise of this mode of operation: the motor operation mode is switched to NiMotion Velocity mode 2002_h:01_h ($00B1_h$)is set to 2.

1. Velocity source selection:
   The object is the source of the auxiliary speed command 2006_h:02_h(0149_h), meaning: 0-number setting, 3-multi-step speed,

   The speed command selection 2006_h:03_h($014A_h$) is actually the main speed command, auxiliary speed command or main and auxiliary speed command.

2. Set multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 Digital Input and Output" for terminal setting details):
   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.

   Configure the other input terminal as multi-segment enable (function number 28), according to the actual configuration of its valid logic level, the default low level is valid.

3. Velocity setting
   Set the corresponding operation mode and operation parameters of each stage through the multi-stage speed function parameter group 2012_h($02E9_h$ ~0323_h).

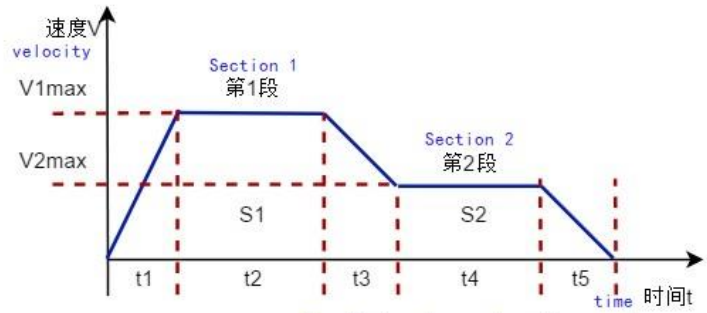
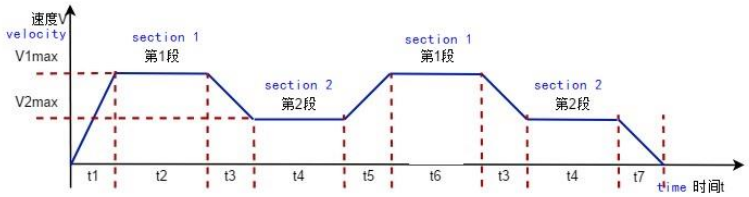
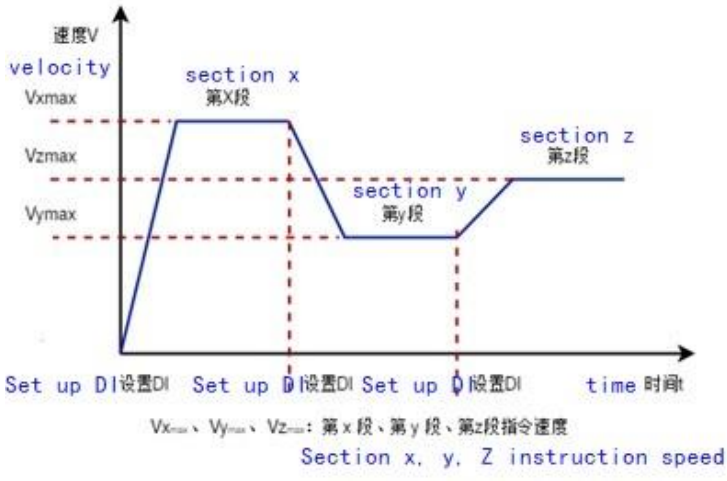
4. Run
   The motor is enabled through one terminal, and the motor runs according to the operation mode set by the multi-stage speed function parameter group 2012_h($02E9_h$ ~0323_h)and the operation parameters of each stage.

5.17.4 Functional description

There are 3 operation modes in the multi-speed mode, which are single operation, cyclic operation and DI switching operation. When using the multi-segment position/speed function, a DI port must be set for motor enable (function number 1) and multi-segment enable (function number 28); during each speed command operation, the motor enable must be guaranteed to be valid, if the motor is turned off If enabled, the motor will stop according to the disabling mode. In the non-DI switching operation mode, in addition to the effective motor enable, the multi-stage enable must be ensured. If the multi-stage enable is turned off, the motor will not be able to perform multi-speed switching.

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

Table 5-41

Operation mode	Explanation	Running curve
Single run $2012_h:01_h$(02 $E9_h$) = 0	Run once from paragraph 1 to paragraph 16; The segment number is automatically incremented and switched. Multi-segment enable is level effective.	 The first and second section speed V_{1max}、V_{2max}：第 1 段、第 2 段指令速度 第1段运行时间为 t1+t2，第2段运行时间为 t3+t4，以此类推 The first run time is T1 + T2, the second run time is T3 + T4, and so on
cycle operation $2012_h:01_h$ ($02E9_h$)= 1	End segment set from paragraph 1 to $2012_h:02_h(02EA_h)$; Run the number of times set in $2012_h:03_h(02EB_h)$; The segment number is automatically incremented and switched. Multi-segment enable is level effective.	 V_{1max}、V_{2max}：第 1 段、第 2 段指令速度 The first and second instruction speed 第1段运行时间为 t1+t2，第2段运行时间为 t3+t4，以此类推 The first segment runs for t1 + t2, the second segment runs for t3 + t4, and so on
DI switching operation $2012_h:01_h$ ($02E9_h$)= 2 DI switch operation $2012_h:01_h$(02 $E9_h$) = 2 **Note: The physical terminal and the virtual terminal can be used in combination,**	The speed segment number switching is determined by the decimal 0~15 corresponding to the 4-bit binary number composed of the multi-function numbers 6, 7, 8, 9; Enable effective and sustainable operation; The running time of each speed command is only determined by the interval time of speed	 V_{xmax}、V_{ymax}、V_{zmax}：第 x 段、第 y 段、第z段指令速度 Section x, y, Z instruction speed

<table>
<tr><th>Operation mode</th><th>Explanation</th><th>Running curve</th></tr>
<tr><td rowspan="2">and the combination must be 4 bits; for example: Physical terminal selection DI1=6, DI2=7, virtual terminal VI1=8, VI2=9</td><td>segment number switching. Multi-segment enable is effective for edge change.</td><td></td></tr>
<tr><td>Correspondence table of the relationship between segment number and multi-function number</td><td>
<table>
<tr><th>Fun9</th><th>Fun8</th><th>Fun7</th><th>Fun6</th><th>Segment 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>

5.17.5 Configuration Examples

Multi-step speed-single operation control

- Parameter setting:

Set to NiMotion Velocity mode 2002_h:01_h($00B1_h$)=2;
Send：01 10 00 B1 00 01 02 00 02(Set to NiMotion speed mode)
Set the speed command source to multi-step speed: 2006h:02h(0149_h)=3, 2006h:03h($014A_h$)=1;
Send：01 10 01 49 00 01 02 00 03（et the source of the auxiliary speed command as multi-segment speed）
Send：01 10 01 4A 00 01 02 00 01（Set the primary speed command as the auxiliary speed reference）
Set multi-step speed function parameter group:
Multi-step speed running mode 2012_h:01_h ($02E9_h$)= 0 (single run);
Send：01 10 02 E9 00 01 02 00 00（Set the multi-segment speed running mode to single run）
Operating parameters of the 1st stage speed:
Speed command 2012_h:$0C_h$ ($02F4_h$)= 500 (unit rpm), command running time 2012_h:$0D_h$($02F5_h$) = 2000 (unit ms),
Send：01 10 02 F4 00 01 02 01 F4（Set the speed of the first segment to 500rpm）
Send：01 10 02 F5 00 01 02 07 D0（Set the first speed running time to 2000ms）
Acceleration and deceleration time 2012_h:$0E_h$ ($02F6_h$)= 500 (unit ms);
Send：01 10 02 F6 00 01 02 01 F4（Set the first speed acceleration and deceleration time to 500ms）
The operating parameters of the 2nd to 16th segments refer to the operating parameter settings of the 1st speed;
Set the motor enable: 2003_h:03_h($00D5_h$)=1, the logic is low level effective: 2003_h:04_h($00D6_h$)=0;
Send：01 10 00 D5 00 01 02 00 01（Set the motor enable）
Send：01 10 00 D6 00 01 02 00 00（Set DI1 logic to active low level）
Set multi-stage position/speed enable: 2003_h:05_h($00D7_h$)=28, logic is low level effective: 2003_h:06_h($00D8_h$)=0;
Send：01 10 00 D7 00 01 02 00 1C（Set the DI2 function to multi-segment enable）
Send：01 10 00 D8 00 01 02 00 00（Set DI2 logic to active low level）

- Run

Motor enable: DI1 input low level;

Multi-stage position/speed enable, DI2 input low level, the motor operates according to the operation mode set by the 2012_h ($02E9_h$ ~0323_h)multi-stage speed function parameter group and the operating parameters of each stage.

6 Special Functions

6.1 Digital Input and Output

The integrated low-voltage motor has input and output interfaces. The integrated digital input function is used to connect sensors for auxiliary motion control, such as limit micro switches, photoelectric switches, Hall sensors, etc. The digital output function can output 8mA current to drive solid state relays or indicator lights.

6.1.1 Digital Input

Related object settings

Table 6-1

Object Index	Description
2003_h（00D5h ~00E6h）	Physical input terminal function configuration
2017_h（0326_h ~0346_h）	Virtual input terminal function configuration
2004_h（00F8h ~0101_h）	DX1 terminal function selection
200C：07_h(0235_h))	Virtual digital input enable, 0-disable, 1-enable
2031_h：01h(0373_h)	Communication given VDI virtual level
$200B_h$：05_h（$01E2_h$）	The actual level status corresponding to the physical terminal (bit0-DI1~bit3-DI4)

Control instruction

There are at most 3 digital input physical terminals of this equipment: fixed input DI1~3;

Communication virtual input terminal, set and used by 2017_h (0326h ~0346h)parameter group. 2031_h: 01_h (0373_h)is the state of setting the corresponding level of the virtual terminal for communication (bit0-VDI1~bit15-VDI16).

6.1.1.1 Physical Terminals

Special function selection

The specific values are as follows:

Table 6-2

Function number	Function definition
0	Undefined
1	Enable
2	Alarm reset
6	Multi-stage running command switching 1
7	Multi-stage running command switching 2
8	Multi-stage running command switching 3
9	Multi-stage running command switching 4
12	Pause
14	Forward Overtravel Switch
15	Reverse overtravel switch
20	Step enable

Function number	Function definition
28	Multi-stage position/speed enable
31	Home switch
33	Set home
36	Analog input position or speed control direction
38	Clear fault history
39	Clear power-on 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 ratio input direction (fixed DI2)

Terminal logic selection

The multi-function terminal logic selection is set and used by the parameter group 2003_h ($00D5_h$ ~$00E6_h$) (entity input terminal). The specific values are as follows:

Table 6-3

Set value	Logical definition
0	Active low
1	Active high
2	Valid on rising edge
3	Valid on falling edge
4	Both rising and falling edges are valid

Note:

1、"1-Enable" is only valid in NiMotion mode, which is equivalent to writing enable control word for 6040_h（0380_h）in 402 mode.

2、The same function number cannot be set for different terminals. If the setting is wrong, a parameter error fault will be reported.

3、If the working mode is set to the HM mode of CiA402, the parameter failure will also be reported if the function numbers of 14, 15, and 31 are not set.

4、The terminal function and logic selection are set to be effective after shutdown.

5、The digital input is only detected once every millisecond. Unable to handle signal changes of less than 1ms at the input.

6、If the corresponding function number of the detection physical terminal is set to 46, 47 and 48, the initial angle of the motor 2000h: 1Eh will be automatically reset to 0. The first enable will make the motor automatically enter the state of finding the initial angle of the motor (402 In the mode, the status machine will return to the Switch on disable state, and the status word needs to be entered into the Operation enabled state again through the control word; when an external incremental encoder is used, the initial angle of the motor needs to be found for the first time the motor is powered on, otherwise the motor cannot run normally.

Examples

Next configure the physical input terminal DI1 as a positive limit switch

- Configure DI1 as a positive limit switch, 2003_h: 03_h($00D5_h$) = 14

Send：01 10 00 D5 00 01 02 00 0E(Set the DI1 function to the positive limit switch)

- Logic selection is low level effective, 2003_h: 04_h($00D6_h$) =0

  Send：01 10 00 D6 00 01 02 00 00(Set DI1 logic to active low level)

6.1.1.2 Virtual Terminal

Virtual VDI terminal special function definition

The specific values are as follows:

Table 6-4

Function Number	Function Definition
0	Undefined
1	Enable
2	Alarm reset
6	Multi-stage running command switching 1
7	Multi-stage running command switching 2
8	Multi-stage running command switching 3
9	Multi-stage running command switching 2
12	Pause
14	Forward Overtravel Switch
15	Reverse overtravel switch
20	Step enable
21	Command of given speed direction in Nimotion Velocity mode
28	Multi-stage position/speed enable
31	Home switch
33	Set zero
36	Analog input position or speed control direction
38	Clear fault history
39	Clear power-on time

Note:

1、"1-step enable" and "12-time out" is only valid in NiMotion mode, which is equivalent to writing the enable control word to 6040h (0380_h)in 402 mode.

Virtual VDI terminal logic selection

Multi-function terminal logic selection is set and used by 2017_h (0326_h ~0346_h) (virtual input terminal) parameter group. The specific values are as follows:

Table 6-5

Set Value	Logic Definition
0	Active high
1	Valid on rising edge

Enable

The virtual VDI terminal of the integrated low-voltage motor is effective when the index $200C_h$: 07_h(0235_h) is 1, and the value of $200C_h$: 07_h is 0 to disable the virtual DI terminal function. It is enabled

through the 2017_h (0326_h ~0346_h)sub-index input terminal function number and logic selection. When the logic selection is valid, the function of the corresponding function number is valid. Index 2031_h: 01_h(0373_h) is the corresponding state of the virtual input terminal (bit0-VDI1~bit15-VDI16), which can only be given through the communication method, the specific definition is as follows:

2031_h：01_h(0373_h)

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

Examples

Next, configure the virtual input terminal DI1 as a positive limit switch

- Configure DI1 as a positive limit switch, 2017_h: 01_h (0326_h)=14

Send：01 10 03 26 00 01 02 00 0E(Set the VDI1 function to the positive limit switch)

- Logic selection is low level effective, 2017_h: 02_h (0327_h)=0

Send：01 10 03 27 00 01 02 00 00(Set VDI1 logic to active low level)

6.1.2 Digital Output

Related object settings

Table 6-6

2004_h（$00F8_h$~0101_h）	Output terminal configuration parameters
2031_h：02h(0374_h)	Communication given VD0 output status (bit0~Do)

Control instruction

There are two digital output physical terminals of this device, and the configuration object 2004_h（$00F8_h$~0101_h）needs to select the output terminal function and logic level. The output status can also be given through object 2031_h:02_h (0374_h)communication.

Output terminal function selection:

Table 6-7

Function Number	Function Definition
0	Undefined
1	Ordinary DO port
2	Motor stops running
3	Goal reached
4	Alarm Output
5	Brake output
6	External braking resistor function
30	Braking Regulator Energy-saving Drive Output function

Attention：One channel of DO can used to External braking resistor function and Braking Regulator Energy-saving Drive Output function.

Logic selection of output terminal:

Table 6-8

Set Value	Logical Definition
0	Active low
1	Active high

Examples

Example 1: Configure DO1 as normal control function and output switch on state:

- Configure DO1 as output, 2004_h: 01_h($00F8_h$) =1;
  Send：01 10 00 F8 00 01 02 00 01(set DO1 as output)
- Configure DO1 logic to be high level effective, 2004_h: 02_h ($00F9_h$)=1;
  Send：01 10 00 F9 00 01 02 00 01(Set DO1 logic to high level valid)
- Communication given DO1 output level, 2031_h: 02_h (0374_h)= 0x01, at this time DO1 output switch on state.
  Send：01 10 03 74 00 01 02 00 01(Set the DO1 output to output level)

6.2 Analog Input

The realization of the analog quantity control method requires voltage input through the analog input port, the voltage input range is 0-10V; through the configuration parameters, set the corresponding quilt of 0-10V Control range; analog input control includes analog speed control, analog torque control, and analog position control;

6.2.1 Analog Position Control

Related objects

Table 6-9

Object Index	Description
2002_h:01_h ($00B1_h$)	Operation mode selection
2005_h:01_h($010D_h$)	Position command source, 3-mode 1 analog position control, 4-mode 2 analog position control
2005_h:$2A_h$(0140_h)	AI1 dead zone judgment time window, unit 1ms
2005_h:$2C_h$(0142_h)	Analog value 10V corresponding position value (encoder unit)
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input")
2003_h:17_h($00E9_h$)	AI1 bias, unit mV
2003_h:18_h($00EA_h$)	AI1 input filter time constant, unit ms
2003_h:19_h($00EB_h$)	AI1 dead zone, unit mV, default 100mV
2003_h:$1A_h$($00EC_h$)	AI1 ratio, unit 0.001

Enable

There is no state machine management in this mode, and the motor can run through the terminal.

The prerequisite for the operation of this mode: the integrated low-voltage servo motor is switched to the NiMotion position mode write 1 in 2002_h:01_h($00B1_h$).

1. Location source selection:
   Set the position command source 2005_h:01_h($010D_h$), meaning: 3-mode 1 analog position control, 4-mode

2 analog position control;

2. Set up multi-function terminals (both physical terminals and virtual terminals are applicable, see "6.1 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 set position command source is 3, you need to set the corresponding input terminal DI1 or DI2 as the analog input position control direction function (multifunction number 36).

3. Position setting
   Configure the position value 2005_h:$2C_h$(0142_h) corresponding to analog 10V, which is the absolute position of the motor.

4. Run
   The motor is enabled through a terminal, and the motor runs in the corresponding position according to the analog input.

Note: The analog position control is absolute position control, the current position needs to be cleared to 0 before use.

Analog position control mode

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

1. Mode 1:
   The given position = sign (corresponding to the level of multi-function number 36) * analog value 10V corresponding position value * (input voltage * AI1 magnification + AI1 offset)/10000, multi-function number 36 is valid for negative direction, invalid Is the positive direction.

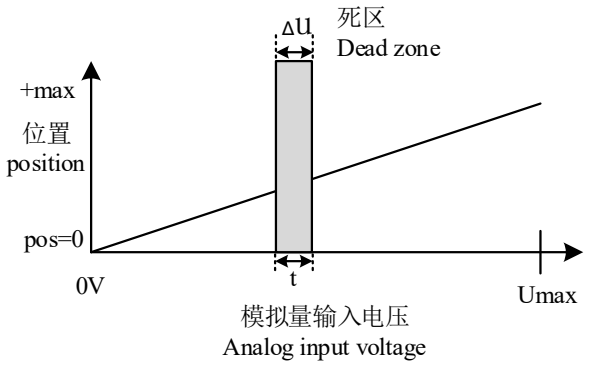


Figure 6-1

2. Mode 2:
   Input voltage 5~10V: the given position = sign (corresponding to the level of multi-function number 36) *the corresponding position value of the analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000;
   Input voltage 0~5V: the given position = - sign (corresponding to the level of multi-function number 36) *the corresponding position value of analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000;

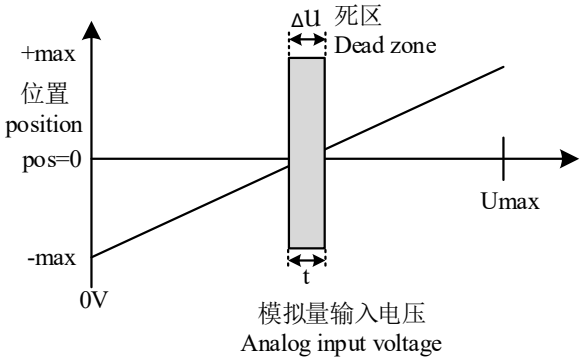


图 Figure 6-2

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

- Parameter setting

Set to NiMotion position mode: 2002_h:01_h($00B1_h$)=1;
Send：01 10 00 B1 00 01 02 00 01(Set to NiMotion position mode)
Set the source of position command to 3: 2005_h: 01_h($010D_h$)=3;
Send：01 10 01 0D 00 01 02 00 03(Set the source of position command to 3)
Set the motor enable: 2003_h:03_h($00D5_h$)=1, the logic is active low: 2003_h:04_h($00D6_h$)=0;
Send：01 10 00 D5 00 01 02 00 01（Set the DI1 function to step-enable）
Send：01 10 00 D6 00 01 02 00 00（Set DI1 logic to active low level）
Set the analog input position control direction: 2003_h:05_h($00D7_h$)=36, the logic is valid at the falling edge: 2003_h:06_h($00D8_h$)=3;
Send：01 10 00 D7 00 01 02 00 24（Set DI2 function to analog input position control direction）
Send：01 10 00 D8 00 01 02 00 03（Set the DI2 logic to be valid as the falling edge）
2005_h: $2C_h$=10000;
Send：01 10 01 42 00 02 04 00 00 27 10(Set position value corresponding to 10V)
The AI interface is connected to the analog voltage value (note that the positive and negative are not reversed), and the position is given by adjusting the voltage;

- Run

DI1 inputs low level, the motor runs according to the position given by the analog quantity, and DI2 controls the direction of motor rotation.

6.2.2 Analog Speed Control

Related objects

Table 6-10

Object Index	Description
2002_h:01_h ($00B1_h$)	Operation mode selection

Object Index	Description
2006_h:01_h (0148_h)	NiMotion speed mode main speed reference source, 1-mode 1 analog speed control
2006_h:02_h (0149_h)	NiMotion speed mode auxiliary speed reference source, 1-mode 2 analog speed control
2006_h:03_h ($014A_h$)	NiMotion speed mode speed command selection, 0-main speed reference, 1-auxiliary speed reference
2006_h:04_h ($014B_h$)	Speed command digital given value, unit rpm
2006_h:07_h ($014E_h$)	Speed command acceleration ramp time constant, in ms, representing the time it takes to accelerate from 0rpm to 1000rpm.
2006_h:08_h ($014F_h$)	Speed command deceleration ramp time constant, unit ms
2006_h:$0A_h$ (0151_h)	Forward speed threshold, unit rpm
2006_h:$0B_h$ (0152_h)	Forward speed threshold, unit rpm
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input")
2003_h:17_h($00E9_h$)	2003_h:17_h: AI1 bias, unit 1mV
2003_h:18_h($00EA_h$)	2003_h:18_h: AI1 input filter time constant, unit 0.01ms
2003_h:19_h($00EB_h$)	2003_h:19_h: AI1 dead zone, unit mV, The default value is 100mV
2003_h:$1A_h$($00EC_h$)	2003_h:$1A_h$: AI1 ratio, unit 0.001
2003_h:$1F_h$($00F1_h$)	The corresponding speed value of analog 10V, unit rpm

Enable

There is no state machine management in this mode, and the motor can run through the terminal.

Operation premise: the integrated low-voltage servo motor is switched to NiMotion speed mode 2002 $_h$:01_h($00B1_h$) is set to 2.

1. Selection of speed source:
   Set the source of the main speed command 2006_h:01_h (0148_h)and the source of the auxiliary speed command 2006_h:02_h (0149_h)to 1, meaning: 1-analog speed control;
   Through the speed command selection 2006_h:03_h($014A_h$), the actual effective is the main speed command (mode 1 analog speed control) or the auxiliary speed command (mode 2 analog speed control), meaning: 0-main speed reference, 1- auxiliary speed reference ；

2. Set up multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 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 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 as the analog input speed control direction function (multi-function number 36);
3. Speed, acceleration and maximum speed setting
   If the speed command selection 2006_h:03_h($014A_h$)is 0, the corresponding speed value of analog 10V 2003_h:$1F_h$ ($00F1_h$)needs to be configured;
   If the speed command selection 2006_h:03_h($014A_h$) is 1, you need to configure the forward and reverse speed thresholds 2003_h:$0A_h$ (0151_h)and 2003_h:$0B_h$(0152_h);
   Configure acceleration and deceleration 2006_h:07_h ($014E_h$)and 2006_h:08_h($014F_h$);
4. Run
   The motor is enabled through a terminal, and the motor runs at a corresponding speed according to the analog input.

Analog speed control mode

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

1. Mode 1:

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

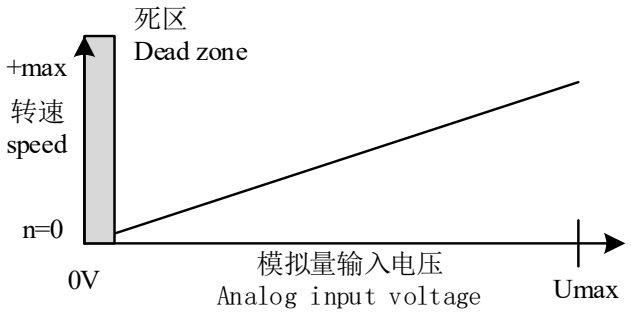


Figure 6-3

2. Mode 2:
   Input voltage 5~10V: the given speed = sign (corresponding to the level of multi-function number 36) * the corresponding speed value of analog 10V * (input voltage * AI1 ratio + AI1 bias)/10000;
   Input voltage 0~5V: the given speed = - sign (corresponding to the level of multi-function number 36) * the corresponding speed value of the analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000;

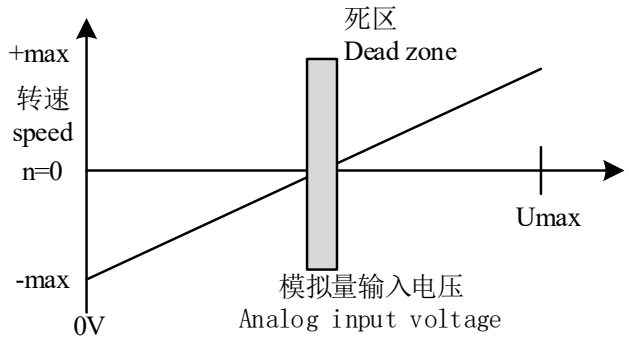


Figure 6-4

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

  Set to NiMotion speed mode: 2002_h:01_h($00B1_h$)=2;

  Send：01 10 00 B1 00 01 02 00 02(Set to NiMotion speed mode)

  Set the main speed command source as 1: 2006_h:01_h(0148_h)=1, set the speed command selection 2006_h:03_h($014A_h$)=0;

  Send：01 10 01 48 00 01 02 00 01 (Set the main speed command source as 1)

  Send：01 10 01 4A 00 01 02 00 00（Set the speed instruction as the primary speed instruction）

  Set the acceleration ramp time constant to 100ms: 2006_h:07_h($014E_h$)=100 (unit ms);

  Send：01 10 01 4E 00 01 02 00 64(Set the acceleration ramp time constant to 100ms)

  Set the deceleration ramp time constant to 100ms: 2006_h:08_h($014F_h$)=100 (unit: ms);

  Send：01 10 01 4F 00 01 02 00 64(Set the deceleration ramp time constant to 100ms)

  Set the corresponding speed value of analog 10V, 2003_h:$1F_h$($00F1_h$)=3000 (unit rpm);

  Send：01 10 00 F1 00 01 02 0B B8(Set the corresponding speed value of simulation 10V to 3000rpm)

  Set DI1 and DI2 terminals:

  DI1 function selection is servo enable: 2003_h:03_h ($00D5_h$)=1;

  Send：01 10 00 D5 00 01 02 00 01（Set DI1 function as servo enable）

  DI1 logic selection is active low: 2003_h:04_h ($00D6_h$)=0;

  Send：01 10 00 D6 00 01 02 00 00(Set DI1 logic to active low level)

  DI2 function selection is the analog input speed control direction: 2003_h:05_h ($00D7_h$)=36;

  Send：01 10 00 D7 00 01 02 00 24(Set DI2 function as the analog input speed control direction)

  DI2 logic selection is active low: 2003_h:06_h ($00D8_h$)=0;

  Send：01 10 00 D8 00 01 02 00 03(Set DI2 logic to active low level)

  The AI interface is connected to the analog voltage value (note that the positive and negative are not reversed), which is used to adjust the voltage speed;

- Run:

  DI1 inputs low level, the motor runs at the speed given by the analog quantity 2006h:06h($014B_h$), and DI2 controls the direction of motor rotation.

6.2.3 Analog torque control

Related objects

Table 6-11

Object Index	Description
2002_h:01_h($00B1_h$)	Operation mode selection
2007_h:01_h($015E_h$)	NiMotion torque mode main torque reference source, 1-mode 1 analog torque control;
2007_h:02_h($015F_h$)	2007_h:02_h: NiMotion torque mode auxiliary torque reference source, 1-mode 2 analog torque control;
2007_h:03_h(0160_h)	NiMotion torque mode torque command selection, 0-main torque reference, 1-auxiliary torque reference;
2007_h:04_h(0161_h)	Torque command digital given value (0.1%)
2007_h:$0A_h$(0167_h)	2007_h:$0A_h$: Positive internal torque limit (0.1%)
2007_h:$0B_h$(0168_h)	Reverse internal torque limit (0.1%)
2007_h:10_h($016D_h$)	Torque control forward speed limit value (rpm)
2007_h:11_h($016E_h$)	Torque control reverse speed limit value (rpm)
2017_h (0326_h ~0346_h)	Virtual input terminal function configuration (see: "6.1.1 Digital Input")
2003_h:17_h($00E9_h$)	AI1 bias, unit mV
2003_h:18_h($00EA_h$)	AI1 input filter time constant, unit 0.01ms
2003_h:19_h($00EB_h$)	AI1 dead zone, unit mV
2003_h:$1A_h$($00EC_h$)	2003_h:$1A_h$: AI1 ratio, unit 0.001
2003_h:20_h($00F2_h$)	Torque value corresponding to analog 10V, unit 0.1%

Enable

There is no state machine management in this mode, and the motor is enabled to run through the terminal. Operation premise: the integrated low-voltage servo motor is switched to NiMotion torque mode 2002 $_h$:01_h ($00B1_h$)is set to 3.

1. Torque source selection:
   Set the main torque command source 2007_h:01_h($015E_h$)and the auxiliary torque command source 2007_h:02_h($015F_h$)to 1, meaning: 1-analog torque control;

   Through the torque command selection 2007_h:03_h(0160_h), the actual effective is the main torque command (mode 1 analog torque control) or auxiliary torque command (mode 2 analog torque control), meaning: 0-main torque reference, 1- Auxiliary torque setting;

2. Set up multi-function terminals (both physical terminals and virtual terminals are applicable, please refer to "6.1 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 torque command selection 2007_h:03_h (0160_h)set in the previous step is 0, you need to set the corresponding input terminal DI1 or DI2 as the analog input torque control direction function (multi-function number 36);

3. Torque setting and torque and speed limit setting

   If the torque command selection 2007_h:03_h(0160_h) is 0, the corresponding torque value 2003_h:20_h ($00F2_h$)of analog 10V needs to be configured;

   If the torque command selection 2007_h:03_h(0160_h) is 1, you need to configure the positive and negative internal torque limit values 2007_h:$0A_h$ (0167_h)and 2007_h:$0B_h$(0168_h);

   Configure the forward and reverse speed limit values 2007_h:10_h($016D_h$) and 2007_h:11_h($016E_h$);

4. Run

   The motor is enabled through a terminal, and the motor performs corresponding torque control according to the analog input.

Analog torque control mode

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

1. Mode 1:

   The given torque = sign (corresponding to the level of multi-function number 36) * analog quantity 10V corresponding torque value * (input voltage * AI1 ratio + AI1 bias)/10000, note: multi-function number 36 is valid Negative direction, invalid as positive direction.

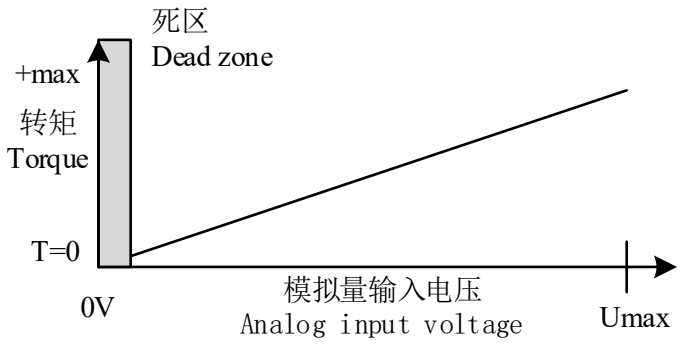


Figure 6-5

2. Mode 2:

   Input voltage 5~10V: the given torque = sign (corresponding to the level of multi-function number 36) * the corresponding torque value of analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000;

   Input voltage 0~5V: the given torque = - sign (corresponding to the level of multi-function number 36) *the corresponding torque value of the analog 10V * (input voltage * AI1 multiplier + AI1 bias)/10000;

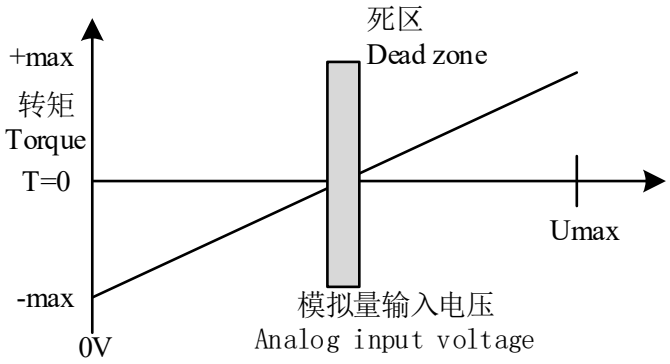


Figure 6-6

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

  Set to NiMotion torque mode: 2002_h:01_h($00B1_h$)=3;

  Send：01 10 00 B1 00 01 02 00 03（Set to NiMotion torque mode）

  Set the main torque command source to 1: 2007_h: 01_h($015E_h$)=1;

  Send：01 10 01 5E 00 01 02 00 01（Set the main torque command source to 1）

  Set the torque command to select the main torque command: 2007_h:03_h(0160_h)=0;

  Send：01 10 01 60 00 01 02 00 00(Set the torque instruction as the main torque instruction)

  Set the positive and negative internal torque limit to 100%: 2007_h:$0A_h$(0167_h)=1000 and 2007_h:$0B_h$($016E_h$)=1000 (unit: 0.1%);

  Send：01 10 01 67 00 01 02 03 E8（Set the positive internal torque limit to 100%）

  Send：01 10 01 68 00 01 02 03 E8（Set the negative internal torque limit to 100%）

  Set the forward and reverse speed limit value to 3000rpm: 2007_h:10_h($016D_h$)=3000 and 2007_h:11_h($016E_h$)=3000 (unit rpm);

  Send：01 10 01 6D 00 01 02 0B B8（Set the forward speed limit value to 3000rpm）

  Send：01 10 01 6E 00 01 02 0B B8（Set the reverse speed limit value to 3000rpm）

  Set the torque value corresponding to the analog 10V, 2003_h:20_h($00F2_h$)=1000 (unit: 0.1%);

  Send：01 10 00 F2 00 01 02 03 E8(Set the analog 10 v corresponding torque value is 1000)

  Set DI1 and DI2 terminals:

  DI1 function selection is servo enable: 2003_h:03_h ($00D5_h$) =1;

  Send：01 10 00 D5 00 01 02 00 01(Set DI1 function as servo enable)

  DI1 logic selection is active low: 2003_h:04_h($00D6_h$) =0;

  Send：01 10 00 D6 00 01 02 00 00(Set DI1 logic to active low level)

  DI2 function selection is the analog input torque control direction: 2003_h:05_h($00D7_h$)=36;

  DI2 logic selection is active low: 2003_h:06_h ($00D8_h$)=0;

  Send：01 10 00 D7 00 01 02 00 24（Set DI2 function as the analog input speed control direction）

  Send：01 10 00 D8 00 01 02 00 03（Set DI2 logic to active low level）

- Run

  DI1 inputs low level, the motor runs according to the torque given by the analog quantity 2007_h:04_h(0161_h), and DI2 controls the direction of motor rotation.

6.2.4 Analog switching function

When the control bus communication is abnormal, the motor cannot respond to the control command sent by the main control system normally, and the main control system cannot recognize the operation status of the controlled motor, the controlled motor reports communication failure and stops running, and the indicator shows the status of failure alarm. If the application requires that the motor output speed or position can still be controlled in case of communication interruption, the system control mode needs to be switched from bus communication control to analog control of the physical terminal, i.e., the controlled output size is controlled by adjusting the analog voltage size. So the analog switching function is suitable for this application scenario.

- **PV mode switch to analog velocity control function**

The steps are as follows：

1. Select the analog velocity mode (mode 1 or mode 2) to be configured, and set it through the data dictionary 2006_h-03_h ($014A_h$)parameters.
2. Set the analog 10V corresponding speed value, i.e. 0~10V corresponding to the set velocity range.
3. Set analog velocity switching function number 50, i.e. 2003_h-03_h ($00D5_h$)is set to 50, low level is valid.
4. Turn on the analog input voltage of AI port.
5. In the stop state, when DI1 input level is valid, the motor will automatically switch to analog velocity mode, adjust the analog voltage value, and the motor velocity will change.

The specific settings of the above parameters related to analog velocity control can be referred to subsection 6.2.2.

- **PP mode switching to analog position control function**

The steps are as follows：

1. Select the analog position mode (mode 1 or mode 2) to be configured, and set it through the data dictionary 2005_h-01_h ($010D_h$)parameters.
2. Set the analog 10V corresponding position value.
3. Set the analog position switching function number 51, that is, 2003_h-03_h ($00D5_h$)is set to 51, low level active.
4. Turn on the analog input voltage of AI port.
5. In the stop state, when the DI1 input level is valid, the motor will automatically switch to the analog position control mode, adjust the analog voltage value, and the motor position will change. (Note: the analog position is given as absolute position, please determine the current position value before switching)

Please refer to section 6.2.1 for the specific settings of the above parameters related to analog position control.

Note: The analog switching function is effective in the motor stop state! If the switchover is made under the communication fault, the communication fault needs to be configured as a fault stop; the virtual terminal function number cannot be configured as the enable function number, otherwise the switchover function is invalid!

6.3 Brake Setting

The holding brake prevents the motor shaft from moving when the servo motor is in the disabled state, thereby locking the position of the motor. The integrated low-voltage motor automatically switches the state of the brake

output according to the CiA402 state machine, as shown below:

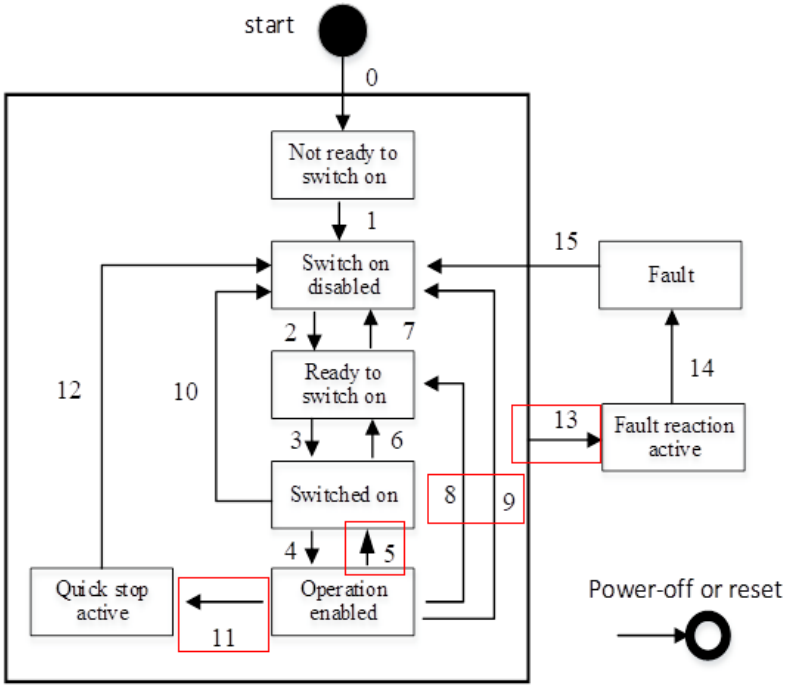


Figure 6-7

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

Table 6-12

Bit	Name	Description
0	Motor ready	When the status word Bit0~bit3 are all 1, the brake output is turned on.
1	Connect the main circuit power	
2	Fast shutdown	
3	Motor running	

Table 6-13

Object Index	Description
2002_h: $0A_h(00BA_h)$	Delay time when the brake is turned off until the motor is powered off, Unit: ms
$200A_h$: $1F_h(01D8_h)$	Brake detection function,0- Disable gravity free preset,1- Enable gravity preset
$200A_h$: $20_h(01D9_h)$	Brake load gravity value(Preset gravity value, The torque output of the motor when the brake is on),Unit: 0.1% rated torque
$200D_h$: $0B_h(0254_h)$	Brake load identification，0- Gravity recognition is not enabled；1- Online gravity identification was enabled

Note:

1. When the motor is used for vertical shaft, the function of gravity preset and online gravity identification should be enabled to prevent the brake release and the load drop when the motor is started due to the gravity preset value or external force.

2. When the motor is used for vertical shaft, it is necessary to set the delay time from the brake output to the time when the motor is not powered, to prevent the motor from running to stop or failure to stop, the load due to dead weight or external force drop.
3. Enable online real-time identification of load gravity. When the servo motor speed is below 300rpm, the preset gravity value of $200A_h$:20_h($01D9_h$) is automatically updated.

Enable and disable

When the terminal function of DX1 2004_h:01_h($00F8_h$)is configured as multifunction number 5 (motor brake status output) and the logic is active low。DO is a switch with no voltage, should be installed in series of delay circuit.

Whether the brake is internal or external, the brake coil will close after the motor is disabled.

6.4 I^2t Motor Overload Protection

After the integrated low-voltage motor is energized, it will continue to generate heat due to the thermal effect of the current, and at the same time release heat to the surrounding environment. When the heat generated by the motor exceeds the released heat, the temperature of the motor rises, and the temperature is too high, which will cause the motor to burn out. The purpose of I2t motor overload protection is to ensure the normal operation of the motor within the temperature limit and prevent damage to the motor.

Related objects

Table 6-14

Object Index	Description
2000_h：09_h（0065_h）	Rated current (unit: 0.01A)
2000_h：$0A_h$（0066_h）	Maximum current (unit: 0.01A)
2000_h：$0B_h$（0067_h）	Maximum current duration (unit: ms)

Trigger mechanism

To enable this protection function, the above three object items must be explained correctly. When the motor runs above the rated current, the drive detects and calculates the time integral of I2. When the integral value is greater than the product of the set maximum current 2000_h: $0A_h$（0066_h） and the maximum current duration 2000_h: $0B_h$（0067_h）, the motor reports an overload Warning and limit torque output to rated value. If these conditions are not met, the I2t overload protection function will not be triggered.

6.5 Brake Settings

The integrated motor prevents the energy from being transmitted from the motor end back to the drive of the motor when the torque and speed of the motor are in opposite directions, causing the bus voltage to be too high and damaging the drive of the motor. The drive will determine when the bus voltage exceeds the discharge voltage. After 2001_h: $0F_h$(0093_h), the excess feedback energy is consumed by controlling the switch tube of the brake circuit. It can through 2002_h：13_h($00C3_h$) setting to chose whether use inner resistor or external resistor. When 2002_h：13_h($00C3_h$)=0, means to use inner resistor; When 2002_h：13_h($00C3_h$)=1, means to use external resistor; inner resistor and external resistor can not be used at the same time. And the supply power must be restart to make sure that the set change be effective. The default brake setting is inner resistor.

The external braking resistor function is realized by configuring the DO terminal function. When the DO

terminal is configured as function number 6, it means that the external braking resistor is used for the function realization，then braking circuit must be configured in outside of DO.

The following table shows the rotor inertia and braking power P0 corresponding to the integrated brushless motor model.

Table 6-15

Integrated motor model	Rotor inertia J (10^{-4}kgm^2)	Braking power P_0 (W)
BLM4203	0.14	5.4

The actual required braking resistor power P is related to the period T (s) of the reciprocating motion, the motor speed n (rpm), the rotor inertia J (kgm^2), and the load and motor inertia ratio N. The specific calculation formula is as follows:

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

Taking BLM4210A as an example, assume that the reciprocating motion period T=2s, the maximum speed is 3000rpm, and the load inertia is 5 times the motor inertia. The specific speed curve is as follows:

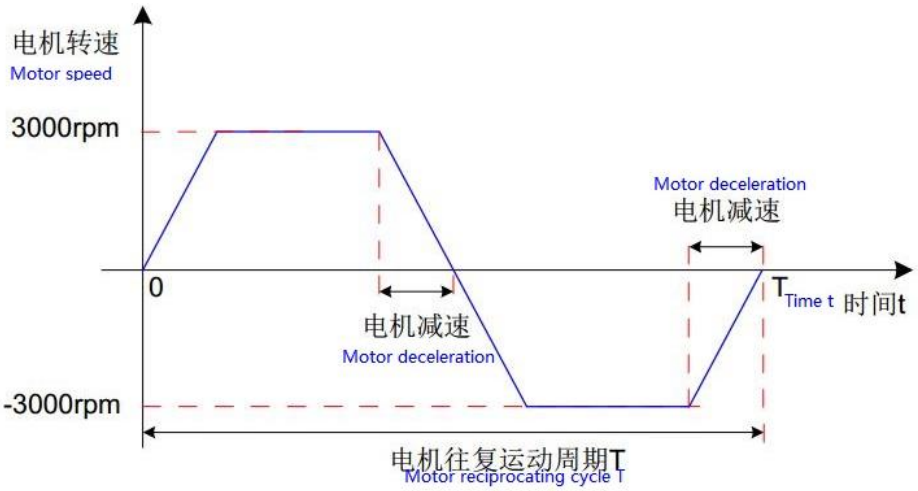


Figure 6-8

The braking resistor power is required:

$$P_1 = \frac{2*(N+1)*J*n*n}{182*T} = \frac{2\times(5+1)\times0.14*0.0001*3000*3000}{182*2}=4.15W$$

P_1 is less than the braking power P_0 (5.4W) that the integrated brushless motor can handle. At this time, the integrated brushless motor can meet this deceleration (or negative acceleration).

If the load inertia in the above assumptions is changed from 5 times to 10 times, and other conditions remain unchanged, the braking resistor power P_2 is required:

$$P_1 = \frac{2*(N+1)*J*n*n}{182*T} = \frac{2\times(10+1)\times0.14*0.0001*3000*3000}{182*2}=7.62W$$

P2 is greater than P0 (5.4W). At this time, the integrated brushless motor can only be decelerated according to the deceleration corresponding to P0 (5.4W) to prevent voltage overcharge. It cannot meet the deceleration requirements of the user. It is recommended that the user adjust the deceleration size.

Note: Unless the DC side power supply allows energy feedback, this product is not suitable for use as a generator.

6.6 Parameter Saving and Restoration

The parameter management is written according to CiA301 standard, 1010_h(0026_h~$003A_h$) is the function object for saving parameters, and 1011_h ($003C_h$~0050_h)is the function object for reading parameters.

6.6.1 Parameter Saving

In the 1010_h(0026_h~$003A_h$) sub-index, a value of 1 indicates that the corresponding save operation is supported, 0x65766173 is the save parameter command, write the save parameter command word to the 1010_h(0026_h~$003A_h$) corresponding sub-index, return 1 to save the parameter success, and return a non-1 to unsuccessful.

For example: 1010_h: 01_h (0026_h)write 0x65766173 to save user parameters.

6.6.2 Parameter Recovery

In the sub-index of 1011_h, a value of 1 indicates that the corresponding recovery operation is supported. 0x64616F6C is recovery parameter commands. Write recovery parameter command words to the subindex of 1011_h. and returning 1 indicates a successful recovery parameter operation, while returning a non-1 value indicates failure.

For example: Writing 0x64616F6C to 1011_h:01_h ($003C_h$) restores the factory area parameters to the user area.

6.7 Location Recovery Function

After the motor is powered off and on again, the absolute position can be restored.

By setting the position recovery mode 2005_h:18_h($012A_h$) value to 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, return to the zero point Need to run in the negative direction.

Parameter 2005_h: 18_h ($012A_h$)position recovery mode, when the value is 0, it is multi-turn absolute value recovery, when the value is 1, it is the absolute position recovery within a single turn;

For the single-turn position recovery, the parameter 2005_h:$1D_h$($012F_h$) blocking position can be set to change the direction of zero return.

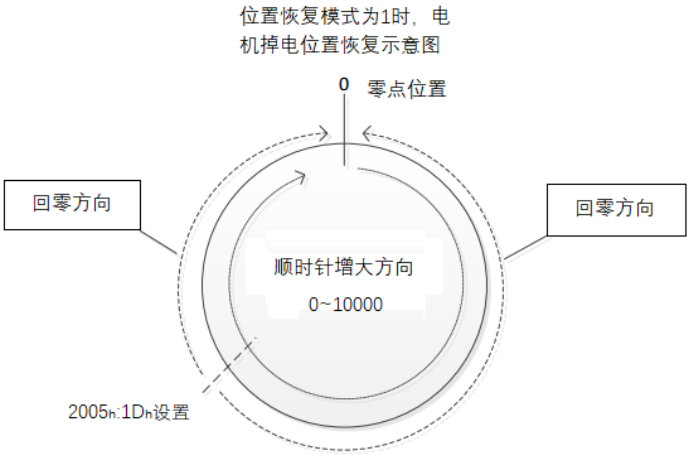


Figure 6-9

When the position recovery mode 2005_h:18_h ($012A_h$)value is 0, it is suitable for multi-turn motion trajectory, and the motor can recover the current absolute position.

6.8 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(0082_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_h (0064_h)Indicates the maximum power that can be output after the power limit is enabled. The highest bit (bit15) in 2000_h:24_h (0082_h)indicates 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_h(0064_h) in 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(0082_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(0064_h)=32845 (1000 0000 0100 1101). When the electric machine is in the power limiting state, the power limiting alarm will be performed.

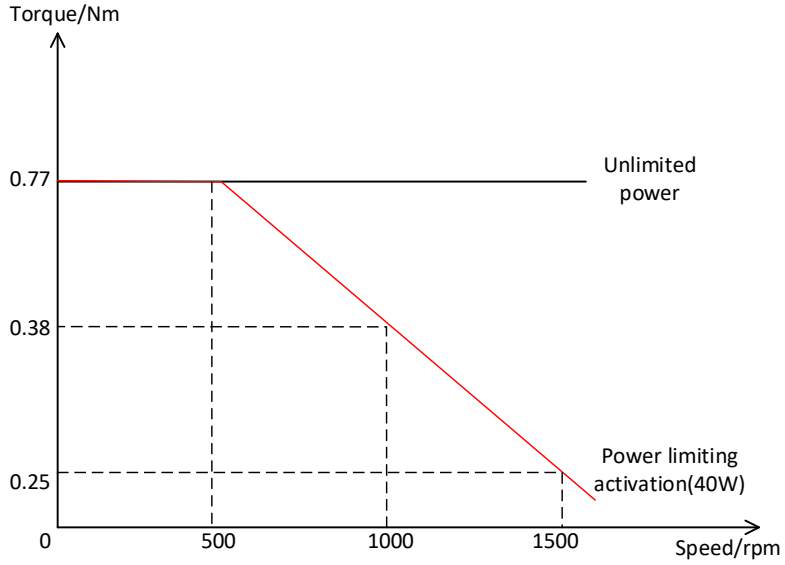


Figure 6-10 Power Limiting Activation Diagram

6.9 Absolute encoder

The integrated motor detects the battery voltage once each time it is powered on and updates it to the parameter 200F-$0B_h$ ($028C_h$). When the integrated motor detects that the battery voltage drops below the specified value during operation, the integrated motor releases a warning and fault information.

Table 6-16

Battery voltage	Alarm message	Alarm recovery	Battery replacement method
2.6V＞DC＞2.4V	0x7302	Multi-turn encoder counting error may occur; Recommended battery replacement	1. The integrated motor is powered on and in a non-running state; 2. Replace the battery in the battery installation box. 3. The alarm is automatically lifted.
DC≤2.4V	0x7301	Please replace the battery immediately	
DC≥2.6V	0x7309	Power off and restart	

6.10 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 H0249 = 0, this function is turned off. When H0249 ≠ 0, this function is activated, and the dog feeding time is the parameter value of H0249. 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 ID of the motor on the modbus bus is 1. Enable the watchdog function and set the watchdog time of the integrated motor to 100ms, then turn off the watchdog function.

Table 6-17

Serial number	Function	Message	Note
1	Modify the watchdog time of the target motor by 100ms	01 10 02 49 00 01 02 00 64	Effective immediately
2	Repeat feeding the dog	01 10 02 49 00 01 02 00 64	No alarm
3	Turn off the watchdog function	01 10 02 49 00 01 02 00 00	
4	Save parameters	01 10 00 26 00 02 04 65 76 61 73	
5	Restart takes effect		The watchdog function is turned off.

7 Adjustment

7.1 Overview

In order to maximize the performance of the integrated brushless motor, so that the device can quickly and accurately respond to commands from the host computer or internal settings, the gain must be adjusted reasonably according to the characteristics of the driving load. The gain includes speed loop gain, position loop gain, speed loop integration time, filtering, etc., and they affect each other, so when adjusting the gain parameters, the balance between each parameter must be considered.

Note: Before performing gain adjustment, it is recommended to perform a jog test run to confirm that the motor can operate normally; during the adjustment process, vibration of the motor may be accompanied, so please pay full attention to safety.

7.2 Manual Gain Adjustment

7.2.1 Related Objects

Table 7-1

Object Index	Description
2008_h：01_h (0178_h)	Speed loop gain (unit 0.1Hz), increase this parameter without noise and vibration, can speed up the positioning time, bring better speed stability and followability; if noise occurs, reduce the parameter setting value ;
2008_h：02_h (0179_h)	Speed loop integral time constant (unit: 0.01ms). Decreasing the set value can strengthen the integral function and speed up the positioning time. However, if the set value is too small, it may cause mechanical vibration.
2008_h：03_h ($017A_h$)	Position loop gain, increasing this parameter can speed up the positioning time and improve the ability of the motor to resist external disturbances when it is stationary. If the setting value is too high, the system may become unstable and oscillate.
2008_h：$0F_h$ (0186_h)	Speed feedforward filter time constant (unit: 0.01ms)
2008_h：10_h (0187_h)	Speed feedforward gain
2008_h：11_h (0188_h)	Torque feedforward filter time constant (unit: 0.01ms)
2008_h：12_h (0189_h)	Torque feedforward gain
2007_h：06_h (0163_h)	The torque command filters the time constant, eliminates high-frequency noise, and suppresses mechanical resonance. Increase When vibration occurs in 2008_h: 01_h, the vibration can be suppressed by adjusting 2007_h: 06_h. If the setting value is too large, the response of the current loop will decrease

7.2.2 Basic gain adjustment instructions

The integrated motor is composed of three control loops, which are the current loop, the speed loop and the position loop in order from inside to outside. The response frequency of the current loop is the highest. The factory default current loop gain parameters of the motor have ensured sufficient responsiveness, and generally no adjustment is required, so only the position loop gain, speed loop gain and speed loop integral time constant need

to be adjusted. In order to ensure the stability of the system, while increasing the position loop gain, it is necessary to increase the speed loop gain.

The basic gain adjustment is as follows:

Figure 7-2 Speed loop gain

Figure 7-3 Speed loop integral time constant

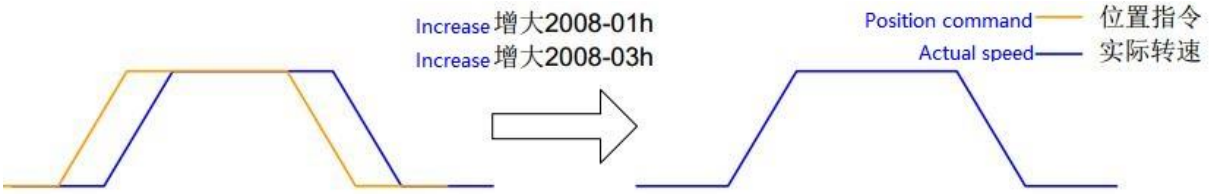


Figure 7-4 Position loop gain

Figure 7-5 Torque command filter time constant

7.2.3 **Parameter adjustment based on rigid tables**

The adjustment based on the rigidity table parameters 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), with the aim of quickly adjusting the stability of the integrated motor.

Table 7-3 Parameters updated by rigid table tuning mode

Rigid grade	**2008-03$_h$(017A$_h$) (Hz)**	**2008-01$_h$(0178$_h$) (0.1Hz)**	**2008-02$_h$(0179$_h$) (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

In no-load conditions, it is recommended that the PID parameter be set to the parameter corresponding to rigid level 12 or 13.

When parameters need to be adjusted, it is recommended to preferentially select PID parameters with a low rigidity level, and then increase the rigidity level step by step until the requirements are met.

When the parameters set according to the rigid table do not achieve the expected effect, the PID parameters can be fine-tuned. Optimize results through more detailed adjustments.

Note: Before applying rigid meter adjustment, please confirm whether the purchase of integrated motor supports this function before technical support.

7.2.4 Explanation of feedforward gain adjustment

Speed feedforward

Speed feedforward can improve the speed command response and reduce the position deviation at a fixed speed. It is mainly used in the full closed-loop position control mode. The basic adjustment method is as follows:

Set 2008_h: $0F_h$ (0186_h)as a fixed value; then, gradually increase the setting value of 2008_h: 10_h (0187_h)from 0 until a certain set value, speed feedforward to achieve the effect. When adjusting, you should repeatedly adjust 2008_h: $0F_h$(0186_h) and 2008_h: 10_h (0187_h)to find a well-balanced setting

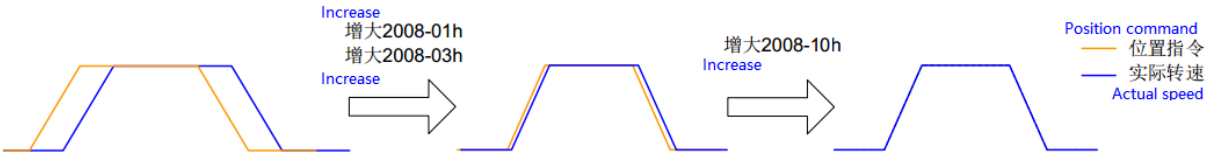


Figure 7-6 Speed feedforward

Torque feedforward

The position control mode, using torque feedforward, can improve the torque command response and reduce the position deviation during fixed acceleration and deceleration; the speed control mode, using torque feedforward, can improve the torque command response and reduce the fixed speed Speed deviation.

The basic adjustment method is as follows:

Set 2008_h: 11_h (0188_h)as a fixed value; then, gradually increase the setting value of 2008_h: 12_h (0189_h)from 0 until a certain set value, speed feedforward to obtain the effect. When adjusting, you should repeatedly adjust 2008_h: 11_h(0188_h) and 2008_h: 12_h(0189_h) to find a well-balanced setting.

7.2.5 Gain switching adjustment instructions

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

- Switch the motor running state to a higher gain to get better command tracking performance;
- Switch to a higher gain when the motor is at a standstill to shorten the positioning time;
- Switch to lower gain when the motor is at rest (enable) to suppress vibration;
- Switch different gain settings through external signals according to the load equipment conditions;

Gain switching flowchart

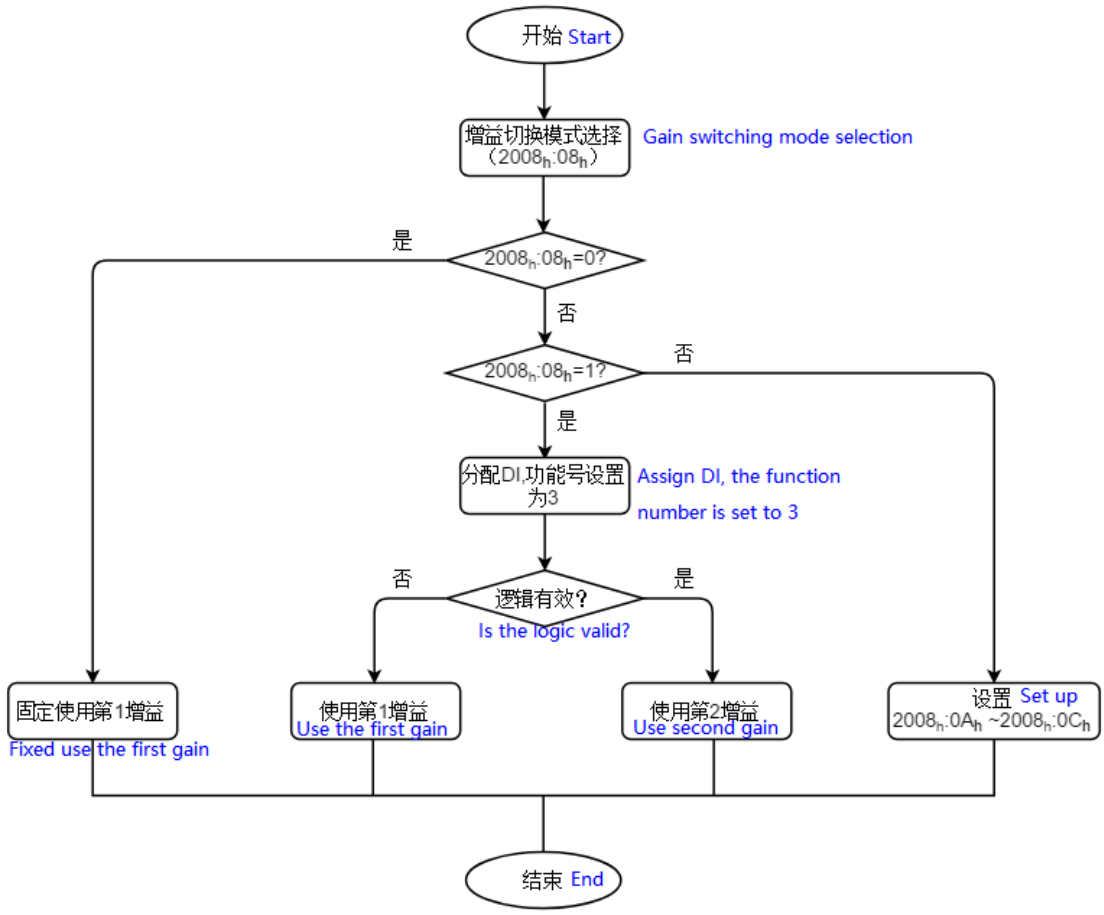


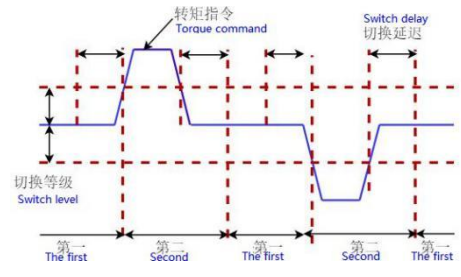
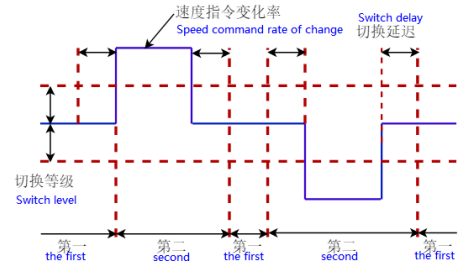
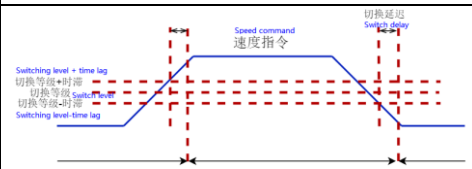
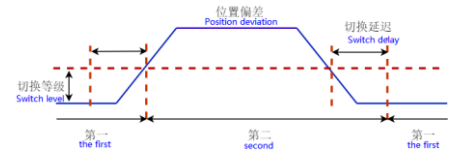
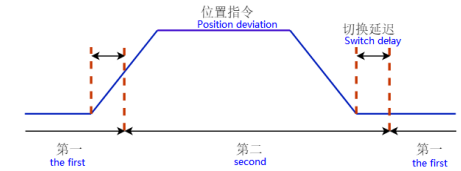
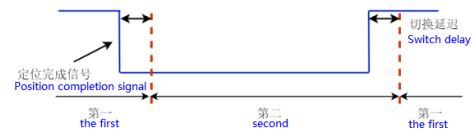
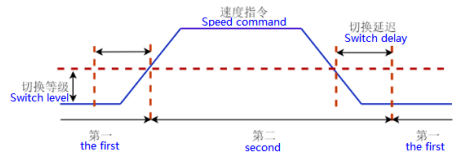
Figure 7-1

Gain switching conditions

There are 10 modes for switching between the first gain 2008_h:01_h(0178_h)~2008_h:03_h($017A_h$)and the second gain 2008_h:04_h($017B_h$)~2008_h:06_h($017D_h$). The schematic diagrams and related parameters of different modes are shown in the following table:

Table 7-2

Gain switching mode			Related parameters		
2008_h:08_h($017F_h$)	Pattern name	Schematic diagram	Delay time 2008_h:$0A_h$ (0181_h)	Switching level2008_h :$0B_h$(0182_h)	Switching time lag2008_h:$0C_h$(0183_h)
0	Fixed the 1st gain	-	Invalid	Invalid	Invalid
1	Use external DI switch	-	Invalid	Invalid	Invalid

Gain switching mode			Related parameters		
2	Torque command		valid (ms)	valid (%)	valid (%)
3	Speed command	The internally effective speed command value ($606B_h$) remains unchanged for 25ms. The first gain is used fixedly, and the second gain is used for changes.	valid (ms)	valid Invalid	valid Invalid
4	Speed command rate of change		valid (ms)	valid (rpm/ms)	valid (rpm/ms)
5	Speed command		valid (ms)	valid (rpm)	valid (rpm)
6	Position deviation		valid (ms)	valid (Encoder unit)	valid (Encoder unit)
7	Position command		valid (ms)	Invalid	Invalid
8	Positioning completed		valid (ms)	Invalid	Invalid
9	Actual speed		valid (ms)	valid (rpm)	valid (rpm)

7.3 Vibration Suppression

7.3.1 Mechanical resonance suppression

In the actual use process of the motor, due to the inherent resonance characteristics of the mechanical system, while the basic gain is increased, resonance may occur near the mechanical resonance frequency, affecting the performance of the motor. The trap can suppress resonance by reducing the gain at a specific frequency.

The integrated motor has 4 sets of manual wave traps. Each set of wave traps has 3 parameters, which are the wave trap frequency, the width and depth of the wave trap. Each parameter is manually set by the user, as shown in the following table:

Table 7-3

Name	Manual trap			
	1	2	3	4
Frequency (Hz)	2009_h：$0D_h$ ($019E_h$)	2009_h：10_h ($01A1_h$)	2009_h：13_h ($01A4_h$)	2009_h：16_h ($01A7_h$)
Notch width (Hz)	2009_h：$0E_h$ ($019F_h$)	2009_h：11_h ($01A2_h$)	2009_h：14_h ($01A5_h$)	2009_h：17_h ($01A8_h$)
Depth(%)	2009_h：$0F_h$ ($01A0_h$)	2009_h：12_h ($01A3_h$)	2009_h：15_h ($01A6_h$)	2009_h：18_h ($01A9_h$)

Note: When the depth level is 0, the trap is invalid.

When using a manual notch filter, you need to set the frequency of the notch filter to the actual center resonance frequency, and enter the width level (central resonance frequency range) and depth level of the group of traps. If the resonance is suppressed, it means that the trap The wave device achieves the effect.

Trap depth

When the notch filter depth level is 100, the input is completely suppressed at the center frequency; when the notch filter depth level is 0, the notch filter is invalid. Therefore, the greater the setting of the notch depth level, the deeper the notch depth, the stronger the suppression of mechanical resonance, but it may cause the system to be unstable, so please pay attention when using it.

The principle of trap suppression is shown in the following figure:

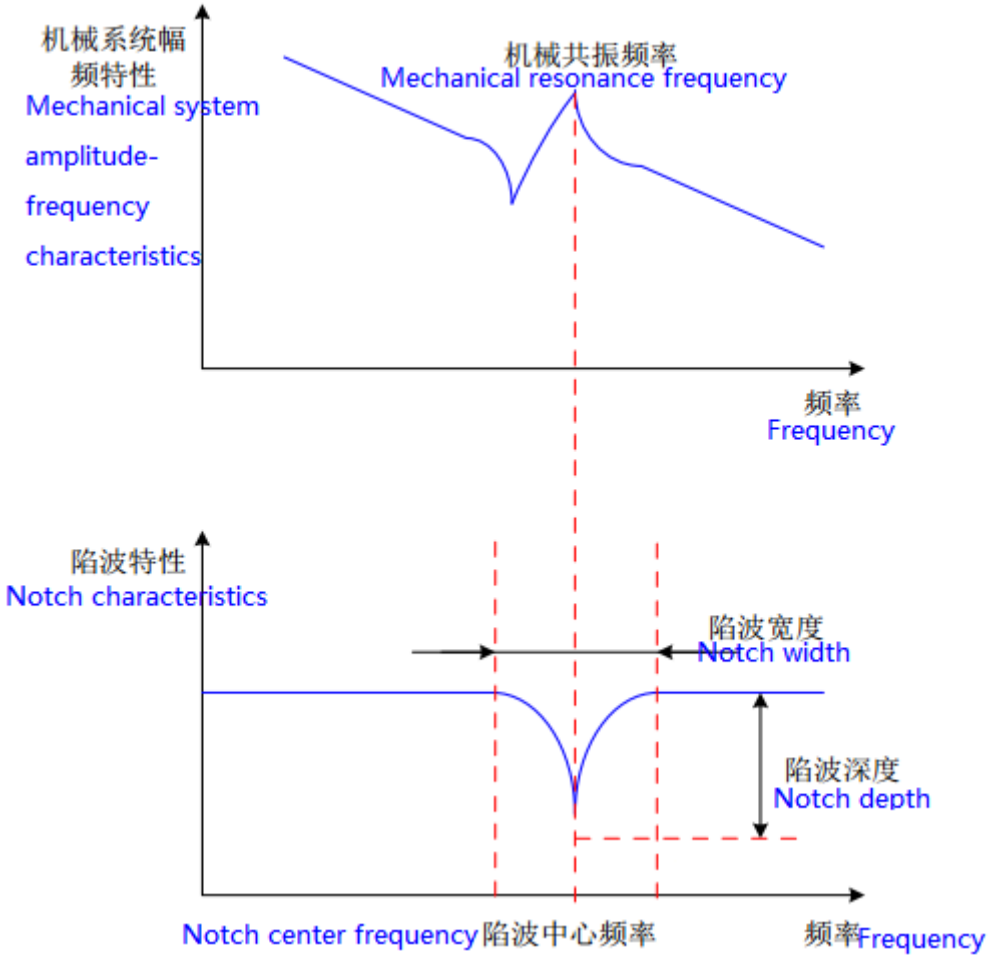


Figure 7-2

7.3.2 Low frequency vibration suppression

Due to the inherent characteristics of the dual inertia system during position control, end vibration is likely to occur during emergency stops, which will affect the positioning effect. The frequency of this vibration is generally within 100Hz. The low-frequency vibration suppression function can reduce the vibration here.

Related objects

Table 7-4

Object Index	Description
2009_h:05_h (0196_h)	Low-frequency resonance suppression mode selection, 0-off, 1-on
2009_h:$1D_h$ ($01AE_h$)	Low frequency vibration frequency (unit: 0.1Hz), set the center frequency of low frequency vibration suppression, the limit range is 10~1000
2009_h:$1E_h$ ($01AF_h$)	Vibration suppression width level (unit: %), set the effective frequency range of vibration suppression, 0 means that only the vibration at the center frequency is suppressed; increasing the setting value can increase the frequency range of low-frequency vibration suppression, but it will lead to

Object Index	Description
	longer positioning time ; The setting value is too small, and the vibration can not be suppressed in the occasion that the load vibration frequency will change (such as belt load), and it can be adjusted and set according to the actual situation. Limit range 0~100
2009_h:23_h ($01B5_h$)	Damping ratio coefficient, set the system vibration damping ratio coefficient, the default value is 5, generally not to be modified, the limit range is 0~50

Set low-frequency vibration suppression parameters

- Turn on the low-frequency vibration suppression mode, set 1 at 2009_h:05_h(0196_h);
- 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;

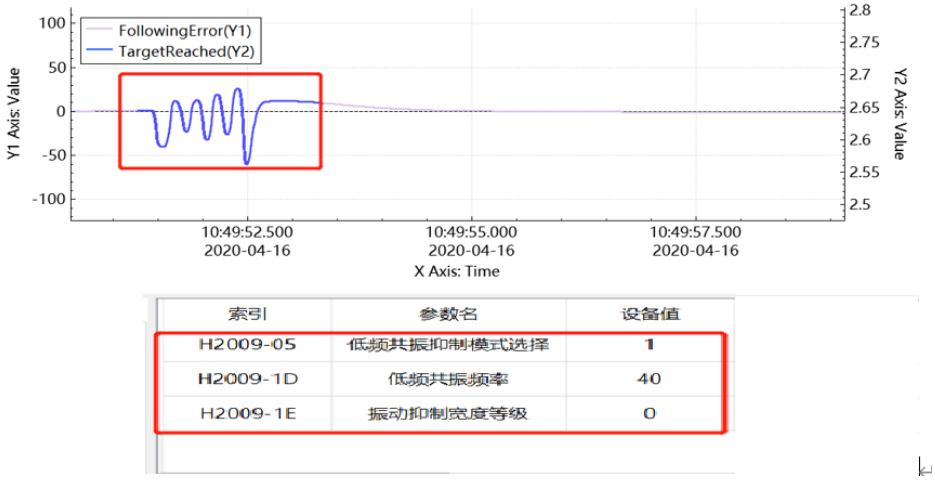
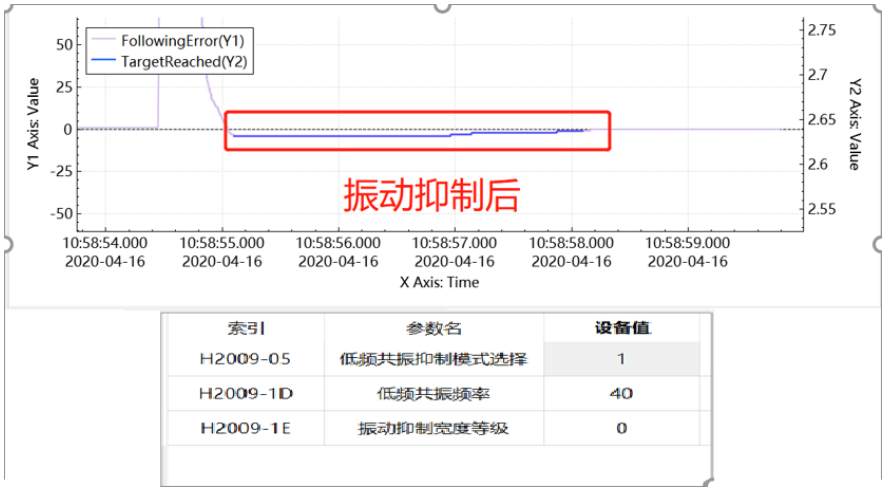


Figure 7-3

- 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 position positioning effect of the motor after using the low-frequency vibration suppression function. If it is within the acceptable range, the low-frequency vibration suppression is effective; if it is not within the reachable range, repeat the above two steps or manually set the low-frequency vibration frequency and vibration suppression width grade.



索引	参数名	设备值
H2009-05	低频共振抑制模式选择	1
H2009-1D	低频共振频率	40
H2009-1E	振动抑制宽度等级	0

Figure 7-4

8 Fault Management

8.1 Alarm Code

When the integrated motor has a warning or a fault, the motor will actively alarm (indicator-yellow light flashes, 0.5s flashing frequency indicates a warning, 0.25s flashing frequency indicates a fault; indicator-green 0.5s flashing indicates communication alarm , Steady light means normal communication), and perform the corresponding action. The corresponding alarm code will be stored in the object 1003_h：$01_h(0002_h)$~1003_h：$10_h(0020_h)$, and the object 1003_h：$00_h(0001_h)$ is the current number of faults.

There are 16 alarm queues in object $1003_h(0002_h$~$0020_h)$, which are stored in a stack and follow the first-in first-out rules. In the case where the historical alarm storage is occupied, a new alarm is generated and the earliest error is deleted, and the previous errors are moved down in turn. The specific alarm codes are shown in the following table:

Table 8-1 List of alarm codes

Alarm Code	Alarm Content	Default Alarm Type	Default Fault Response Code	Whether to Reset	$037F_h$ (Error Code)
0x2300	Motor overcurrent	Fault	0	No	0x2300
0x4012311	Motor overload	Warning	4	Yes	0x2311
0x3002312	Motor blocked	Pause and lock the shaft	3	No	0x2312
0x13210	Power supply overvoltage	Fault	0	Yes	0x3210
0x1013220	Power supply undervoltage	Fault	1	Yes	0x3220
0x1003221	Power supply failure	Fault	1	No	0x3221
0x14210	High temperature alarm	Fault	0	Yes	0x4210
0x14220	Low temperature alarm	Fault	0	Yes	0x4220
0x5080	Drive failure*	Fault	0	No	0x5080
0x5540	Flash operation failure*	Fault	0	No	0x5540
0x5541	Flash initialization failure*	Fault	0	No	0x5541
0x4005542	Flash check error warning*	Warning	4	No	0x5542
0x5543	No parameters in Flash user area*	Fault	0	No	0x5543
0x6000	Hardware initialization failure*	Fault	0	No	0x6000
0x16320	Parameter setting error	Fault	0	Yes	0x6320
0x6321	Registration failure*	Fault	0	No	0x6321
0x7301	Encoder battery low voltage fault	Fault	0	No	0x7301
0x4017302	Encoder battery undervoltage alarm	Warning	4	Yes	0x7302
0x1017304	Encoder multi-turn count error	Fault	1	Yes	0x7304
0x7305	Z pulse failure*	Fault	0	No	0x7305
0x7308	Encoder failure*	Fault	0	No	0x7308
0x4017307	Encoder warning*	Warning	4	Yes	0x7307

Alarm Code	Alarm Content	Default Alarm Type	Default Fault Response Code	Whether to Reset	037F$_h$ (Error Code)
0x7309	Encoder battery failure fault	Fault	1	No	0x7309
0x17310	Overspeed	Fault	0	Yes	0x7310
0x4017501	Illegal function code in Modbus communication	Warning	4	Yes	0x7501
0x4017502	Illegal address in Modbus communication	Warning	4	Yes	0x7502
0x4017503	Illegal data value in Modbus communication	Warning	4	Yes	0x7503
0x4017505	Confirmation in Modbus communication	Warning	4	Yes	0x7505
0x4017506	Slave device busy in Modbus communication	Warning	4	Yes	0x7506
0x401750C	Synchronous message request data data in Modbus communication is greater than total mapped data	Warning	4	Yes	0x750C
0x401750D	The number of synchronous message request data in Modbus communication corresponds to the unequal mapping data	Warning	4	Yes	0x750D
0x401750E	The node address of the unicast message is incorrect under the synchronous function in Modbus communication	Warning	4	Yes	0x750E
0x8610	Origin return timeout	Fault	0	No	0x8610
0x8611	Location out of tolerance	Fault	0	No	0x8611
0x3018613	Software limit error	Pause and lock the shaft	3	No	0x8613
0x3018614	Limit switch error	Pause and lock the shaft	3	No	0x8614
0x4018615	Curve planning calculation error	Warning	4	Yes	0x8615
0x18616	Target position overflow	Fault	0	No	0x8616
0x5018617	Curve planning parameters are too small	Ignore	5	Yes	0x8617
0xFF01	Motor parameter identification failure*	Fault	0	No	0xFF01
0x0401FF02	Parameter saveing fault	Warning	4	Yes	0xFF02
0x04017308	No calibration data in flash	Warning	4	Yes	0x7308

Definition of corresponding bit of alarm code

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

- Definition of fault response code

  0-state machine switches to the fault state (refer to "4.1 CiA402 state machine"). If $605E_h(03C1_h)$>=0, the fault shutdown mode selected by$605E_h(03C1_h)$ will be followed; otherwise: the power tube is fully closed and free shutdown.

  1- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if $605E_h(03C1_h)$>=0, the fault shutdown mode selected by $605E_h(03C1_h)$ is followed; otherwise: the output is shut down according to the slow curve.

  2- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if $605E_h(03C1_h)$>=0, the fault shutdown mode selected by $605E_h(03C1_h)$ is followed; otherwise: the output is shut down according to the fast curve stop.

  3- Press the quick curve to suspend the axis lock without switching to the fault state and not affected by the object 605Eh.

  4- Warning and do not change the current operating state, do not switch to the fault state and are not affected by the object $605E_h(03C1_h)$.

  5- Ignore and record in $1003_h(0002_h$~$0020_h)$, do not switch to the fault state and not be affected by the object $605E_h(03C1_h)$.

  6- Ignore but do not record in $1003_h(0002_h$~$0020_h)$, do not switch to the fault state and are not affected by object $605E_h(03C1_h)$.

- Self-reset allowed

  0-not allowed

  1- Allowed (automatically reset when the warning or fault is eliminated)

Note:

- The above table (alarm code list) The default alarm type in the alarm code is "fault", which means that after this alarm is generated, the motor enters the fault state (status word value is 0x228).
- The default alarm type, default fault response code and whether self-reset of the alarm code can be configured through the object $200E_h(025B_h$~$027B_h)$.
- Alarm content with * symbol cannot be configured through the object $200E_h(025B_h$~$027B_h)$. After the driver and hardware initialization failure, the integrated motor needs to be restarted to operate normally.
- $603F_h$ $(037F_h)$contains the last error of the integrated motor, corresponding to the lower 16 fault codes in the alarm code.

8.2 Fault Action Settings

The alarm code consists of a total of 32 bits including a fault code (16 bits), a fault response code (8 bits), and a self-reset permission (8 bits). The fault code indicates the fault phenomenon of the motor in an abnormal state. The fault code is used to determine the current state of the motor. It generally corresponds to the object $603F_h(037F_h)$ (error code). 0 indicates that the motor is currently fault-free; the fault response code indicates the corresponding action after the motor fault. Method and treatment of the status word fault bit, such as free stop-fault state, stop at a

certain deceleration-fault state, pause shaft lock-non-fault state, etc.; self-reset permission indicates whether the motor is allowed to automatically reset after the fault of the motor is eliminated The fault bit of the current status word, 1-allow, 0-disallow. 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
Fault response code[8]								Self reset allowed[8]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Error code[16]															

Specific definition of fault response code

0-state machine switches to the fault state (refer to "4.1 CiA402 state machine"). If $605E_h(03C1_h)$>=0, the fault shutdown mode selected by $605E_h(03C1_h)$ will be followed; otherwise: the power tube is fully closed and free shutdown.

1- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if $605E_h(03C1_h)$>=0, the fault shutdown mode selected by $605E_h(03C1_h)$ is followed; otherwise: the output is shut down according to the slow curve.

2- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if $605E_h(03C1_h)$>=0, the fault shutdown mode selected by $605E_h(03C1_h)$ is followed; otherwise: the output is shut down according to the fast curve stop.

3- Press the quick curve to suspend the axis lock without switching to the fault state and not affected by the object 605Eh.

4- Warning and do not change the current operating state, do not switch to the fault state and are not affected by the object $605E_h(03C1_h)$.

5- Ignore and record in $1003h(0002_h\sim0020_h)$, do not switch to the fault state and not be affected by the object $605E_h(03C1_h)$.

6- Ignore but do not record in $1003_h(0002_h\sim0020_h)$, do not switch to the fault state and are not affected by object $605E_h(03C1_h)$.

Alarm code configuration

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

Table 8-1

Fault Code Group $200E_h$	Meaning	Default Fault action
0x4012311	Motor overload	No
0x3002312	Motor blocked	No
0x13210	Power supply overvoltage	Yes
0x1013220	Power supply undervoltage	Yes
0x14210	High temperature alarm	Yes
0x14220	Low temperature alarm	Yes
0x16320	Parameter setting error	Yes
0x17310	Overspeed	Yes
0x17501	Illegal function code in Modbus communication	No
0x17502	Illegal address in Modbus communication	No
0x17503	Illegal data value in Modbus communication	No

Fault Code Group $200E_h$	Meaning	Default Fault action
0x17505	Confirmation in Modbus communication	No
0x17506	Slave device busy in Modbus communication	No
0x1750C	Synchronous message request data data in Modbus communication is greater than total mapped data	No
0x1750D	The number of synchronous message request data in Modbus communication corresponds to the unequal mapping data	No
0x1750E	The node address of the unicast message is incorrect under the synchronous function in Modbus communication	No
0x8610	Origin return timeout	Yes
0x8611	Location out of tolerance	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: The "Yes (No)" of the fault action in the above list indicates whether the motor has entered a fault state after this fault occurs.

Example of alarm code configuration

Power supply undervoltage setting

$200E_h$:04_h (0261_h) (power supply undervoltage) is set to 0x1003220. When the motor is running, the power supply voltage is lower than the detection threshold (default 24V), the motor stops according to the shutdown mode set by $605E_h$($03C1_h$) (refer to the order of corresponding bits, the fault response code corresponds to 1); When the power supply voltage returns to normal, the manual reset operation is required for the motor to perform normal operation (refer to the order of the corresponding bits, self-reset is allowed to correspond to 0).

8.3 Fault Reset

After the integrated fault is restored, the fault codes that are not allowed by self-reset need to be manually reset, so that 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).
- One of the functions is configured as No. 2 function code (alarm reset) through the physical input terminal 2003_h(00D5h ~$00E6_h$), and the corresponding logic selection is set, and it can be reset according to the logic selection of the setting terminal to make it effective.
- Through virtual input terminal 2017_h(0326_h ~0346_h), one of the functions is configured as No. 2 function code (alarm reset), and set the corresponding logic selection, according to the logic selection of the setting terminal to make it effective to reset.

8.4 Fault Detection Instructions

- Drive failure detection

This fault is triggered by the interruption response of the driver chip fault signal, and the driver will be reported as triggered.

- I^2T overload detection
  When the motor runs above the rated current, the fault detection will calculate the time integral of I2. When the integral value is greater than the product of the set maximum current $2000_{h:}$ $0A_h$ (0066_h)and the maximum current duration 2000_h: $0B_h$(0067_h), an overload warning will be reported. , And limit the torque output to the rated value, and lift the limit after 30s.
- Locked rotor fault detection
  When the motor is running below 5 rpm and above the rated current, the fault detection will calculate the time integral of I2. When the integral value is greater than the set maximum current 2000_h: $0A_h$(0066_h) and maximum current duration 2000_h: $0B_h$ (0067_h)When the product of is, a locked fault is reported and the output is stopped.
- High temperature and low temperature fault detection
  When the temperature of the driver is higher than 105℃, a high temperature fault is reported, and when the temperature is lower than 70℃, the fault is removed; when the temperature of the driver is lower than -25℃, the fault is reported to be too low, and when the temperature is higher than -20℃, the fault is removed.
- Overvoltage and undervoltage fault detection
  When the power supply voltage of the motor is higher than the voltage set in 2001_h: $0E_h$ (0092_h), an overvoltage fault is reported; when the power supply voltage of the motor is lower than the voltage set in 2001_h: 10_h(0094_h), an undervoltage fault is reported.
- Parameter setting fault detection
  Parameter setting fault detection The setting of multi-function terminals is wrong, including different terminals (physical terminals and virtual terminals) are set with the same function number or CiA402 zero return mode does not set 14, 15 and 31 function numbers.
- Overspeed fault detection
  When the speed of the motor is higher than the set speed of $200A_h$: 06_h ($01BC_h$)and maintained for 10ms, an overspeed fault is reported, and if it is lower, the fault status is released.
- Illegal address fault detection in Modbus communication
  This error is identified by the slave protocol error flag under Modbus communication.
- Malfunction detection of illegal data values in Modbus communication
  This error is identified by the slave protocol error flag under Modbus communication.
- Malfunction detection of illegal function codes in Modbus communication
  This error is identified by the slave protocol error flag under Modbus communication.
- The fault detection of the unicast message node address error under the synchronization function in Modbus communication
  This error is identified by the slave protocol error flag under Modbus communication.
- Slave device busy fault detection in Modbus communication
  This error is identified by the slave protocol error flag under Modbus communication.
- In the Modbus communication, the synchronization request data data is greater than the total data mapping fault detection
  This error is identified by the slave protocol error flag under Modbus communication.
- In the Modbus communication, the synchronization request data data is greater than the total data mapping

fault detection

This error is identified by the slave protocol error flag under Modbus communication.

- The number of synchronous message request data in Modbus communication corresponds to the unequal mapping data fault detection

  This error is identified by the slave protocol error flag under Modbus communication.

- Hardware initialization failure detection

  When the motor is powered on, if the hardware (driver, encoder) is not initialized successfully, it will report a hardware initialization failure.

- Flash fault detection

  When operating the Flash of the motor, when the saved parameters or the power-on reading storage parameters fail to pass the test, a Flash fault is reported.

- Origin return overtime fault detection

  When the motor is enabled in the CiA402 homing mode, if there is no homing operation within the limited time (2005h: 1Ch), a homing homing fault will be reported.

- Tracking fault detection

  When the motor is under position mode control, when the position deviation exceeds ±6065h, and the time reaches 6066h (ms), a tracking failure is reported.

- Software overrun fault detection

  In the position control mode, the motor is judged according to the software limit value (user unit) set by 607Bh and 607Dh. If the actual position exceeds the set value, it will report a software overrun fault.

- Over-limit switch fault detection

  According to the 14 and 15 function numbers set by the multi-function terminal, it is detected whether the positive and negative limit switches are triggered, and if the trigger is triggered, the fault of the limit switch is reported.

- Overflow fault detection of target position

  When the motor is in position control mode, if the given position ($607A_h$) exceeds the data range of the data type int32 (-2147483583~+2147483583) after the gear ratio (6091_h), it will report the target position overflow fault.

- Curve planning parameter too small fault detection

  In the position control mode of the motor, if the given impact or acceleration cannot achieve the S-shaped curve within the given displacement interval, the curve planning parameter will be reported to be too small.

- Flash initialization failure (0x5541)

  The controller does not save the parameters in the Nimotion area.

- No calibration data in flash (0x7308)

  When 2000_h:20_h($007E_h$)!=0, and no date in flash area, the error will be happen. After the calibration of coder, the fault may be cleared automatically.

- Command that cannot be executed

  When the motor is in the enabled state or when the supply voltage is lower than 80% of the rated voltage 2000h: 07h (0063h), the motor reports a fault that it cannot execute commands when saving and restoring parameters.

- Encoder battery low voltage fault

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

- Encoder battery undervoltage alarm

  When the servo motor is powered on, the encoder battery voltage is detected to be between 2.4V and 2.6V, and a fault of 0x4017302 is reported.
- Encoder communication failure

  The servo motor cannot communicate with the encoder and reports a fault of 0x7303.
- 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.

9 NiMotion Agreement

The NiMotion protocol optimizes the Modbus protocol and creates two new message formats (process data message and synchronization message) based on Modbus. Through these two messages, multi-command control can be achieved, which greatly reduces the data processing time, improves the coordination of multi-motor control, and further enables the motor to achieve the function of two-to-one synchronization. Process data messages are used to freely combine multiple parameters into a frame of messages and send and receive at the same time. Synchronous messages are used to control when the command takes effect.

R_PDO: Receive process data;

T_PDO: send process data.

9.1 Timing chart of master and slave

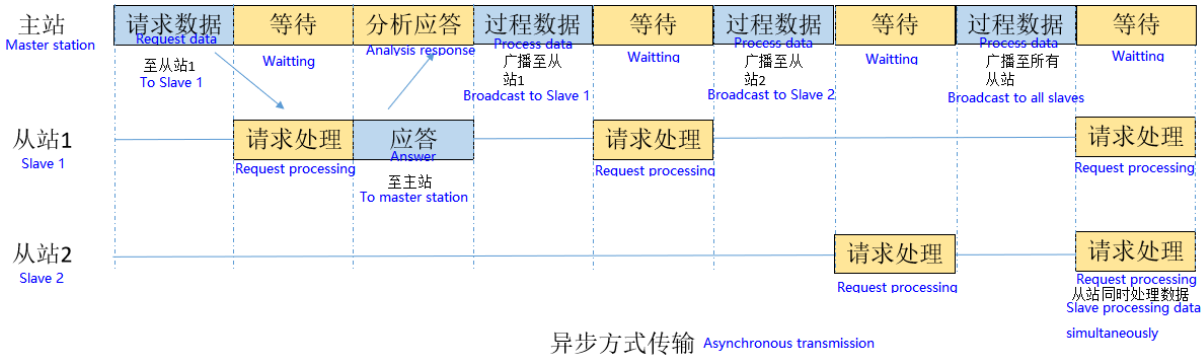


Figure 9-1

In the asynchronous transmission process, the unicast process data for the slave station 1 is sent to the master station after processing by the slave station 1, and the master station sends the broadcast process data for the slave station 1 after receiving the error-free response. Slave 1 processing, but no response; master station sends broadcast process data for slave station 2, only slave station 2 processes, but no response; master station sends broadcast process data for all slave stations, all slave stations process , But no response.

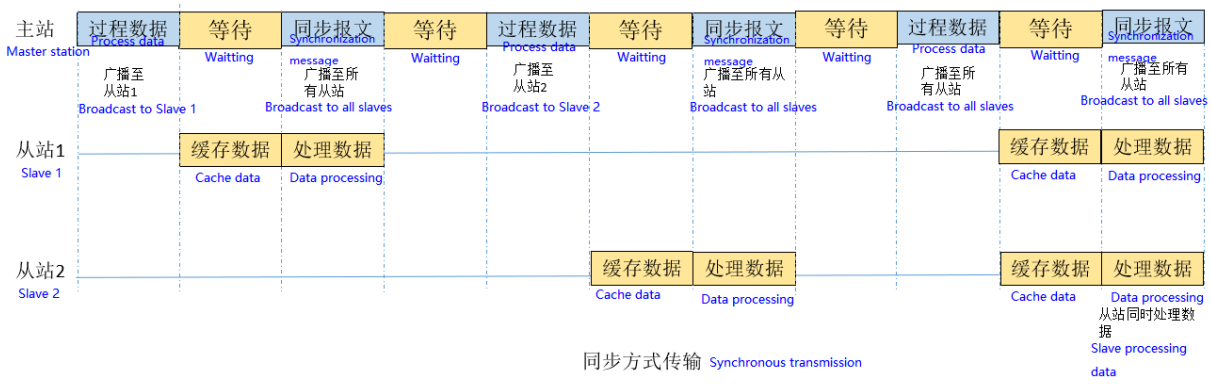


Figure 9-2

In the synchronous transmission mode, the master station sends the broadcast process data for the slave station 1. After the slave station 1 receives the message, it only buffers the data and does not process it. The master station broadcasts the synchronization message again, and the slave station 1 starts processing the data; The master station

sends the broadcast process data for the slave station 2. After the slave station 2 receives the message, it only buffers the data and does not process it. The master station broadcasts the synchronization message again, and the slave station 2 starts processing the data; the master station sends the data for all slave stations. The broadcast process data of slave station 1 and slave station 2 are both buffered data and not processed after receiving the message. The master station broadcasts and sends the synchronization message again, and slave station 1 and slave station 2 start processing data at the same time.

9.2 Message Format

9.2.1 Synchronization message

Synchronization message is a broadcast message that is used to determine the process data update time to register time in synchronization mode by redefining the meaning of the register address.

The function of the synchronization message is to update the data stored in the buffer area to the register in the synchronization mode, thereby controlling the change of the slave station.

The detailed description of the synchronous broadcast message is as follows:

Table 9-1 Master station synchronization request message

Serial number	Field Name	Byte length (bytes)	Content	Remarks
1	Address field	1	0x00	Broadcast mode
2	function code	1	0x10	Write multiple registers
3	Register address high byte	1	0x80	0x8000 Synchronization message identifier 0x8000
4	Register address low byte	1	0x00	
5	Number of registers high byte	1	0x00	1 register
6	Register low byte	1	0x01	
7	Bytes	1	0x02	Data part 2 bytes
8	Register value high byte	1	0x66	Sync message data value flag Fixed at 0x6688
9	Register value low byte	1	0x88	
10	CRC16 low order	1	Calculate CRC low byte	16-bit CRC check
11	CRC16 high order	1	Calculate CRC high byte	

The synchronous message is to write data to the slave station, considering that the process data message is to write multiple registers, so in order to reduce the changes of the existing program, the 0x10 function code is selected uniformly.

In order to formulate a specific synchronization message, the register address is changed to the identifier of the synchronization message. Considering that it needs to be distinguished from the existing register address, the highest bit is set to 1, which is 0x8000.

The register value serves as a synchronization flag, so the fixed data is 0x6688.

9.2.2 Process Data Write Operation

9.2.2.1 Request message

Support modbus protocol message format

Read the content and length, set according to the register (0x6000~0x6021).

Table 9-2 Process data request

Serial number	Field Name	Byte length (bytes)	Content	Remarks
1	Address field	1	0x00	Broadcast/unicast mode
2	Function code	1	0x10	Write multiple registers
3	Register address high	1	0x83	Message pattern
4	Register address low	1	0x00	
5	Number of registers high byte	1	0x00	Number of registers （1~33）
6	Number of registers low byte	1	0x05	
7	Number of bytes	1	0x0A	Number of registers*2 (2~66)
8	Register value high byte	1	0x00	Motor address (0~247) 0 is sent to all motors
9	Register value low byte	1	0x01	
10	Register value high byte	1	0x00	Register value (Set according to actual needs)
11	Register value low byte	1	0x07	
12	Register value high byte	1	0x00	
13	Register value low byte	1	0x01	
14	Register value high byte	1	0x00	
15	Register value low byte	1	0x00	
16	Register value high byte	1	0x03	
17	Register value high byte	1	0x84	
18	CRC16 low order	1	Calculate CRC low byte	16-bit CRC check
19	CRC16 high order	1	Calculate CRC high byte	

Process data is data from writing multiple registers to the slave station, so the 0x10 function code is used.

In order to formulate a specific process data message, the register address is changed to the identification of the process data message, considering that it needs to be distinguished from the existing register address, so the highest bit is set to 1, refer to the CAN custom process data format, process The data identifier is +0x300, which is 0x8300.

There are a maximum of 16 maps, each with a maximum of two registers, so the map has a maximum of 32 registers, plus a motor address to occupy a register, so the number of registers ranges from 1 to 33.

Each register occupies two bytes, so the number of bytes ranges from 2 to 66.

The node address occupies a register, and there can be at most 247 slave stations on the modbus bus. So the range is 1~247. If it is 0, it means to send to all slaves.

9.2.2.2 Reply Message

Please refer to modbus protocol 0x10 function code response message.

9.2.3 Process Data Read Operation

9.2.3.1 Request Message

The process data read operation can only be used in asynchronous mode and supports the modbus protocol message format.

Read the content and length, set according to the register (0x6000~0x6021).

Table 9-3 Process data read request message

Serial Number	Field Name	Byte length (bytes)	Content	Remarks
1	Address field	1	0x01	Node address
2	Function code	1	0x04	Read multiple registers
3	Register address high	1	0x83	Register address (buffer address)
4	Register address low	1	0x80	
5	Number of registers high byte	1	0x00	Number of registers (1~32)
6	Number of registers low byte	1	0x02	
13	CRC16 low order	1	Calculate CRC low byte	16-bit CRC check
14	CRC16 high order	1	Calculate CRC high byte	

Function code 0X10 is to write register, here is to read the value of register, so 0x10 function code is no longer suitable, choose 0x04 function code.

Process data reads the data in the cache area, so the address is also the address of the cache area.

The cache area has a maximum of 16 mapping entries, and each entry has a maximum of two registers, so the number of registers ranges from 1 to 32.

9.2.3.2 Reply Message

Please refer to modbus protocol 0x04 function code response message.

9.3 Exception Handling

Please refer to Chapter 8 Fault Management.

9.4 Examples

1. Set the number of mapping entries register (0x5000) in the buffer area of the slave station, and determine the number of entries according to requirements.
2. According to the number of entries, set from the R_PDO map 1 register, the maximum can be set to the map 12 registers, the register high byte determines the starting address of the mapped register, and the register low byte determines the number of mapped registers.
3. Set the number of entries mapping register (0x6000) in the buffer area of the slave station, and determine the number of entries according to the requirements.
4. According to the number of entries, set from the T_PDO map 1 register, the maximum can be set to the

map 12 registers, the register high byte determines the starting address of the mapped register, and the register low byte determines the number of mapped registers.

5. Set the synchronization mode to enable 200Ch: 18h (corresponding to Modbus holding register address 0x0246) as enable (0x01).
6. The master station sends the process data message with register address 0x8300, and the slave station executes the function according to the function code.
7. The master station sends a synchronization message with a register address of 0x8000 and a data value of 0x6688, and the slave station executes a function code function. The data update is triggered and the motor status changes accordingly.

Implementation process:

1. The number of RPDO mapping entries for motor 1 is two.

Table 9-1

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
01	0x10	50 00	00 02	04	00 00 00 02	Skip

2. The RPDO mapping control word of motor No. 1 is 6040h: 0h (corresponding to Modbus holding register address 0x0380), target position 607Ah: 0h (corresponding to Modbus holding register address 0x03E7).

Table 9-2

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
01	0x10	50 02	00 04	08	03 80 00 01 03 E7 00 02	Skip

3. Motor No. 1 starts synchronous mode 200Ch: 18h (corresponding to Modbus holding register address 0x0246).

Table 9-3

Slave address	Function code	Register address	Register value	CRC check value
01	0x06	02 46	00 01	Skip

4. After saving the parameters, the No. 1 motor restarts after power off.

Table 9-4

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
01	0x10	00 26	00 02	04	65 76 61 73	Skip

5. The master station broadcasts the process data (the control word is 6, the target position is 1000) to the No. 1 motor.

Table 9-5

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
00	0x10	83 00	00 04	08	00 01 00 06 00 00 03 E8	Skip

6. The master station broadcasts the synchronization message update parameters.

Table 9-6

Slave address	Function code	Register address	Number of registers	Bytes	Register value	CRC check value
00	0x10	80 00	00 01	02	66 88	Skip

9.5 Mapping Parameter Settings

This mapping is done for multi-instruction control of process data. Each mapping occupies two registers. The high register is used to store the first address of the register of the parameter that the user wants to map, and the low register is used to store the register occupied by the parameter in the parameter table. Quantity. Addresses 0x5000 and 0x6000 represent the number of mapping entries enabled by R_PDO and T_PDO, respectively. The maximum number of mapping entries is 12.

Example: The value of R_PDO mapping 1 is 0x02310001

The upper 16 bits 0x0231 refer to the parameter mapped to register address 0x0231: communication baud rate.

The lower 16 bits of 0x0001 indicate that the parameters of this map occupy a register.

Table 9-7

Modbus Address	Name	Description	Number of Registers	Data Range
5000	R_PDO map entries	Number of entries in the process data message mapping register	2	0~0x0C
5002	R_PDO mapping parameter 1	High register: mapped register start address, Low register: number of registers	2	0~4294967295
5004	R_PDO mapping parameter 2	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5006	R_PDO mapping parameter 3	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5008	R_PDO mapping parameter 4	High register: mapped register start address , Low register: number of registers	2	0~4294967295
500A	R_PDO mapping parameter 5	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5006C	R_PDO mapping parameter 6	High register: mapped register start address, Low register: number of registers	2	0~4294967295
500E	R_PDO mapping parameter 7	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5010	R_PDO mapping parameter 8	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5012	R_PDO mapping parameter 9	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5014	R_PDO mapping parameter 10	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5016	R_PDO mapping parameter 11	High register: mapped register start address , Low register: number of registers	2	0~4294967295
5018	R_PDO mapping parameter 12	High register: mapped register start address, Low register: number of registers	2	0~4294967295

Modbus Address	Name	Description	Number of Registers	Data Range
6000	Number of T_PDO mapping entries	Number of entries in the process data message mapping registers	2	0~0x0C
6002	T_PDO mapping parameter 1	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6004	T_PDO mapping parameter 2	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6006	T_PDO mapping parameter 3	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6008	T_PDO mapping parameter 4	High register: mapped register start address , Low register: number of registers	2	0~4294967295
600A	T_PDO mapping parameter 5	High register: mapped register start address , Low register: number of registers	2	0~4294967295
600C	T_PDO mapping parameter 6	High register: mapped register start address , Low register: number of registers	2	0~4294967295
600E	T_PDO mapping parameter 7	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6010	T_PDO mapping parameter 8	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6012	T_PDO mapping parameter 9	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6014	T_PDO mapping parameter 10	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6016	T_PDO mapping parameter 11	High register: mapped register start address , Low register: number of registers	2	0~4294967295
6018	T_PDO mapping parameter 12	High register: mapped register start address , Low register: number of registers	2	0~4294967295

10 Object Dictionary

The object dictionary is an important part of the device specification. It is an ordered set of parameters and variables that contains all the parameters of the device description and the network status of the device. Each object contains an index, sub-index, name, description, data type, data range, accessibility, ability to map, factory settings, units, and effective mode.

Glossary

"Index": Specify the position of the same type of object in the object dictionary, expressed in hexadecimal
"Sub-index": Under the same index, contains multiple objects, and the offset of each object under this category
"Data Type and Range": Please refer to the table below for details

Table 10-1

Type of data	Data range	Data length
Int8	-128~127	1 byte
Int16	-32768~32767	2 bytes
Int32	-2147483583~2147483583	4 bytes
Uint8	0~255	1 byte
Uint16	0~65535	2 bytes
Uint32	0~4294967295	4 bytes
String	ASCII	~

"Accessibility": see the table below for details

Table 10-2

Accessibility	Explanation
RW	Can read and write
WO	Write only
RO	Read only

"Can be mapped": see the table below for details

Table 10-2

Can it be mapped	Explanation
NO	Not mapped in PDO
RPDO	Can be used as RPDO
TPDO	Can be used as TPDO

"Effective Mode": Please refer to the table below for details

Table 10-3

Effective method	Explanation
Effective immediately	After the parameter setting is completed, the set value takes effect immediately
Effective downtime	After the parameter setting is completed, wait until the drive is not in the running state, the set value takes effect
Power on again	After the parameter setting is completed, turn on the drive power again, the set value takes effect

10.1 Communication Parameter 1000h Group

The 1000h object group contains the parameters required for Modbus communication.

Table 10-4

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
1001_h	-	00_h	Error register	Uint8	-	RO	NO	0	VAR	-
1003_h	00_h	01_h	Number of alarms present in the device	uint8	-	RO	NO	-	VAR	-
	01_h	02_h	History alarm cache	uint32	-	RO	NO	-	VAR	-
	02_h	04_h		uint32	-	RO	NO	-	VAR	-
	03_h	06_h		uint32	-	RO	NO	-	VAR	-
	04_h	08_h		uint32	-	RO	NO	-	VAR	-
	05_h	$0A_h$		uint32	-	RO	NO	-	VAR	-
	06_h	$0C_h$		uint32	-	RO	NO	-	VAR	-
	07_h	$0E_h$		uint32	-	RO	NO	-	VAR	-
	08_h	10_h		uint32	-	RO	NO	-	VAR	-
	09_h	12_h		uint32	-	RO	NO	-	VAR	-
	$0A_h$	14_h		uint32	-	RO	NO	-	VAR	-
	$0B_h$	16_h		uint32	-	RO	NO	-	VAR	-
	$0C_h$	18_h		uint32	-	RO	NO	-	VAR	-
	$0D_h$	$1A_h$		uint32	-	RO	NO	-	VAR	-
	$0E_h$	$1C_h$		uint32	-	RO	NO	-	VAR	-
	$0F_h$	$1E_h$		uint32	-	RO	NO	-	VAR	-
	10_h	20_h		uint32	-	RO	NO	-	VAR	-
1010_h	01_h	26_h	Save all current parameters	uint32	-	RW	NO	1	VAR	-

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
1011_h	01_h	$3C_h$	Read all parameters	uint32	-	RW	NO	1	VAR -	-
1017_h	-	52_h	Producer heartbeat time	Uint16	-	RW	NO	0	ms	-
1018_h	03_h	53_h	Software version number	uint32	-	RO	NO	-	-	-
	04_h	55_h	Product Serial Number	uint32	-	RO	NO	-	-	-

10.2 Description of manufacturer-defined parameters 2000h group

Table 10-5

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2000_h (Motor parameters)	06_h	62_h	Bus encoder type	uint16	0~1	RW	NO	1	-	Shutdown effective
	07_h	63_h	Rated voltage	uint16	24~48	RW	NO	48	1V	Shutdown effective
	08_h	64_h	Rated power	uint16	-	RW	NO	40	0.01KW	Shutdown effective
	09_h	65_h	Rated current	uint16	-	RW	NO	1250	0.01A	Shutdown effective
	$0A_h$	66_h	Maximum current	uint16	-	RW	NO	2250	0.01A	Shutdown effective
	$0B_h$	67_h	Maximum current duration	uint16	-	RW	NO	3000	0.1s	Shutdown effective
	$0C_h$	68_h	Rated torque	uint16	-	RW	NO	127	0.01Nm	Shutdown effective
	$0D_h$	69_h	Max Torque	uint16	-	RW	NO	2000	0.01Nm	Shutdown effective
	$0E_h$	$6A_h$	Rated speed	uint16	-	RW	NO	3000	1rpm	Shutdown effective
	$0F_h$	$6B_h$	Maximum speed	uint16	-	RW	NO	6000	1rpm	Shutdown effective
	10_h	$6C_h$	Moment of inertia Jm	uint16	-	RW	NO	30	0.01kgcm^2	Shutdown effective
	11_h	$6D_h$	Number of pole pairs of permanent magnet synchronous motor	uint16	-	RW	NO	4	1	Shutdown effective
	12_h	$6E_h$	Stator resistance	uint16	-	RW	NO	115	0.001Ω	Shutdown effective
	13_h	$6F_h$	Stator inductance Lq	uint16	-	RW	NO	25	0.01m	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2000_h (Motor parameters)									H	
	14_h	70_h	Stator inductance Ld	uint16	-	RW	NO	25	0.01mH	Shutdown effective
	16_h	72_h	Torque coefficient Kt	uint16	-	RW	NO	11	0.01Nm/Arms	Shutdown effective
	19_h	75_h	Encoder absolute digits	uint32	-	RW	NO	14	1bit	Shutdown effective
	$1A_h$	77_h	Number of encoder poles	uint16	-	RW	NO	1	1	Shutdown effective
	$1C_h$	$7A_h$	Position sensor selection	uint16	-	RW	NO	2	-	Shutdown effective
	$1D_h$	$7B_h$	Z signal corresponds to electrical angle	uint16	-	RW	NO	0	1/4096	Shutdown effective
	$1E_h$	$7C_h$	U-phase rising edge corresponds to the corresponding angle	int16	-	RW	NO	0	1	Shutdown effective
2001_h (Driver parameters)	07_h	$8B_h$	Rated output current	Uint16	0~65535	RO	NO	1200	0.01A	Shutdown effective
	08_h	$8C_h$	Maximum output current	uint16	0~65535	RO	NO	1500	0.01A	Shutdown effective
	$0D_h$	91_h	Switch dead time	uint16	0~65535	RW	NO	100	us	Shutdown effective
	$0E_h$	92_h	DC bus overvoltage protection point	uint16	0~65535	RW	NO	100	1V	Shutdown effective
	$0F_h$	93_h	DC bus voltage relief point	uint16	0~65535	RW	NO	55	1V	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2001_h (Driver parameters)	10_h	94_h	DC bus voltage undervoltage point	uint16	0~65535	RW	NO	24	1V	Shutdown effective
	11_h	95_h	Drive overcurrent protection point	uint16	0~65535	RW	NO	10	1A	Shutdown effective
	18_h	$9C_h$	Current sampling filter parameters	uint16	0~65535	RW	NO	100	0.01	Shutdown effective
	$1C_h$	$A0_h$	Current loop cutoff frequency	uint16	0~65535	RW	NO	800	1Hz	Shutdown effective
	$1D_h$	$A1_h$	Open-loop operating current	uint16	0~65535	RW	NO	625	0.01A	Shutdown effective
2002_h (Basic control parameters)	01_h	$B1_h$	Control mode selection	uint16	0~5	RW	NO	4	-	Shutdown and setting effective immediately
			Specific description: 0~CIA402 mode, 1~NiMotion position mode, 2~NiMotion Velocity mode, 3~NiMotion torque mode, 4~NiMotion open loop mode, 5~off-line identification of motor parameters							
	04_h	$B4_h$	Duty cycle of output pulse	uint16	0~100	RW	NO	50	1%	Operation setting shutdown takes effect
	13_h	$C3_h$	Selection of internal and external	uint16	0~2	RW	NO	0	1	Power on again
	$1D_h$	CD_h	MCU identification code 0	uint32	-	RO	NO	0	-	-
	$1E_h$	CF_h	MCU identification code 1	uint32	-	RO	NO	0	-	-
	$1F_h$	$D1_h$	MCU identification code 2	uint32	-	RO	NO	0	-	-

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2002_h (Basic control parameters)										
2003_h （Physical terminal input parameters）	03_h	$D5_h$	DI1 terminal function selection	uint16	0~55	RW	NO	1	1	Shutdown effective
			Specific description: see chapter 6.1							
	04_h	$D6_h$	DI1 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 6.1							
	05_h	$D7_h$	DI2 terminal function selection	uint16	0~48	RW	NO	2	1	Shutdown effective
	06_h	$D8_h$	DI2 terminal logic selection	uint16	0~4	RW	NO	2	1	Shutdown effective
	07_h	$D9_h$	DI3 terminal function selection	uint16	0~48	RW	NO	12	1	Shutdown effective
	08_h	DA_h	DI3 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	09_h	DB_h	DI4 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	$0A_h$	DC_h	DI4 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	$0B_h$	DD_h	DI5 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	$0C_h$	DE_h	DI5 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2003_h（Physical terminal input parameters）	$0D_h$	DF_h	DI6 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	$0E_h$	$E0_h$	DI6 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	$0F_h$	$E1_h$	DI7 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	10_h	$E2_h$	DI7 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	11_h	$E3_h$	DI8 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	12_h	$E4_h$	DI8 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	13_h	$E5_h$	DI9 terminal function selection	uint16	0~48	RW	NO	0	1	Shutdown effective
	14_h	$E6_h$	DI9 terminal logic selection	uint16	0~4	RW	NO	0	1	Shutdown effective
	15_h	$E7_h$	Effective power distribution	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: bit0-DI1, bit1-DI2, bit2-DI3; the corresponding bit is 1 means the corresponding DI port is high level 3.0V							
	16_h	$E8_h$	Effective power distribution	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: bit0-DI1, bit1-DI2, bit2-DI3; the corresponding bit is 1 means the corresponding DI port is high level 3.0V							
2003_h（Physical	17_h	$E9_h$	AI1Offset	int16	-32768~32767	RW	NO	0	1mV	
	18_h	EA_h	AI1FilterTimeConst	int16	0~65535	RW	NO	0	-	
	19_h	EB_h	AI1DeadZone	uint16	0~65535	RW	NO	0	1mV	

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
terminal input parameters）	1A h	EC h	AI1ZeroDrift	int16	-32768~32767	RW	NO	0	0.001	
	1B h	ED h	AI2Offset	int16	-32768~32767	RW	NO	0	1mV	
	1C h	EE h	AI2FilterTimeConst	int16	0~65535	RW	NO	0	-	
	1D h	EF h	AI2DeadZone	uint16	0~65535	RW	NO	0	1mV	
	1E h	F0 h	AI2ZeroDrift	int16	-32768~32767	RW	NO	0	0.001	
	1F h	F1 h	SpeedValAt10V	uint16	0~4000	RW	NO	0	rpm	
	20 h	F2 h	TorqueValAt10V	uint16	0~1000	RW	NO	0	0.1%	
2004 h (Terminal output parameter)	01 h	F8 h	DO1 terminal function selection	uint16	0~30	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 6.1							
	02 h	F9 h	DO1 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 6.1							
	03 h	FA h	DO2 terminal function selection	uint16	0~30	RW	NO	0	1	Shutdown effective
	04 h	FB h	DO2 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	05 h	FC h	DO3 terminal function selection	uint16	0~30	RW	NO	0	1	Shutdown effective
	06 h	FD h	DO3 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	07 h	FE h	DO4 terminal function selection	uint16	0~30	RW	NO	0	1	Shutdown effective
	08 h	FF h	DO4 terminal logic selection	uint16	0~1	RW	NO	0	1	hutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
	09h	100h	DO5 terminal function selection	uint16	0~30	RW	NO	0	1	Shutdown effective
	0Ah	101h	DO5 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
2005h （Position control parameter）	01h	10Dh	Source of position command	uint16	0~65535	RW	NO	1	1	Shutdown effective
			Specific description: 0-pulse, 1-step							
	05h	112h	Step	int16	-32768~32767	RW	NO	100	inc	Shutdown effective
	1C	12Eh	Origin return time limit	uint16	0~65535	RW	NO	10000	ms	Shutdown effective
2006h (Speed control parameters)	01h	148h	Source of main speed command A	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0~digital setting (2006h: 04h setting value), 3~duty ratio setting							
	02h	149h	Source of auxiliary speed command B	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0-number setting (2006h: 04h setting value)							
	03h	14Ah	Speed command selection	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0-A (only effective at 2006h: 01h), 1-B (only effective at 2006h: 02h), 2-A+B (effective at both 2006h: 01h and 2006h: 02h)							
	04h	14Bh	Speed command keyboard setting	int16	-32768~32767	RW	NO	10	1rpm	Effective immediately
	06h	14Dh	Effective speed setting	int16	-32768~32767	RW	NO	0	1rpm	Effective immediately
	07h	14Eh	Speed command acceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	Effective immediately

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2006h (Speed control parameters)	08h	14Fh	Speed command deceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	Effective immediately
	09h	150h	Maximum speed threshold	uint16	0~65535	RW	NO	6000	1rpm	Effective immediately
	0Ah	151h	Forward speed threshold	uint16	0~65535	RW	NO	4000	1rpm	Effective immediately
	0Bh	152h	Reverse speed threshold	uint16	0~65535	RW	NO	4000	1rpm	Effective immediately
2007h (Torque control parameter)	01h	15Eh	Source of main torque command A	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0~digital setting (setting value of 2007h: 04h)							
	02h	15Fh	Source of auxiliary torque command B	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0-digit setting (2007h: 04h setting value)							
	03h	160h	Torque command selection	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 0-A (only effective at 2007h: 01h), 1-B (only effective at 2007h: 02h), 2-A+B (effective at both 2007h: 01h and 2007h: 02h)							
	04h	161h	Torque command keyboard setting	int16	-32768~32767	RW	NO	10	0.1%	Shutdown effective
	05h	162h	Effective torque command setting	int16	-32768~32767	RW	NO	10	0.1%	Shutdown effective
2007h (Torque control parameter)	06h	163h	Torque command filter time constant	uint16	0~65535	RW	NO	0	0.01ms	Effective immediately
			If the setting value is too large, the response will be reduced.							

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
	$0A_h$	167_h	Positive internal torque limit	uint16	0~65535	RW	NO	1000	0.1%	Effective immediately
	$0B_h$	168_h	Inverse internal torque limit	uint16	0~65535	RW	NO	1000	0.1%	Effective immediately
	10_h	$16D_h$	Torque control forward speed limit value	uint16	0~65535	RW	NO	3000	1rpm	Effective immediately
	11_h	$16E_h$	Torque control reverse speed limit	uint16	0~65535	RW	NO	3000	1rpm	Effective immediately
	13_h	170_h	Detecting torque when finding the origin in locked rotation (HM Blocking Torque)	uint16	0~65535	RW	NO	500	0.1%	
	15_h	172_h	Detection time when blocking to find the origin (HM Blocking Time)	uint16	0~65535	RW	NO	500	ms	
2008_h (Gain parameters)	01_h	178_h	Speed loop gain	uint16	1~20000	RW	NO	500	0.1Hz	Run settings take effect immediately
			Description: Set the proportional gain of the speed loop. This parameter determines the response of the speed loop. The larger the speed loop, the faster the response, but too large a setting may cause vibration. In 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	179_h	Speed loop integral time constant	uint16	0~51200	RW	NO	800	0.01ms	Run settings take effect immediately
			Description: Set the integral time constant of the speed loop. The smaller the set value, the stronger the integration effect, and the deviation value at the time of stopping is closer to 0 faster. When this parameter is set to 51200, there is no integral effect.							

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
	03h	17Ah	Position loop gain	uint16	0~20000	RW	NO	1500	1	Run settings take effect immediately
			Description: Set the proportional gain of the position loop. This parameter determines the responsiveness of the position loop. Setting a larger position loop gain can shorten the positioning time, but setting it too large may cause vibration.							
	04h	17Bh	Second speed loop gain	uint16	1~20000	RW	NO	1	0.1Hz	Effective immediately
	05h	17Ch	Integral time constant of the second speed loop	uint16	0~51200	RW	NO	15	0.01ms	Effective immediately
	06h	17Dh	Second position loop gain	uint16	0~20000	RW	NO	0	1	Effective immediately
	08h	17Fh	Second gain mode setting	uint16	0~9	RW	NO	0	1	Effective immediately
			9~Actual speed, when the absolute value of the actual speed exceeds (level + time lag) [rpm] status continues during the delay time, switch to the second gain, the absolute value of the actual speed is less than (level-time lag) When the state of [rpm] continues during the delay time, it returns to the first gain.							
	09h	180h	Gain switching condition selection	uint16	0~10	RW	NO	0	1	Effective immediately
			Specific description: 0~group 1 gain, 1~group 2 gain							
	0Ah	181h	Gain switching delay time	uint16	0~10000	RW	NO	0	0.1ms	Effective immediately
	0Bh	182h	Gain switching level	uint16	0~20000	RW	NO	0	1	Effective immediately
			The actual switching action is affected by both the level and the time lag. For the specific impact mode, see the description at 2008h:08h. Depending on the gain switching mode, the unit of the switching level will change accordingly.							
	0Ch	183h	Gain switching lag	uint16	0~20000	RW	NO	0	1	Effective immediately
			The actual switching action is affected by both the level and the time lag. For the specific impact mode, see the description at 2008h:08h. Depending on the gain switching mode, the unit of the switching level will change accordingly.							

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2008h (Gain parameters)	0Fh	186h	Speed feedforward filter time constant	uint16	0~65535	RW	NO	0	0.01ms	Effective immediately
	10h	187h	Speed feedforward gain	uint16	0~65535	RW	NO	0	0.1%	Effective immediately
			When adjusting, you should repeatedly adjust 2008h: 0Fh and 2008h: 10h to find a well-balanced setting							
	11h	188h	Torque feedforward filter time constant	uint16	0~65535	RW	NO	0	0.01ms	Effective immediately
	12h	189h	Torque feedforward gain	uint16	0~65535	RW	NO	0	0.001	Effective immediately
			When adjusting, you should repeatedly adjust 2008h: 11h and 2008h: 12h to find a well-balanced setting.							
2008h (Gain parameters)	14h	18Bh	Speed feedback low-pass filter cutoff frequency	uint16	0~4000	RW	NO	900	1Hz	Effective immediately
			Specific description: The smaller the setting, the smaller the speed feedback fluctuation, but the larger the feedback delay, generally keep the default parameters.							
2009h	06h	197h	OfflInertiaAutoTunMode	uint16	0~65535	RO	NO	0	1	-
	0Dh	19Eh	Group 1 notch frequency	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
			Description: Set the center frequency of the trap, that is, the mechanical resonance frequency.							

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
(Gain parameter)	0E$_h$	19F$_h$	Notch width of the first group of traps	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
			Specific description: Set the frequency band width on both sides of the center frequency of the mechanical resonance							
	0F$_h$	1A0$_h$	Group 1 trap depth	uint16	0~100	RW	NO	0	%	Effective immediately
			Specific description: 0~ no notch, 100~ maximum notch depth							
	10$_h$	1A1$_h$	Group 2 notch frequency	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	11$_h$	1A2$_h$	Notch width of the second group notch filter	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	12$_h$	1A3$_h$	Group 2 trap depth	uint16	0~100	RW	NO	0	1	Effective immediately
	13$_h$	1A4$_h$	Group 3 notch frequency	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	14$_h$	1A5$_h$	Group 3 notch width	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	15$_h$	1A6$_h$	Group 3 trap depth	uint16	0~100	RW	NO	0	1	Effective immediately
2009$_h$ (Gain parameter)	16$_h$	1A7$_h$	Group 4 notch frequency	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	17$_h$	1A8$_h$	4th wave trap notch width	uint16	0~2000	RW	NO	0	1Hz	Effective immediately
	18$_h$	1A9$_h$	Group 4 trap depth	uint16	0~100	RW	NO	0	1	Effective immediately

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
200Ah (Fault and protection parameters)	06h	1BC$_h$	Overspeed fault threshold	uint16	0~65535	RW	NO	0	1	Effective immediately
200B$_h$ (Monitoring parameters)	01$_h$	1DE$_h$	Motor drive internal state	uint16	0~12	RO	NO	0	1	-
			Specific description: 0~ not ready, 1~ ready, 6~ position closed loop control, 8~ speed closed loop control, 9~ torque control, 10~ open loop control, 12~ error							
	02$_h$	1DF$_h$	Actual motor speed	int16	-32768~32767	RO	NO	-	1rpm	-
	04$_h$	1E1$_h$	Internal torque command	int16	-32768~32767	RO	NO	0	0	-
	05$_h$	1E2$_h$	Input signal (DI signal) monitoring	uint16	0~65535	RO	NO	0	1	-
	06$_h$	1E3$_h$	Output signal (DO signal) monitoring	uint16	0~65535	RO	NO	0	1	-
	0A$_h$	1E8$_h$	Input PWM frequency	uint16	0~65535	RO	NO	0	1Hz	-
	0F$_h$	1F0$_h$	Total power-on time	uint32	0~4294967295	RO	NO	0	0.1s	-
	10	1F2$_h$	AI1 Sampling Voltage	uint16	-1200-1200	RO	NO	-1200	mV	
	11	1F3$_h$	AI2 Sampling Voltage	uint16	-1200-1200	RO	NO	-1200	mV	
	12$_h$	1F4$_h$	A phase current effective value	uint16	0~65535	RO	NO	0	0.01A	-
	13$_h$	1F5$_h$	B phase current effective value	uint16	0~65535	RO	NO	0	0.01A	-

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
200B$_h$ (Monitoring parameters)	14$_h$	1F6$_h$	C phase current effective value	uint16	0~65535	RO	NO	0	0.01A	-
	15$_h$	1F7$_h$	Bus voltage	uint16	0~65535	RO	NO	0	0.1V	-
	16$_h$	1F8$_h$	Module temperature value	int16	-32768~32767	RO	NO	0	1°C	-
	2D$_h$	211$_h$	Actual motor speed	uint16	0~65535	RO	NO	-	0.1rpm	-
200C$_h$ (Communication parameters)	01$_h$	22F$_h$	Selection of communication method	uint16	0~2	RW	NO	1	1	Power on again
			Specific description: 0~External pulse control (default CAN communication), 1~EtherCAT, 2~CAN							
	02$_h$	230$_h$	Drive axis address	uint16	1~247	RW	NO	1	1	Power on again
			1~247: When multiple motors are networked, each motor can only have a unique address, otherwise it will cause abnormal communication or unable to communicate.							
	03$_h$	231$_h$	Serial port baud rate setting	uint16	0~9	RW	NO	5	1	Power on again
			The communication rate of the motor must be the same as the communication rate of the host computer, otherwise it cannot communicate.							
	18$_h$	246$_h$	Synchronous mode enable	uint16	0~1	RW	NO	0	1	Power on again
	01$_h$	25B$_h$	Motor overload	int32	-214748364~2147483583	RW	NO	67183377	1	Effective immediately

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
200E$_h$ (error code)	02$_h$	25D$_h$	Motor blocked	int32	-214748364~2147483583	RW	NO	8978	1	Effective immediately
	03$_h$	25F$_h$	Power overvoltage	int32	-214748364~2147483583	RW	NO	78352	1	Effective immediately
	04$_h$	261$_h$	Power supply undervoltage	int32	-214748364~2147483583	RW	NO	16855584	1	Effective immediately
	05$_h$	263$_h$	Temperature is too high	int32	-214748364~2147483583	RW	NO	82448	1	Effective immediately
	06$_h$	265$_h$	The temperature is too low	int32	-214748364~2147483583	RW	NO	82464	1	Effective immediately
	07$_h$	267$_h$	Parameter setting error	int32	-214748364~2147483583	RW	NO	90912	1	Effective immediately
200E$_h$ (error code)	08$_h$	269$_h$	Motor overspeed	int32	-214748364~2147483583	RW	NO	94992	1	Effective immediately
	09$_h$	26B$_h$	Communication failure	int32	-214748364~2147483583	RW	NO	95489	1	Effective immediately

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
200E$_h$ (error code)	0A$_h$	26D$_h$	Origin return timeout error	int32	-214748364~2147483583	RW	NO	34320	1	Effective immediately
	0B$_h$	26F$_h$	Location error	int32	-214748364~2147483583	RW	NO	34321	1	Effective immediately
	0C$_h$	271$_h$	Software overrun fault	int32	-214748364~2147483583	RW	NO	50431507	1	Effective immediately
	0D$_h$	273$_h$	Over-limit switch failure	int32	-214748364~2147483583	RW	NO	50431507	1	Effective immediately
	OE$_h$	275$_h$	Output phase failure	int32	-214748364~2147483583	RW	NO	12933	1	Effective immediately
	0F$_h$	277$_h$	Curve planning calculation error	int32	-214748364~2147483583	RW	NO	67208725	1	Effective immediately
	10$_h$	279$_h$	Target position overflow	int32	-214748364~2147483583	RW	NO	99862	1	Effective immediately
	11$_h$	27B$_h$	Curve planning parameters are too small	int32	-214748364~2147483583	RW	NO	83985943	1	Effective immediately

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
200E$_h$ (error code)										
2011$_h$ (Multi-stage position function parameters)	01$_h$	291$_h$	Multi-step position operation mode	uint16	0~2	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 5.16.4 for details							
	02$_h$	292$_h$	Position command end segment number	uint16	1~16	RW	NO	1	1	Shutdown effective
			Description: Set the total number of multi-segment position commands							
	04$_h$	294$_h$	Multi-stage position cycle times	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: The number of cycles for the multi-segment position operation mode is 1							
	05$_h$	295$_h$	Position command type selection	uint16	0~1	RW	NO	0	1	Shutdown effective
			Specific description: 0-relative displacement; 1-absolute displacement Shutdown effective							
	06$_h$	296$_h$	Multi-segment position cycle start segment	uint16	0~16	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011h (Multi-stage position function parameters)			Specific description: the starting point segment with multi-segment position running mode of 1							
	07h	297h	1st stage displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	08h	299h	Maximum running speed of the first stage displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	09h	29Ah	1st stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
			Specific description: Set the time for the 1st stage motor of the multi-stage position to be changed from 0rpm to 1000rpm							
	0Ah	29Bh	Waiting time after the 1st displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
			Specific description: Waiting time before running the first displacement after the completion of the first displacement in the multi-segment position							
	0Bh	29Ch	2nd moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	0Ch	29Eh	Maximum running speed of 2nd stage displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	0Dh	29Fh	2nd stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011h (Multi-stage position function parameters)	0Eh	2A0h	Waiting time after completion of the 2nd displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	0Fh	2A1h	3rd moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	10h	2A3h	Maximum running speed of the third stage displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	11h	2A4h	3rd stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	12h	2A5h	Waiting time after the third stage displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	13h	2A6h	4th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	14h	2A8h	Maximum running speed of the 4th stage displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	15h	2A9h	4th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011_h (Multi-stage position function parameters)	16_h	$2AA_h$	Waiting time after the 4th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	17_h	$2AB_h$	5th moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	18_h	$2AD_h$	Maximum running speed of the 5th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	19_h	$2AE_h$	5th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1A_h$	$2AF_h$	Waiting time after the 5th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1B_h$	$2B0_h$	6th moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	$1C_h$	$2B2_h$	6th displacement maximum operating speed	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	$1D_h$	$2B3_h$	6th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011_h (Multi-stage position function parameters)	$1E_h$	$2B4_h$	Waiting time after the 6th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1F_h$	$2B5_h$	7th moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	20_h	$2B7_h$	7th displacement maximum running speed	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	21_h	$2B8_h$	7th stage displacement acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	22_h	$2B9_h$	Waiting time after completion of the 7th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	23_h	$2BA_h$	8th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	24_h	$2BC_h$	Maximum running speed of the 8th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	25_h	$2BD_h$	8th displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011h (Multi-stage position function parameters)	26h	2BEh	Waiting time after the 8th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	27h	2BFh	9th moving displacement	int32	-2147483583~2147483583	RW	NO	-10737418 24	User units	Shutdown effective
	28h	2C1h	9th displacement maximum operating speed	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	29h	2C2h	9th displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Ah	2C3h	Waiting time after the 9th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Bh	2C4h	10th movement displacement	int32	-2147483583~2147483583	RW	NO	-10737418 24	User units	Shutdown effective
	2Ch	2C6h	Maximum running speed of the 10th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	2Dh	2C7h	10th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011_h (Multi-stage position function parameters)	$2E_h$	$2C8_h$	Waiting time after the 10th displacement is completed	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$2F_h$	$2C9_h$	11th moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	30_h	$2CB_h$	Maximum running speed of the 11th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	31_h	$2CC_h$	11th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	32_h	$2CD_h$	Waiting time after the 11th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	33_h	$2CE_h$	12th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	34_h	$2D0_h$	Maximum running speed of the 12th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	35_h	$2D1_h$	12th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011h (Multi-stage position function parameters)	36h	2D2h	Waiting time after completion of the 12th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	37h	2D3h	13th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	38h	2D5h	Maximum running speed of the 13th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	39h	2D6h	Stage 13 displacement acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	3Ah	2D7h	Waiting time after completion of the 13th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	3Bh	2D8h	14th moving displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	3Ch	2DAh	Maximum running speed of the 14th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	3Dh	2DBh	14th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011_h (Multi-stage position function parameters)	$3E_h$	$2DC_h$	Waiting time after completion of the 14th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$3F_h$	$2DD_h$	15th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	40_h	$2DF_h$	Maximum running speed of the 15th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	41_h	$2E0_h$	15th stage displacement acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	42_h	$2E1_h$	Waiting time after the 15th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	43_h	$2E2_h$	16th movement displacement	int32	-2147483583~2147483583	RW	NO	-1073741824	User units	Shutdown effective
	44_h	$2E4_h$	Maximum running speed of the 16th displacement	uint16	0~3000	RW	NO	1	rpm	Shutdown effective
	45_h	$2E5_h$	16th stage displacement acceleration/deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	46_h	$2E6_h$	Waiting time after the 16th displacement	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2011_h (Multi-stage position function parameters)										
2012_h (Multi-stage speed function parameter)	01_h	$2E9_h$	Multi-step speed command operation mode	uint16	0~2	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 5.17.4 for details							
	02h	$2EA_h$	Speed command end segment selection	uint16	0~16	RW	NO	1	1	Shutdown effective
	03h	$2EB_h$	Multi-speed cycle times	uint16	0~65535	RW	NO	0	1	Shutdown effective
	$0C_h$	$2F4_h$	1st speed instruction	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	$0D_h$	$2F5_h$	Section 1 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$0E_h$	$2F6_h$	1st acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$0F_h$	$2F7_h$	Instruction 2	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2012_h (Multi-stage speed function parameter)	10_h	$2F8_h$	Section 2 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	11_h	$2F9_h$	2nd acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	12_h	$2FA_h$	Instruction 3	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	13_h	$2FB_h$	Section 3 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	14_h	$2FC_h$	3rd acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	15_h	$2FD_h$	Instruction 4	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	16_h	$2FE_h$	Section 4 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	17_h	$2FF_h$	4th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	18_h	300_h	Instruction 5	int16	0~3000	RW	NO	-6000	rpm	
	19_h	301_h	Section 5 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1A_h$	302_h	5th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1B_h$	303_h	Instruction 6	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	$1C_h$	304_h	Section 6 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1D_h$	305_h	6th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	$1E_h$	306_h	Instruction 7	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2012h (Multi-stage speed function parameter)	1Fh	307h	Section 7 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	20h	308h	7th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	21h	309h	Instruction 8	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	22h	30Ah	Section 8 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	23h	30Bh	8th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	24h	30Ch	Instruction 9	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	25h	30Dh	Section 9 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	26h	30Eh	9th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	27h	30Fh	Instruction 10	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	28h	310h	Section 10 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	29h	311h	10th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Ah	312h	Instruction 11	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	2Bh	313h	Section 11 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Ch	314h	11th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Dh	315h	Instruction 12	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2012h (Multi-stage speed function parameter)	2Eh	316h	Section 12 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	2Fh	317h	12th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	30h	318h	Instruction 13	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	31h	319h	Section 13 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	32h	31Ah	13th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	33h	31Bh	Instruction 14	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	34h	31Ch	Section 14 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	35h	31Dh	14th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	36h	31Eh	Instruction 15	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	37h	31Fh	Section 15 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	38h	320h	15th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	39h	321h	Instruction 16	int16	0~3000	RW	NO	-6000	rpm	Shutdown effective
	3Ah	322h	Section 16 instruction running time	uint16	0~65535	RW	NO	0	ms	Shutdown effective
	3Bh	323h	16th acceleration/ deceleration time	uint16	0~65535	RW	NO	0	ms	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2017_h (Virtual input terminal parameters)	01_h	326_h	VDI1 terminal function selection	uint16	0~44	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 6.1							
	02_h	327_h	VDI1 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
			Specific description: see chapter 6.1							
	03_h	328_h	VDI2 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	04_h	329_h	VDI2 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
2017_h (Virtual input terminal parameters)	05_h	$32A_h$	VDI3 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	06_h	$32B_h$	VDI3 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	07_h	$32C_h$	VDI4 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	08_h	$32D_h$	VDI4 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	09_h	$32E_h$	VDI5 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2017_h (Virtual input terminal parameters)	$0A_h$	$32F_h$	VDI5 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	$0B_h$	330_h	VDI6 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	$0C_h$	331_h	VDI6 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	$0D_h$	332_h	VDI7 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	$0E_h$	333_h	VDI7 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	$0F_h$	334_h	VDI8 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	10_h	335_h	VDI8 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	11_h	336_h	VDI9 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	12_h	337_h	VDI9 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	13_h	338_h	VDI10 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	14_h	339_h	VDI10 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	15_h	$33A_h$	VDI11 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
2017h	16h	33Bh	VDI11 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	17h	33Ch	VDI12 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	18h	33Dh	VDI12 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	19h	33Eh	VDI13 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	1Ah	33Fh	VDI13 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	1Bh	340h	VDI14 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	1Ch	341h	VDI14 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	1Dh	342h	VDI15 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	1Eh	343h	VDI15 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective
	1Fh	344h	VDI16 terminal function selection	uint16	0~39	RW	NO	0	1	Shutdown effective
	20h	345h	VDI16 terminal logic selection	uint16	0~1	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding modbus address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
(Virtual input terminal parameters)										
2031_h (Communication given relevant variables)	01_h	373_h	Communication given VDI virtual level	uint16	0~65535	RW	NO	0	1	Shutdown effective
			Specific description: 2031h: 01h bit(n)=1 means VDI(n+1) is high level; bit(n)=0 means VDI(n+1) is low level							
	02_h	374_h	Communication given Do output status	uint16	0~65535	RW	NO	0	1	Shutdown effective
			When logic selection in 2004h is negative, 2031h: 02h=1, Do outputs low level; 2031h: 02h=0, Do outputs high level							

10.3 Sub-protocol definition parameter 6000h group description

Table 10-7

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
603F$_h$	00$_h$	37F$_h$	Error code, see chapter 8.1 for details	uint16	0~65535	RO	TPDO	0	-	-
6040$_h$	00$_h$	380$_h$	Control word	uint16	0~65535	RW	RPDO	0	1	Shutdown effective
6041$_h$	00$_h$	381$_h$	Status word	uint16	0~65535	RO	TPDO	0	1	-
6042$_h$	00$_h$	382$_h$	Target speed in VM mode	int16	-3000~3000	RW	RPDO	0	1rpm	Shutdown effective
6043$_h$	00$_h$	383$_h$	Target speed for VM mode to take effect	int16	-3000~3000	RO	TPDO	0	1rpm	-
6044$_h$	00$_h$	384$_h$	Actual speed in VM mode	int16	--2147483583~2147483583	RO	TPDO	0	1rpm	-
6046$_h$	01$_h$	385$_h$	Minimum speed in VM mode	uint32	0~6000	RW	RPDO	10	1rpm	Shutdown effective
	02$_h$	387$_h$	Maximum speed in VM mode	uint32	0~6000	RW	RPDO	3000	1rpm	Shutdown effective
6048$_h$	01$_h$	389$_h$	Acceleration in VM mode = delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Shutdown effective
	02$_h$	38B$_h$		uint16	1~65535	RW	RPDO	1	1s	Shutdown effective
6049$_h$	01$_h$	38C$_h$	VM mode deceleration = delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Shutdown effective

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
	02_h	$38E_h$		uint16	1~65535	RW	RPDO	1	1s	Shutdown effective
$604A_h$	01_h	$38F_h$	Emergency stop speed in VM mode = delta speed/delta time	uint32	0~4294967295	RW	RPDO	800	1rpm	Shutdown effective
	02_h	391_h		uint16	1~65535	RW	RPDO	1	1s	Shutdown effective
$604C_h$	01_h	394_h	Scaling factor for VM mode	int32	-2147483583~2147483583	RW	RPDO	0	1	Shutdown effective
	02_h	396_h		int32	-2147483583~2147483583	RW	RPDO	0	1	Shutdown effective
$605A_h$	00_h	$3BF_h$	For quick stop mode selection, see chapter 3.1.4 for details	int16	0~2	RW	NO	2	1	Shutdown effective
605Bh	00	3BDh	Shutdown option code selection, see chapter 3.1.4 for details	int16	0~1	RW	NO	0	1	Shutdown effective
605Ch	00	3BEh	Disable operation option code selection, see chapter 3.1.4 for details	int16	0~1	RW	NO	0	1	Shutdown effective
$605D_h$	00_h	$3C0_h$	For the selection of pause mode, please refer to chapter 3.1.4	int16	1~2	RW	NO	1	1	Shutdown effective

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
605E$_h$	00$_h$	3C1$_h$	For the selection of fault shutdown mode, please refer to chapter 3.1.4	int16	0~2	RW	NO	0	1	Shutdown effective
6060$_h$	00$_h$	3C2$_h$	Mode selection, see chapter 5.1 for details	int8	0~10	RW	RPDO	1	1	Shutdown effective
6061$_h$	00$_h$	3C3$_h$	Operating mode display, see chapter 5.2 for details	int8	0~10	RO	TPDO	1	1	-
6062$_h$	00$_h$	3C4$_h$	The currently valid target position command value inside the drive	int32	-2147483583~2147483583	RO	TPDO	0	User units	-
6063$_h$	00$_h$	3C6$_h$	Motor current absolute position feedback	int32	-2147483583~2147483583	RO	TPDO	0	Encoder unit	-
6064$_h$	00$_h$	3C8$_h$	Motor current user absolute position feedback	int32	-2147483583~2147483583	RO	TPDO	0	User units	-
			Specific description: position feedback 6064$_h$× gear ratio (6091$_h$) = position feedback 6063$_h$							
6065$_h$	00$_h$	3CA$_h$	Threshold for excessive position deviation	uint32	0~4294967295	RW	RPDO	50	User units	Shutdown effective
			Specific description: When the position deviation exceeds ±6065h, and the time reaches 6066h, a tracking error is reported							
6066$_h$	00$_h$	3CC$_h$	Excessive position deviation timeout	uint16	0~65535	RW	RPDO	30000	1ms	Shutdown effective
			Specific description: When the position deviation exceeds ±6065h, and the time reaches 6066h, a tracking error is reported							

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
6067h	00h	3CDh	Location reached threshold	uint32	0~4294967295	RW	RPDO	10	User units	Shutdown effective
			Specific description: When the position deviation is within ±6067h, and the time reaches 6068h, the position is considered to be reached. In position mode, bit10 of status word 6041h=1							
6068h	00h	3CFh	Position arrival time window	uint16	0~65535	RW	RPDO	5	1ms	Shutdown effective
			Specific description: When the position deviation is within ±6067h and the time reaches 6068h, it is considered that the position has arrived. In position mode, bit10 of status word 6041h=1							
6069h	00h	3D0h	Feedback value of speed sensor	int32	-2147483583~2147483583	RO	TPDO	0	User unit/s	-
606Bh	00h	3D3h	Speed command value valid inside the controller	int32	-2147483583~2147483583	RO	RPDO	0	User unit/s	-
606Ch	00h	3D5h	Current actual speed feedback value	int32	-2147483583~2147483583	RO	TPDO	0	rpm	-
606Dh	00h	3D7h	Speed reaches threshold	uint16	0~65535	RW	RPDO	500	rpm	Shutdown effective
			Specific description: The difference between the target speed 60FFh (converted to the motor speed rpm) and the actual motor speed is within ±606D, and when the time reaches 606Eh, the speed is considered to be reached, and bit10=1 of status word 6041h							
606Eh	00h	3D8h	Speed arrival time window	uint16	0~65535	RW	RPDO	5	ms	Shutdown effective
			Specific description: The difference between the target speed 60FFh (converted to the motor speed rpm) and the actual motor speed is within ±606D, and when the time reaches 606Eh, the speed is considered to be reached, and bit10=1 of status word 6041h							

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
606F$_h$	00$_h$	3D9$_h$	Zero reaches the threshold	uint16	0~65535	RW	RPDO	5	User unit/s	Shutdown effective
			Specific description: The current speed of 6069h is within ±606F, and the time reaches 6070h. It is considered that the zero speed has arrived, and bit12=1 of the status word 6041h							
6070$_h$	00$_h$	3DA$_h$	Zero-degree arrival time window	uint16	0~65535	RW	RPDO	5	ms	Shutdown effective
			Specific description: The current speed of 6069h is within ±606F, and the time reaches 6070h. It is considered that the zero speed has arrived, and bit12=1 of the status word 6041h							
6071$_h$	00$_h$	3DB$_h$	Target torque	int16	-1000~1000	RW	RPDO	0	0.10%	Shutdown effective
			Specific description: 100.0% corresponds to 1 times the rated torque of the motor							
6072$_h$	00$_h$	3DC$_h$	Max Torque	uint16	0~2000	RW	RPDO	1000	0.10%	Shutdown effective
6073$_h$	00$_h$	3DD$_h$	Maximum current	uint16	0~2000	RW	NO	1500	0.10%	Shutdown effective
6074$_h$	00$_h$	3DE$_h$	Torque command, reflecting the torque command that has been input in the enabled state	int16	-32768~32767	RO	TPDO	0	0.10%	-
6075$_h$	00$_h$	3DF$_h$	Rated current	uint32	0~11700	RW	RPDO	11700	0.001A	Shutdown effective
6076$_h$	00$_h$	3E1$_h$	Rated torque	uint32	0~1270	RW	RPDO	1270	0.01N.m	Shutdown effective
6077$_h$	00$_h$	3E3$_h$	Torque feedback value	int16	-32768~32767	RO	TPDO	-	0.001N.m	-

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
6078h	00h	3E4h	Current feedback value (corresponding to the percentage of rated current)	int16	-32768~32767	RO	TPDO	-	0.10%	-
607Ah	00h	3E7h	Target location	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
607Bh	01h	3E9h	Position range minimum limit	int32	-2147483583~2147483583	RW	RPDO	-1048576	User units	Shutdown effective
	02h	3EBh	Maximum limit of location range	int32	-2147483583~2147483583	RW	RPDO	1048576	User units	Shutdown effective
607Ch	00h	3EDh	Origin offset value	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
			Specific description: Set the physical position where the mechanical zero point deviates from the motor origin in the homing mode; when the homing operation is completed, bit 15 of status word 6041h becomes effective							
607Dh	01h	3EFh	Software absolute position minimum limit	int32	-2147483583~2147483583	RW	RPDO	-65535	User units	Shutdown effective
	02h	3F1h	Maximum absolute position of software	int32	-2147483583~2147483583	RW	RPDO	65535	User units	Shutdown effective
607Eh	00h	3F3h	Command polarity, see chapter 4.3 for details	uint8	0~255	RW	NO	0	1	Shutdown effective

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
607F$_h$	00$_h$	3F4$_h$	Maximum profile speed	uint32	0~1000000	RW	RPDO	500000	User unit/s	Shutdown effective
6080$_h$	00$_h$	3F6	Motor speed	uint32	0~6000	RW	RPDO	4000	rpm	Shutdown effective
6081$_h$	00$_h$	3F8$_h$	profile speed	uint32	0~1000000	RW	RPDO	500000	User unit/s	Shutdown effective
			Specific description: Set the uniform speed of the displacement command in the profile position mode, the specific calculation is as follows: $\text{motor speed (rpm)} = \frac{6081h * \text{gear ratio (6091h)}}{\text{Encoder resolution}} * 60$							
6082$_h$	00$_h$	3FA$_h$	profile end speed	uint32	0~1000000	RW	RPDO	0	User unit/s	Shutdown effective
6083$_h$	00$_h$	3FC$_h$	profile acceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Shutdown effective
6084$_h$	00$_h$	3FE$_h$	profile deceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Shutdown effective
6085$_h$	00$_h$	400$_h$	Quick stop deceleration	uint32	0~4294967295	RW	RPDO	500000	User unit/s^2	Shutdown effective
6086$_h$	00$_h$	402$_h$	Type of profile slope	int16	0~3	RW	RPDO	3	1	Shutdown effective
			Description: Set the curve type of motor position command or speed command. 0~linear, 3~S curve							
6087$_h$	00$_h$	403$_h$	Torque ramp	uint32	0~4294967295	RW	RPDO	10	0.10%	Shutdown effective

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
			Specific description: Set the torque command acceleration in profile torque mode, its meaning is: torque command increment per second							
6088h	00h	405h	Torque ramp type	int16	0~2	RW	RPDO	0	1	Shutdown effective
			Specific description: 0~slope, 2~no slope							
608Fh	01h	406h	Encoder increment	uint32	0~4294967295	RW	NO	10000	1	Shutdown effective
	02h	408h	Motor speed	uint32	0~4294967295	RW	NO	1	1	
6091h	01h	40Eh	Transmission ratio	uint32	0~4294967295	RW	NO	1	1	Shutdown effective
	02h	410h		uint32	0~4294967295	RW	NO	1	1	Shutdown effective
6098h	00h	416h	For the origin return method, see chapter 5.7.4 for details	int8	17~45	RW	RPDO	20	1	Shutdown effective
6099h	01h	417h	Find the switching speed (high speed)	uint32	0~4294967295	RW	RPDO	10000	User unit/s	Shutdown effective
6099h	02h	419h	Find origin speed (low speed)	int32	-2147483583~2147483583	RW	RPDO	2730	User unit/s	Shutdown effective
609Ah	00h	41Bh	Origin return acceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s	Shutdown effective
60A3h	00h	41Dh	The number of profile acceleration used, this device	Usint8	2	RW	RPDO	2	1	Shutdown effective

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
			only uses $60A4_h$:01_h and $60A4_h$:02_h							
$60A4_h$	01_h	$41E_h$	profile acceleration	uint32	0~4294967295	RW	NO	50000	User unit/s^2	Shutdown effective
	02_h	420_h	profile deceleration	uint32	0~4294967295	RW	NO	50000	User unit/s^2	Shutdown effective
$60B0_h$	00_h	426_h	Position offset	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
			Specific description: Set the position command offset in the cycle synchronous position mode, after offset: target position=$607A_h$+$60B0_h$							
$60B1_h$	00_h	428_h	Speed offset	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
			Specific description: Set the speed command offset in cycle synchronous Velocity mode, after offset: target speed=60FFh+60B1h							
$60B2_h$	00_h	$42A_h$	Torque bias	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
			Specific description: Set the torque command offset in the cyclic torque mode, after offset: target torque = 6071_h+$60B2_h$							
$60C1_h$	01_h	$42D_h$	Interpolated target position	int32	-2147483583~2147483583	RW	RPDO	0	User units	Shutdown effective
$60C2_h$	01_h	$42F_h$	Interpolation cycle time constant t	uint8	0~255	RW	NO	20	s	Shutdown effective

<table>
<tr><th>Index</th><th>Subindex</th><th>Corresponding MODBUS address</th><th>Description</th><th>Type of data</th><th>Data range</th><th>Accessibility</th><th>Can it be mapped</th><th>Factory setting</th><th>Unit</th><th>Effective method</th></tr>
<tr><td></td><td>02_h</td><td>430_h</td><td>Interpolation cycle time index n</td><td>Int8</td><td>-128~127</td><td>RW</td><td>NO</td><td>-1</td><td>1</td><td>Shutdown effective</td></tr>
<tr><td>$60C5_h$</td><td>00_h</td><td>$43B_h$</td><td>Maximum profile acceleration</td><td>uint32</td><td>0~4294967295</td><td>RW</td><td>RPDO</td><td>500000</td><td>User unit/s^2</td><td>Shutdown effective</td></tr>
<tr><td>$60C6_h$</td><td>00_h</td><td>$43D_h$</td><td>Maximum profile deceleration</td><td>uint32</td><td>0~4294967295</td><td>RW</td><td>RPDO</td><td>500000</td><td>User unit/s^2</td><td>Shutdown effective</td></tr>
<tr><td rowspan="2">$60F2_h$</td><td rowspan="2">00_h</td><td>$43F_h$</td><td>Targeting options</td><td>uint16</td><td>0~2</td><td>RW</td><td>RPDO</td><td>0</td><td>1</td><td>Shutdown effective</td></tr>
<tr><td></td><td colspan="7">The specific description is as follows:
<table>
<tr><th>Bit1</th><th>Bit0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Relative position movement, relative to the target position ($607A_h$)</td></tr>
<tr><td>0</td><td>1</td><td>Relative position movement, relative to the input position command ($60FC_h$)</td></tr>
<tr><td>1</td><td>0</td><td>Relative position movement, relative to the input position command (6064_h)</td></tr>
</table></td></tr>
<tr><td>$60F4_h$</td><td>00_h</td><td>440_h</td><td>User position deviation</td><td>int32</td><td>-2147483583~2147483583</td><td>RO</td><td>TPDO</td><td>0</td><td>User units</td><td>-</td></tr>
<tr><td>$60FA_h$</td><td>00_h</td><td>444_h</td><td>Regulator output</td><td>int32</td><td>-2147483583~2147483583</td><td>RO</td><td>TPDO</td><td>0</td><td>Command unit</td><td>-</td></tr>
</table>

Index	Subindex	Corresponding MODBUS address	Description	Type of data	Data range	Accessibility	Can it be mapped	Factory setting	Unit	Effective method
60FC$_h$	00$_h$	446$_h$	Motor position command	int32	-2147483583~2147483583	RO	TPDO	0	User units	-
			Specific description: position command 60FCh=position command 6062h×electronic gear ratio (6091$_h$)							
60FDh	00$_h$	44C$_h$	Input signal (DI signal) monitoring	uint32	0~4294967295	RO	TPDO	0	1	-
60FEh	01$_h$	44E$_h$	Physical output	uint32	0~4294967295	RO	TPDO	0	1	-
	02$_h$	450$_h$	Physical output enable	uint32	0~4294967295	RO	TPDO	0	1	-
60FF$_h$	00$_h$	448$_h$	Target speed	int32	-2147483583~2147483583	RW	RPDO	0	User unit/s	Shutdown effective
			Specific description: Set the user speed command in the profile Velocity mode and cycle synchronous Velocity mode							

11 Version Information

Manual version	Date	Modify record
A	2025/1/18	Establish
A1	2025/8/30	Modify description
A02	2026/1/15	Modify description
A03	2026/6/27	1. Modify some parts 2. Delete the anti-hand-clinging function

- All or part of this manual is prohibited to be reproduced or copied without permission.
- Product performance, specifications and appearance may change due to improvements without prior notice, please understand.
- We strive to make the content 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
Postcode： 102628
Mob： (010)60213882 Fax： (010)60213882
Email： NiMotion@NiMotion.com
WEB：http：//www.NiMotion.com