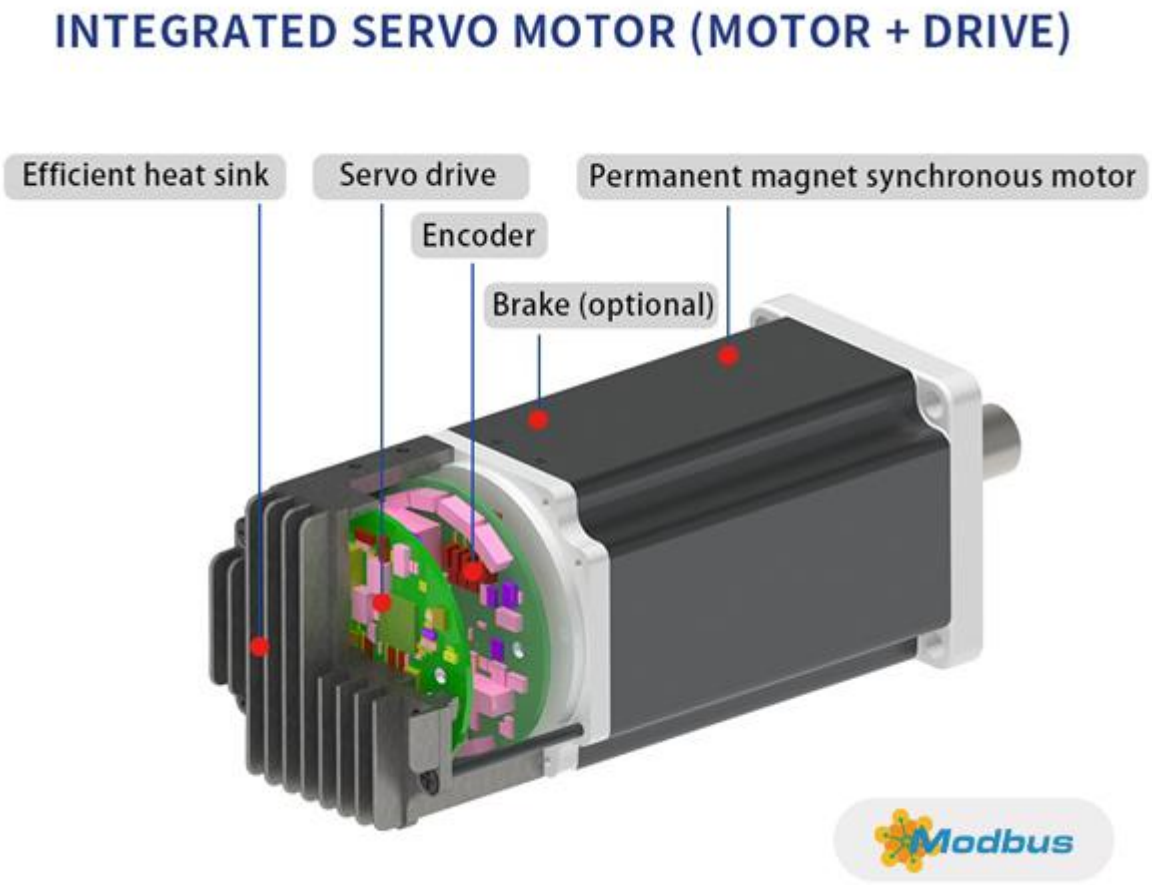
Integrated Low-voltage Servo Motor

Modbus Communication User Manual

Version number: A02



Beijing NiMotion Control Technology Co., Ltd

CONTENTS

1 About the manual

1.1 Introduction

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

1.2 Meaning of Numerical Values

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

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

<index>:<sub-index>

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

1.3 Bit

Bits of an object always start from 0 at LSB.

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

55_{hex} =85_{dec}

2 Communications Network

2.1 Modbus Overview

The communication mode of the Modbus protocol is of the single-master/multi-slave type. Only the master station can send out queries (inquiries). The slave stations carry out the processing as required by the queries and reply with response messages.

2.2 Modbus-RTU Transmission Mode

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

Frame Description:

Table2-1

Child Node Address	Function Code	Digital	CRC
1 byte	1 byte	0 to 252 bytes	2 bytes CRC low CRC high

The message frame is distinguished by an idle interval of at least 3.5 character time, called t3.5, and this module adopts the standard t3.5 as the idle interval distinction. the RTU frame transmission idle interval distinction diagram is shown in Figure 1.

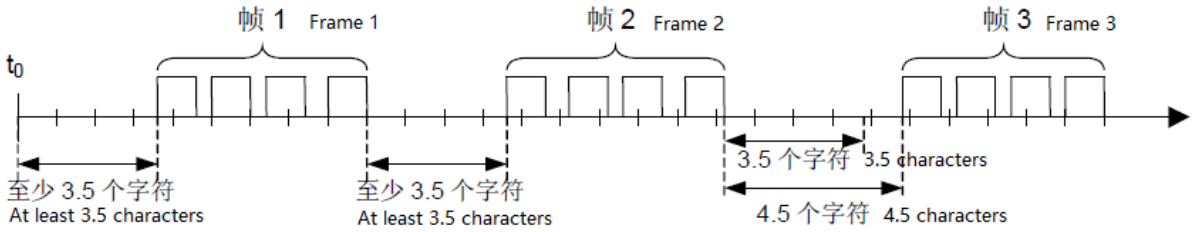


Figure 2-1 Schematic diagram of RTU frame transmission idle interval differentiation

2.3 Data Type

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

Table2-2

Basic Forms	Object Type	Type of Access	Element
Input register	16-bit	read-only	The I/O system provides this type of data
Holding register	16-bit	Read and write	Changing this type of data through the application

2.4 Modbus-RTU Function Code

Table2-3

	Function Code Name	Function Code	Sub-function Code	Broadcast Message
16-bit access	Read Input Register	0x04	-	unsupported
	Read Holding Register	0x03	-	unsupported
	Write Single Register	0x06	-	supported
	Write multiple registers	0x10	-	supported

2.4.1 Read Holding Register (0x03)

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

request message

Table2-4

Slave Address	Function Code	Register Address	Number of Registers	CRC Value
01	0x03	02 31	00 01	-

response message

Table 2-5

Slave Address	Function Code	Byte Count	Register Value	CRC Value
01	0x03	02	00 00	-

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

2.4.2 Read Input Register (0x04)

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

request message

Table2-6

Slave Address	Function Code	Register Address	Number of Registers	CRC Value
01	0x04	01 F7	00 01	-

response message

Table 2-7

Slave Address	Function Code	Byte Count	Register Value	CRC Value
01	0x04	02	00 E7	-

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

2.4.3 Write Single Holding Register (0x06)

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

request message

Table2-8

Slave Address	Function Code	Register Address	Register Value	CRC Value
01	0x06	02 31	00 01	-

response message

Table2-9

Slave Address	Function Code	Byte Count	Register Value	CRC Value
01	0x06	02 31	00 01	-

2.4.4 Write Multiple Holding Registers (0x10)

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

request message

Table2-10

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
01	0x10	02 30	00 02	04	00 02 00 01	-

response message

Table2-11

Slave Address	Function Code	Register Address	Number of Registers	CRC Value
01	0x10	02 30	00 02	-

2.5 Communication Exception Response

When a master device sends a request to a slave device, the master expects a normal response. The master's query may result in one of the following four events:

- If the slave device receives the request without communication errors and can handle the query normally, then the slave device will return a normal response.
- If the slave does not receive the request due to a communication error, the response cannot be returned. The master program will process it as a timeout.
- If the slave receives a request but detects a communication error (parity, CRC...), then no response can be returned. The master program will be processed on timeout.
- If the slave device receives a request for no communication error, but cannot process the request (e.g., if requesting a read of a non-existent input register), then the slave will return an exception response informing the client of the cause of the error.

The exception response message has two fields that are different from the normal response message:

Function Code Field:

In a normal response, the slave assigns the function code of the original request in the function code field of the response. All function codes have an MSB of 0 (they all have values below hex 80). In an exception response, the slave sets the MSB of the function code to 1. This makes the function code value in the exception response hex 80 higher than the function code value in the normal response.

By setting the function code MSB, the master's application program recognizes the exception response and can detect the data fields of the exception code.

Data Fields:

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

In an exception response, the slave returns an exception code in the data field. This indicates the reason for the exception.

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

All exception function codes are 0x83, 0x84, 0x86, and 0x90.

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

Table 2-12

Slave Address	Function Code	Exception Code	CRC Value

01	0x86	01	-

Table2-13

Exception Code	Exception Name	Description
01	Illegal function code	Function code cannot be recognized, not within 0x00~0x0F.
02	Illegal data address	The data address exceeds the definition.
03	Illegal data value	Values outside of register storage
04	Slave equipment failure	Generate non-recapturable errors
05	Recognize	Slaves take longer to process, requiring frequent interrogation by the master
06	Slave equipment busy	Wait for the host to send a request while the slave device is idle
12	Slave equipment alarm	Alarm exists for 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 parameters need to be saved after setting the baud rate, and will take effect after the device is next powered on or rebooted.

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

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

Request messages sent

Table2-14

Slave Address	Function Code	Register Address	Register Value	CRC Value
01	0x06	02 31	00 05	-

response message

Table2-15

Slave Address	Function Code	Register Address	Register Value	CRC Value
01	0x06	02 31	00 05	-

Saving the set communication baud rate parameter.

request messages

Table2-16

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
01	0x10	00 26	00 02	04	65 76 61 73	-

response message

Table2-17

Slave Address	Function Code	Register Address	Number of Registers	CRC Value
01	0x10	00 26	00 02	-

Table of setting values and correspondence with baud rate:

Table2-18

Hexadecimal	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 by setting the slave address $200C_h$:02_h (0230_h), and the range of setting is 1~247. The factory default slave address is 0x01. After the successful setting of the slave address, it is necessary to save the parameter setting, and it will take effect after the device is powered on or rebooted next time.

2.6.3 Network Data Format

The network data format for device communication is changed by setting the network data format $200C_h$: 04_h (0232_h). The corresponding table of setting values is shown in the table below.

Table2-19

hexadecimal	descriptive
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 Connection of Communication Lines

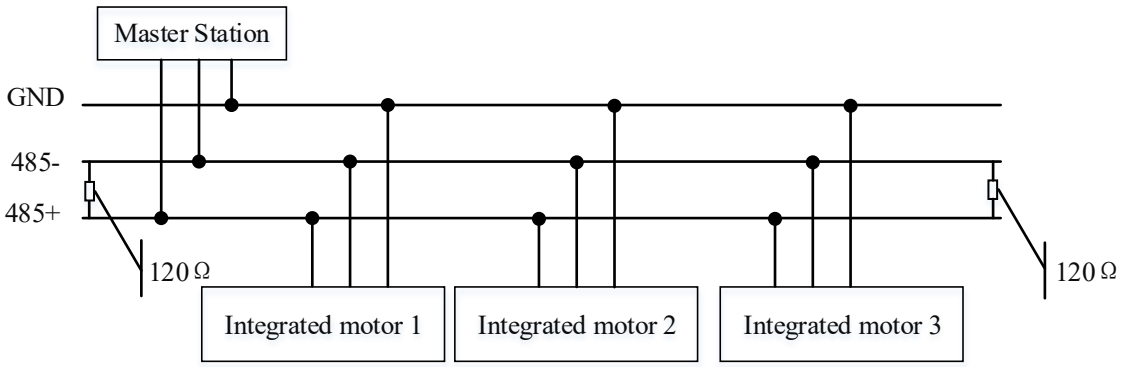


Figure2-2

2.6.5 Broadcast Seizure

When the communication parameters of the motor are not sure to forget, it can be restarted after power failure by broadcasting telegram to seize it. The broadcast preemption request message should be sent within 1 second before the motor is powered on, and the RUN indicator blinks after successful preemption. After successful seizure, the communication parameters of the motor are restored to the default parameters, i.e., 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, there are 8 data bits and 1 stop bit. The motor serial number adopts the byte sequence of hexadecimal high byte in the front.

Table 2-20 Broadcast preemption request messages

Slave Address (1B)	Function Code (1B)	Motor Serial Number (4B)	CRC Value (2B)
0x00	0xD2	0x00 00 00 00	-

3 Equipment Control

3.1 The CiA402 State Machine

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

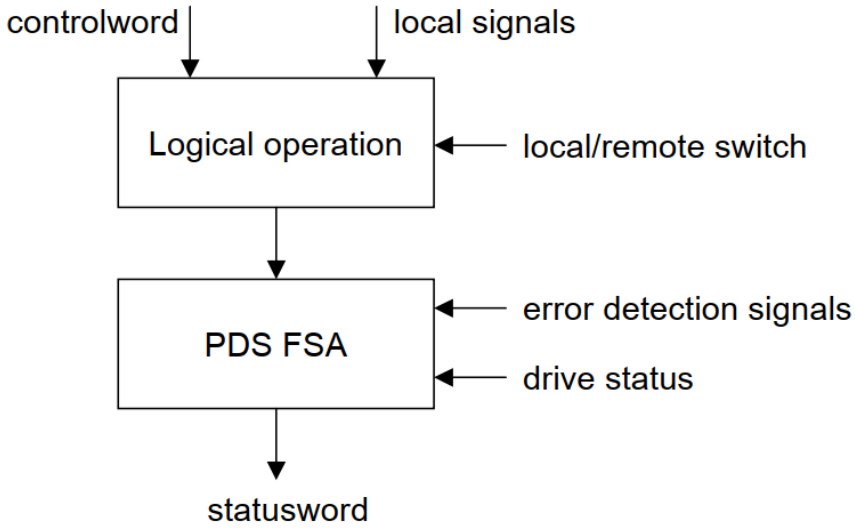


Figure3-1

3.1.1 The State Transition Diagram

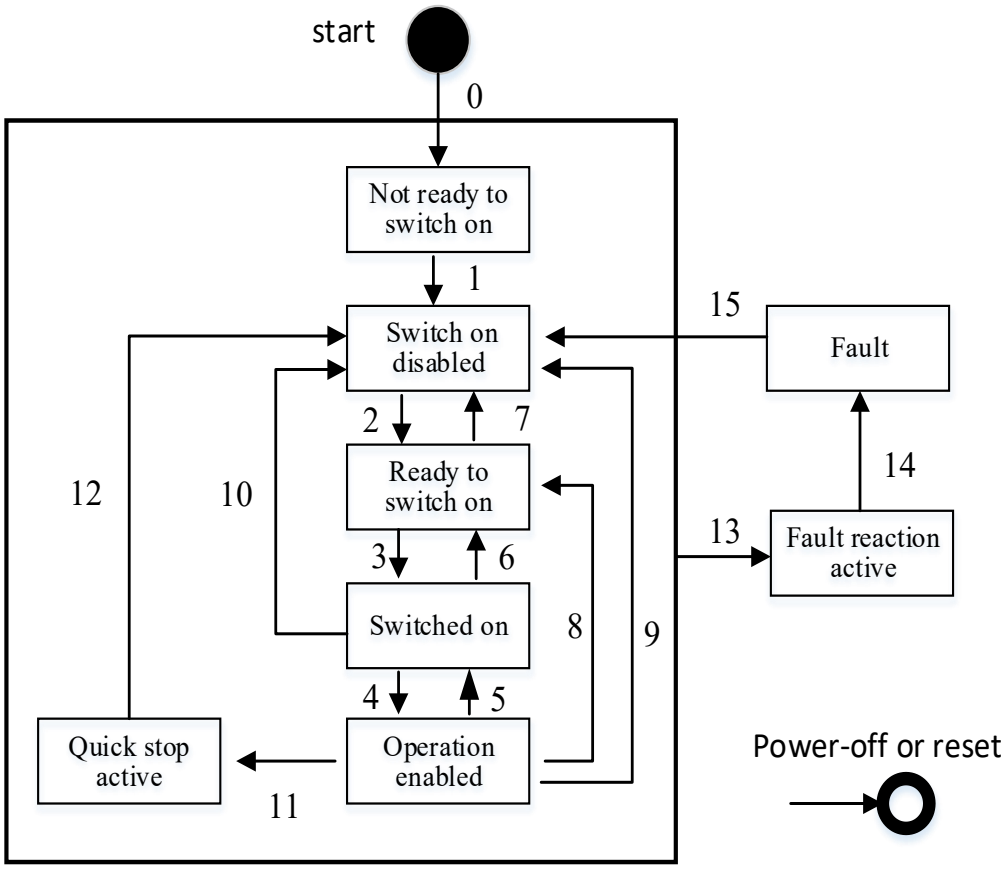


Figure3-2

Each state corresponds to the motor as shown in the table below:

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

The control word and state machine state switching is shown in the table below:

Table3-2

Command	Bits of the controlword					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 for each state are listed below:

Table3-3

Statusword	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 this bit does not been care, bit 8 ~ bit15 has different meaning in different modes. Please check its specific status in each mode.

3.1.2 Control Word

The bits of the control word, defined as follows:

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

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

- Note

It is meaningless to assign each bit of the control word individually,It must be combined with other bits to form a control instruction. Bit 0 ~bit 3 and bit 7 have the same meaning in each mode, Only by sending commands

sequentially can the motor drive be guided into the predicted state according to the switching process of the CiA402 state machine,and each command corresponds to 1 definitive state. Bit4 ~ bit6 are related to each operation mode (please check the control instructions in different modes).

3.1.3 Status Word

The bits of the status word are defined as follows:

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

Table3-4

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

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

3.1.4 The Stop Mode

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

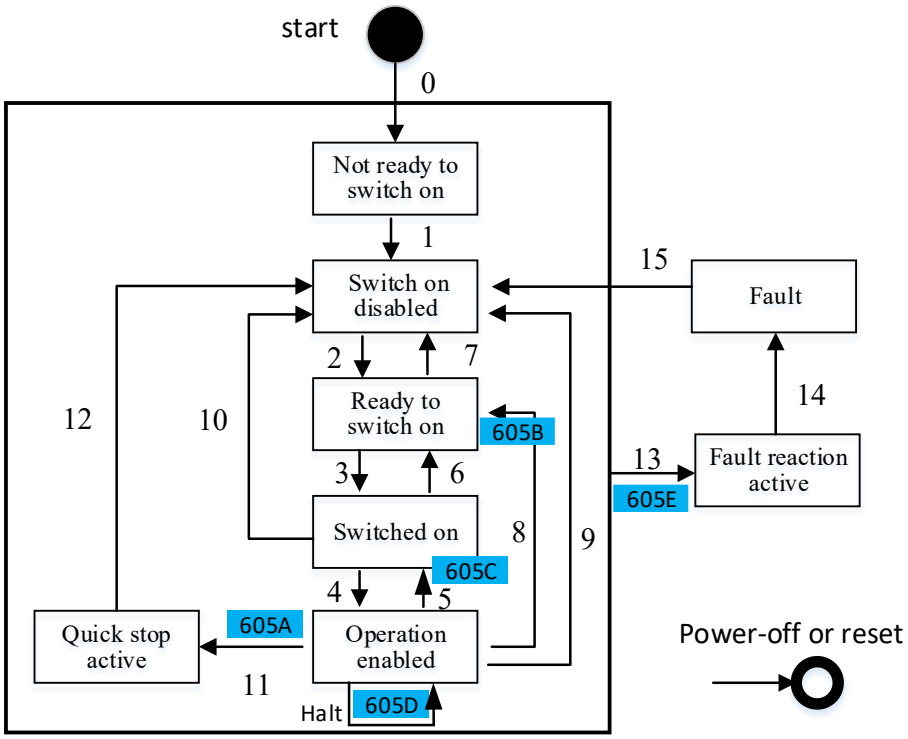


Figure3-3

- **Fault**

When a error or a warning occurs, automatically the servo switches to the Fault state which is related to $605E_h$ ($03C1_h$) (Fault response option code). When $605E_h$ ($03C1_h$)= -1, the fault reaction is related to the reaction code in the fault code which is configurable. When $605E_h$ ($03C1_h$)$_h$ > 0, the fault reaction is selected by the object dictionary $605E_h$ ($03C1_h$). The relationship between the setting value in $605E_h$ ($03C1_h$) and the fault reaction is as follows:

Table3-5

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

- **Quick Stop**

A quick stop is performed when the control word 6040_h (0380_h) = 0x02 in a non-fault condition.

The quick stop method is selected via the object dictionary $605A_h$ ($03BF_h$) and the meaning of the set values for $605A_h$ ($03BF_h$) is shown in the table below:

Table3-6

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

- **Halt**

Bit8 of control word 6040h (0380h) is the pause function bit, and the pause stop is executed when bit8 changes from 0 to 1.

The pause stop mode is selected by the object dictionary 605Dh (Halt option code), and the meaning of the setting value of 605Dh (03C0h) is shown in the following table:

Table3-7

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

- **Disabled Operation**

The deactivation mode is selected using the object dictionary 605Ch (03BEh), and the set values and meanings of 605Ch (03BEh) are shown in the table below:

Table3-8

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

- **Shutdown**

The shutdown is selected via the object dictionary 605Bh (03BDh) and the meaning of the set values for 605Bh (03BDh) is shown in the table below:

Table3-9

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

4 Unit Conversion

In order to adapt to applications in various industries and to facilitate users to define their respective command units, the internal unit conversion unit of this device can convert any user unit (User Unit: UU for short) into the drive's internal operating unit (Encoder Increment: Inc for short).

4.1 $608F_h$: Position Encoder Resolution

This object should indicate the configured encoder increment and the number of motor revolutions. The formula for calculating the position encoder resolution is given below:

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

The following table provides a description of the object:

Table4-1

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

The following table describes the sub-index:

Table4-2

Attribution	Description	
Sub-index	01_h $(0406)_h$	02_h $(0408)_h$
Description	Increment of encoder	Motor turns
Accessibility	rw	rw
Mappability	NO	NO
Data type	Unsigned32	Unsigned32
Default value	10000	1

4.2 6091_h : Transmission Ratios

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

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

The object description is specified in the table below :

Table4-3

Attribution	Description
Index	6091_h ($040E_h$, 0410_h)
Name	Gear ratio
Data structure	Array
Data type	Unsigned32

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

Table4-4

Attribution	Description	
Sub-index	01_h ($040E_h$)	02_h (0410_h)
Description	Motor shaft revolutions	Drive shaft revolutions
Accessibility	rw	rw
Mappability	No	No
Data type	Unsigned32	Unsigned32
Default value	1	1

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

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

  Motor Shaft Position Feedback = Drive Shaft Position Feedback × Gear Ratio

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

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

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

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

4.3 $607E_h$: Polarity

This object dictionary is used to set the position and velocity commands to be multiplied by 1 or -1. Position polarity is only used within the Contour Position and Cyclic Synchronized Position modes and has no effect on the Home Return mode. Velocity polarity is only used within the Contour Velocity Mode and Cyclic Synchronized Velocity Mode. The specific definitions are shown in the table below:

Table4-5

Bit-Value Meaning		
7	6	5~0
Position polarity	Velocity polarity	reserved (0)

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

Table 4-6

Attribution	**Description**
Index	$607E_h$ ($03F3_h$)
Name	Polarity
Data structure	Variable
Accessibility	rw
Data type	Unsigned8
Default value	0

4.4 Example

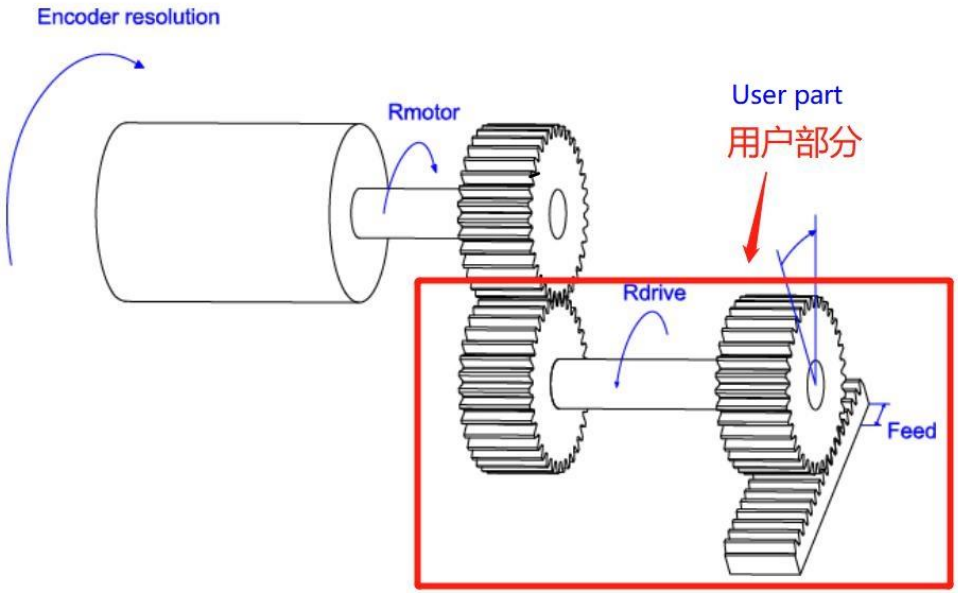


Figure 4-1

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

Position encoder resolution $608F_h$:

$$\text{Resolution of position encoder} = \frac{\text{increment of encoder(608Fh: 01h)}}{\text{motor turns (608Fh: 02h)}} = \frac{10000}{1} = 10000$$

Gear Ratio 6091_h :

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

For example, under the above conditions, a target position of 10,000 pulses (user units) is converted to internal encoder units: 10000 * 3 (transmission ratio) = 30000 pulses.

5 Operating Mode

5.1 Overview of Operation Model

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

Table5-1

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

Sub-index 01h to determine the mode:

Table5-2

Sub-Index	01_h ($00B1_h$)
Description	Control mode option
Accessibility	rw
PDO mapping	No
Data type	Unsigned8
Default value	0
Meaning of Value	0x00: CiA402 mode 0x01: NiMotion position mode 0x02: NiMotion speed mode 0x03: NiMotion Torque Mode 0x04: NiMotion open loop mode

After selecting the CiA402 mode, you need to set 6060_h ($03C2_h$) to select the specific motion mode (control mode selection, invalid operation status setting), and the objects are described as follows:

Table5-3

Sub-Index	6060_h ($03C2_h$)
Description	Mode Option
Accessibility	rw
PDO mapping	No
Data type	Unsigned8
Default value	0x01
Sub-Index	-
Meaning of Value	0x00: No definition 0x01: Profile Position Mode (PP) 0x02: Velocity Mode (VM)

Meaning of Value	0x03: Profile Velocity Mode (PV) 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)

When the servo is enabled and the motor shaft speed is 0, the working mode of the servo driver supports switching between PP and HM, as well as between CSP and HM.

5.2 Position Control Functions

5.2.1 Structure Chart

Target Following Error

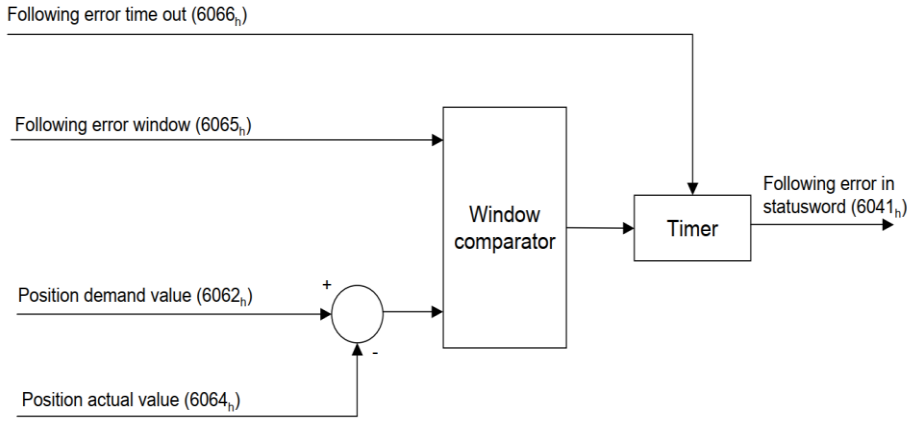


Figure5-1

Target Reached

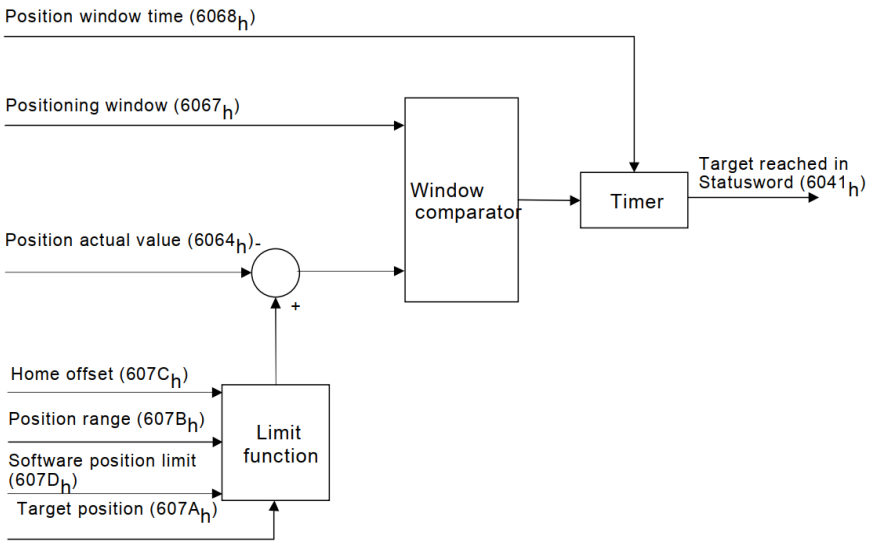


Figure5-2

5.2.2 Objects Related

Target following error

Table 5-4

Object Index	Description
6041_h (0381)$_h$	Status word
6062_h ($03C4$)$_h$	Current position demand value inside the drive (user unit)
6064_h ($03C8$)$_h$	Current position actual value feedback (user unit)
6065_h ($03CA$)$_h$	This object shall indicate the configured range of tolerated position values symmetrically to the position demand value. If the position actual value is out of the following error window, a following error occurs. A following error may occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed-loop coefficients.
6066_h ($03CC$)$_h$	Status word

Target Reached

Table5-5

Object Index	Description
6041_h (0381_h)	Status word
6064_h ($03C8_h$)	Current position value feedback (user unit)
6067_h ($03CD_h$)	The configured symmetrical range of accepted positions relative to the target position. If the actual value of the position encoder is within the position window, this target position shall be regarded as having been reached.
6068_h ($03CF_h$)	Position window time(unit：ms)
$607A_h$ ($03E7_h$)	Target position(user unit)
$607B_h$	Position range limit(user unit)
$607C_h$ ($03ED_h$)	Difference between the 0-position in the application and the machine home position (user unit)
$607D_h$ ($03EF_h$ and $03F1_h$)	Target position limit (user unit)

5.2.3 Functional Description

Target following error

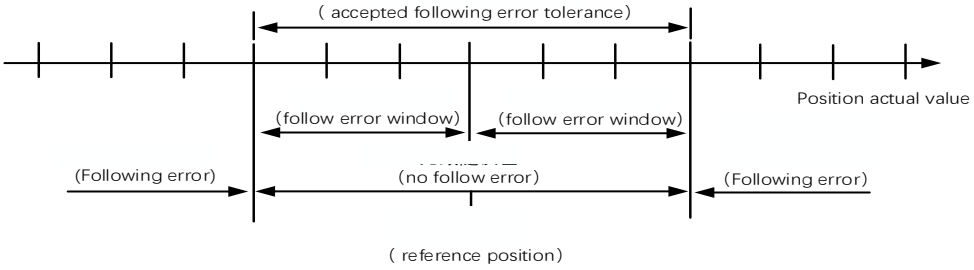


Figure5-3

The following error refers to the deviation between the reference position 6062_h ($03C4_h$) and the actual position 6064_h ($03C8_h$). If the following error value remains greater than the value in the following error window 6065_h ($03CA_h$) for the time set by 6066_h ($03CC_h$) (as shown in Fig. 6.2, the following error is outside the acceptable following error range), then bit13 (following error) of the status word 6041_h (0381_h) is set to 1, and the timing diagram is shown below:

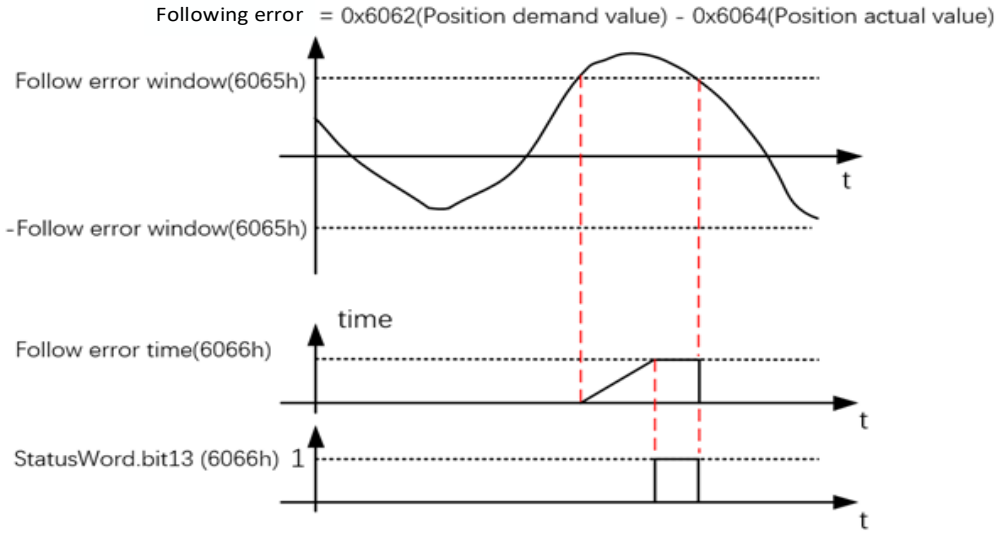


Figure5-4

Target Reached

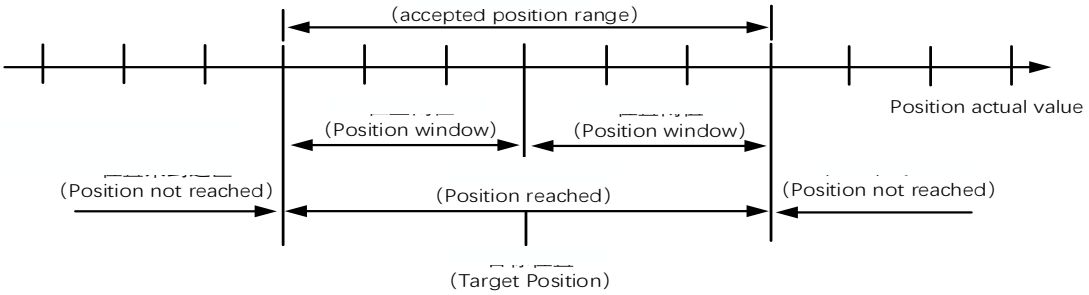


Figure 5-5 Schematic diagram of the acceptable range of error:

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 stabilizes within the acceptable position range (as shown above) and reaches the set time 6068$_h$ (03CF$_h$), then bit10 (target reached) of the status word will be set to 1, which means that the target position has been reached. The timing diagram is shown below:

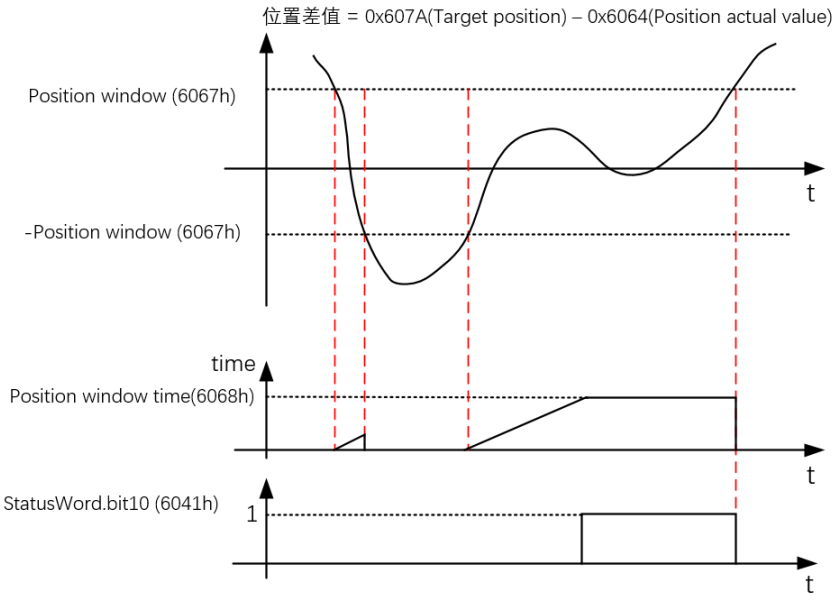


Figure5-6

5.3 Contour Position Mode (PP)

This mode is mainly used for point-to-point positioning applications. In this mode, the host computer gives the target position (absolute or relative), the speed of the position curve, acceleration and deceleration and deceleration, the internal trajectory generator will generate the target position curve instruction according to the setting, and the driver will complete the position control, speed control and torque control internally. The overall structure is shown below:

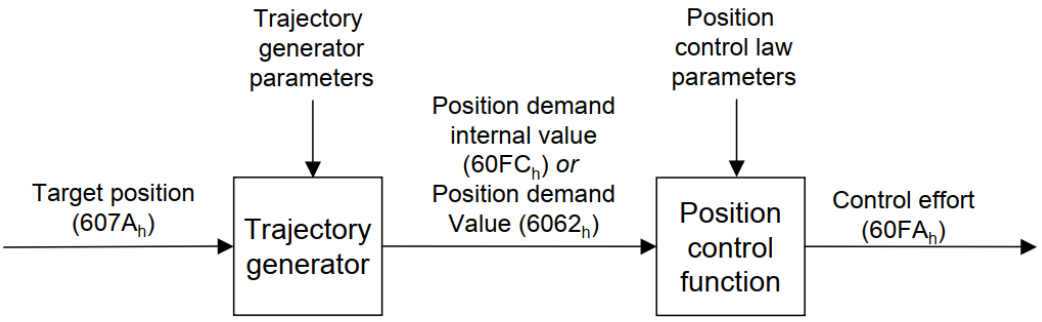


Figure5-7

5.3.1 Structure Plan

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

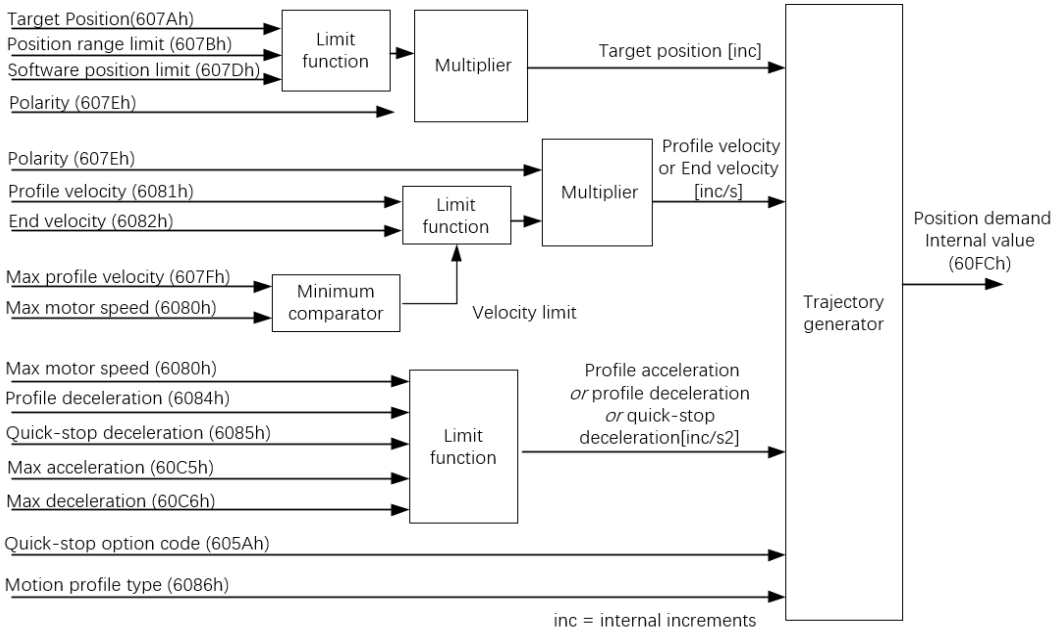


Figure5-8

5.3.2 Relevant Object

The following objects require attention in this mode:

Table5-6

Object Index	Description
$605D_h$ ($03C0_h$)	Pause mode selection
$605A_h$ ($03BF_h$)	Quick stop mode selection
6062_h ($03C4_h$)	Drive internal current target position command value (user units)
6063_h ($03C6_h$)	Feedback of the current absolute position of the motor (encoder unit)
6064_h ($03C8_h$)	Current position of the motor (user unit)
$607A_h$ ($03E7_h$)	Preset target position (user units)
$607D_h$ ($03EF_h$ and $03F1_h$)	Limitations on target location (user units)
$607E_h$ ($03F3_h$)	Direction of rotation (polarity), see "4.3 $607E_h$ ($03F3_h$): Polarity".
$607F_h$ ($03F4_h$)	Maximum profile speed during operation (user unit/s), which acts as a speed limiter.
6080_h ($03F6_h$)	Maximum motor speed (rpm)
6081_h ($03F8_h$)	The profile velocity (user units/s) of the uniform phase of the run of this segment displacement command, i.e., the velocity at which the end of the acceleration ramp is reached during positioning. The size is limited by $607F_h$ ($03F4_h$).
6082_h ($03FA_h$)	Contour End Velocity, the velocity (in user units/s) at which the target position is reached, the velocity at the end of the ramp, usually this object is set to zero so that the velocity is reduced to exactly 0 when the target position is reached. size is limited by $607F_h$ ($03F4_h$).
6083_h ($03FC_h$)	Contour acceleration during operation (user units/s^2), size limited by $60C5_h$ ($043B_h$).
6084_h ($03FE_h$)	Contour deceleration during the run (user units/s^2), size limited by $60C6_h$ ($043D_h$).
6085_h (0400_h)	Stopping deceleration rate when performing a "quick stop" (user unit/s^2)
6086_h (0402_h)	Selection of the type of planner curve, i.e. slope. If the value is "0", then there will be no limitation of shocks (plus acceleration), i.e. a trapezoidal curve; If the value is "3", the shock (plus acceleration) will be limited by the value of $60A4_h$: 01_h -02_h ($041E_h$ and 0420_h), i.e., S-curve, and only the indexes 01_h and 02_h are used in this device.
$60C5_h$ ($043B_h$)	Maximum acceleration limits (user units/s^2)
$60C6_h$ ($043D_h$)	Maximum deceleration limit (user units/s^2)
$60F2_h$ ($043F_h$)	Positioning Options
$60FC_h$ (0446_h)	Output of the trajectory generator, i.e. internally planned real-time position commands (encoder units)

The following figure illustrates the role of velocity, acceleration, and shock parameters on the running process:

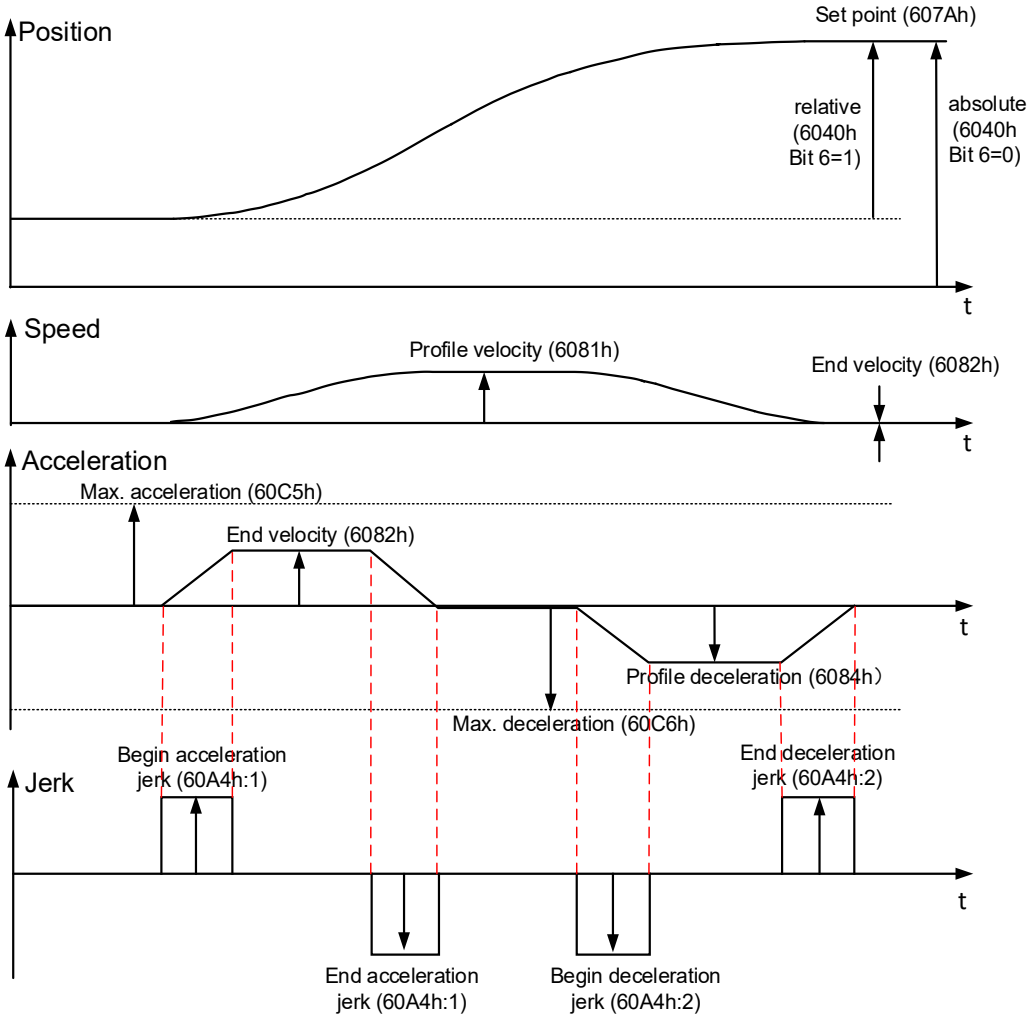


Figure5-9

5.3.3 Control Commands and Status Messages

start

To enable this mode, the value "0" must be set 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	(see 3.1.2)	abs/rel	Change set immediately	New set-point	(see 3.1.2)

MSB LSB

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

Table5-7

Bit 5	Bit 4	Description
0	0->1	Non-immediate updates will complete the running task before starting the next one.
1	0-> 1	Immediate update that will immediately execute the runtime task triggered by bit 4

Table5-8

Bit	Value	Description
6	0	The target position is an absolute position command
	1	The target position is a relative position instruction. The target position is a relative position based on the current position, and the reference position depends on bits 0 and 1 of $60F2_h$ ($043F_h$).
8	0	Positioning should be implemented or continued
	1	The motor will decelerate 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

The following bits in object 6041_h (0381_h) have a special function in profile position mode:

Table5-9

Bit	Value	Description
10	0	Non-pause state (bit8 = 0 for 6040_h (0380_h)): target not reached Pause state (bit8 = 1 for 6040_h (0380_h)): Motor deceleration
	1	Non-pause state (bit8 = 0 for 6040_h (0380_h)): target reached Pause state (bit8 = 1 for 6040_h (0380_h)): motor speed 0
12	0	Positioning of the previous positioning point is complete, waiting for the new set point

	1	The previous locator is still being processed and the new locator will overwrite the old one
13	0	No following error
	1	With following error

5.3.4 Functional Description

The control instruction timing is explained below in terms of immediate and non-immediate updates:

1） Control Instruction Timing - Update Now

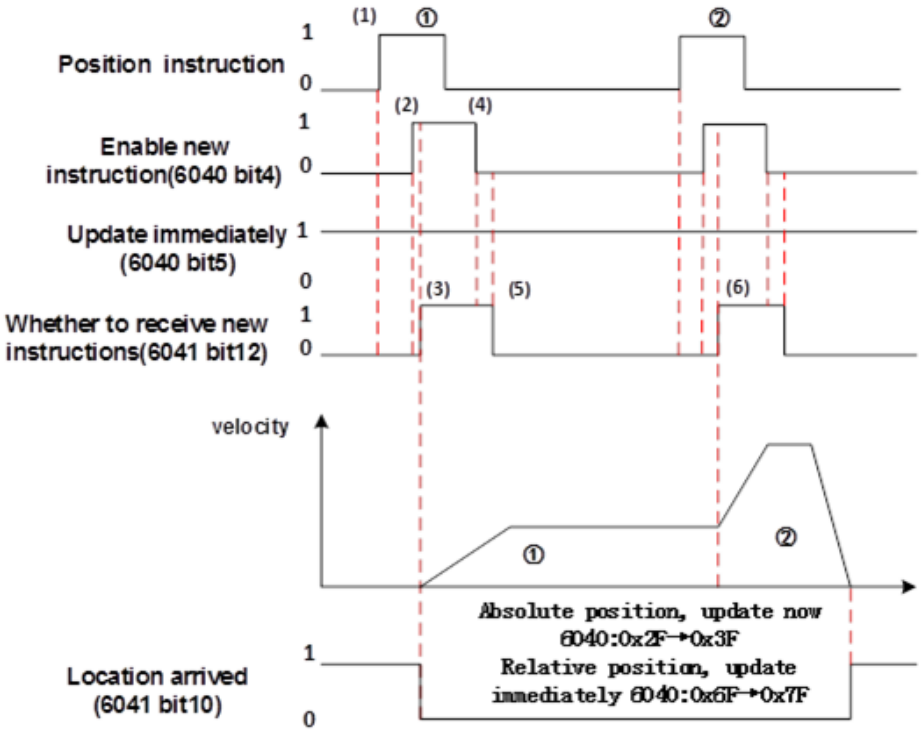


Figure5-10

The table below is a paraphrase of some of the labeling that was done during the course of the above diagram:

Table5-10

(1)	The upper computer updates the displacement commands (including target displacement $607A_h$ ($03E7_h$), acceleration time 6083_h ($03FC_h$), deceleration time 6084_h ($03FE_h$), profile speed 6081_h ($03F8_h$), etc.)
(2)	Set bit4 of 6040_h (0380_h) from 0 to 1 to indicate a new displacement command from the slave (bit5 of 6040_h (0380_h) is set to 1 for immediate updates)
(3)	After receiving the rising edge of bit4 of 6040_h (0380_h), the slave station makes a judgment on whether it can receive the new displacement instruction: if bit12 of 6041_h (0381_h) is 0 at this time, it indicates that the slave station can receive the new displacement instruction ①; after the slave station receives the new displacement instruction, bit12 of 6041_h (0381_h) is set from 0 to 1, which indicates that the new displacement instruction ① has been received and the slave station is currently in the state of not being able to continue to receive the new displacement instruction.

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

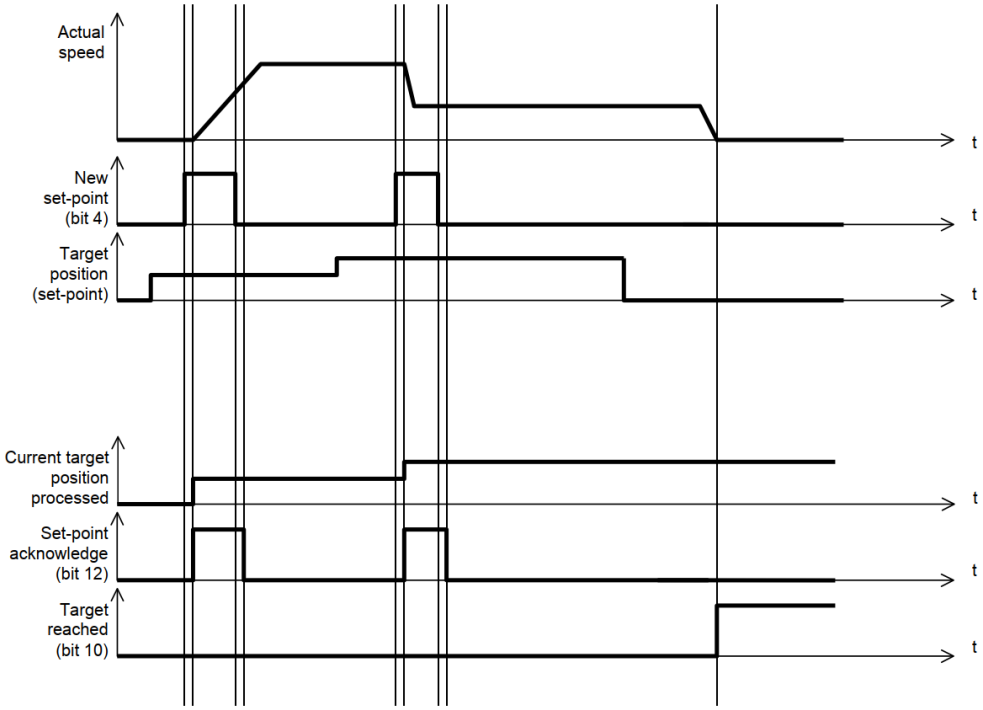


Figure5-11

2） Control Instruction Timing - Not Immediately Updated

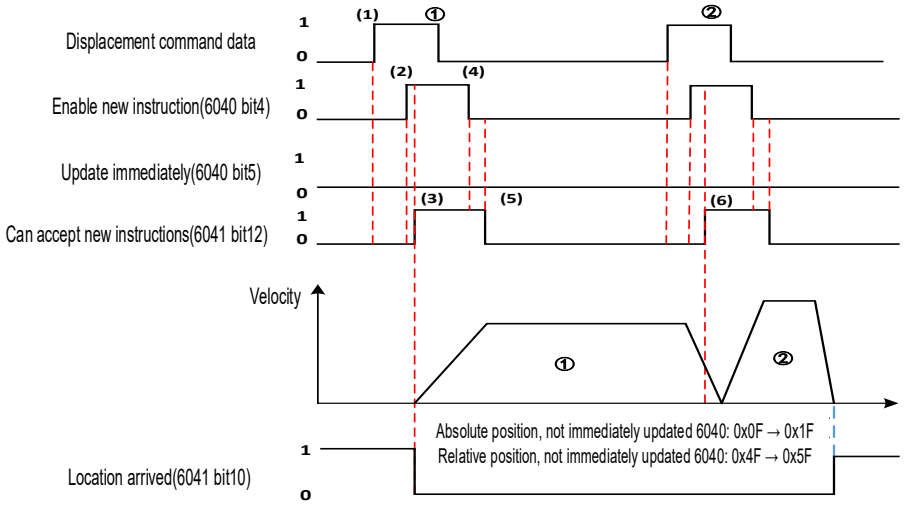


Figure5-12

In the case of non-immediate updating, the labels (1)-(5) in the figure are similar to the interpretation in the case of immediate updating, which can be seen by comparing position (6) in the figure. The difference is that: in the case of non-immediate updating, although the new target position (2) is accepted, it is necessary to arrive at position (1) first, decelerate and stop, and then travel to position (2) again.

In the case of non-immediate updates, the motor is designed with 5 caches, which means that there can be 5 target positions in the sequence at the same time, and the motor will travel to these positions in sequence. The motor will travel to these positions in sequence. If the caches are full, it will not be able to accept new positions until one of the caches is empty. These caches are cleared when the motor stops.

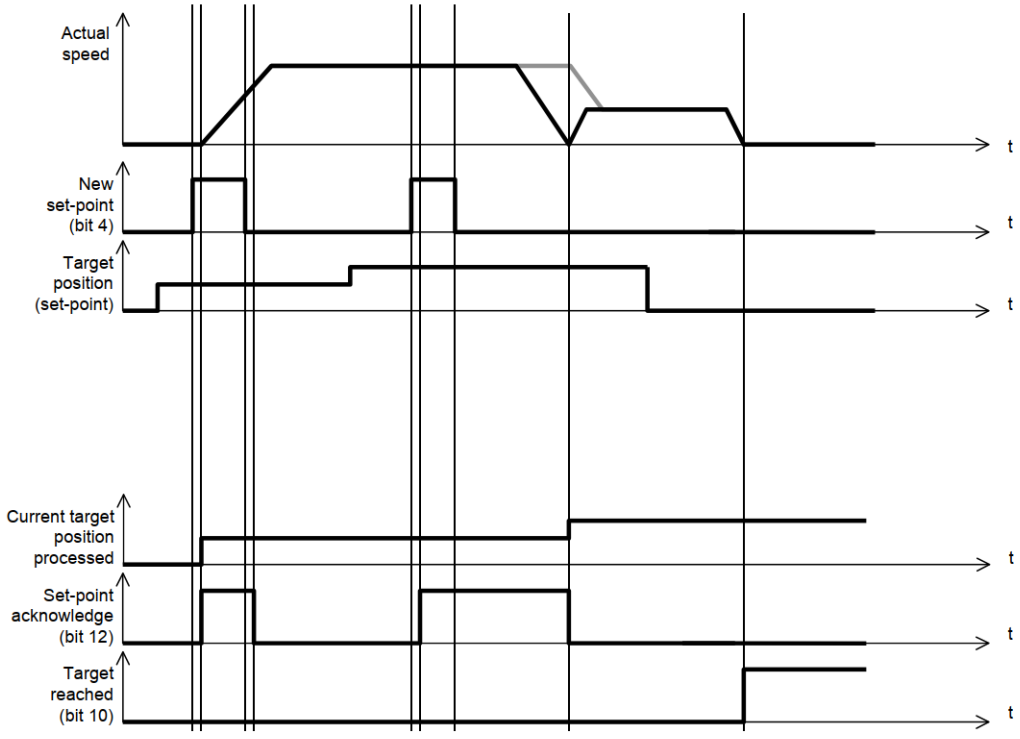


Figure5-13

5.3.5 Shock-constrained and Unconstrained Modes

In principle, models can be categorized as "shock-constrained" or "shock-unconstrained" according to shocks.

- **impingement limited model**

  Setting object 6086_h (0402_h) to "3" enables the execution of shock-restricted positioning runs. In this case, the shock-related items in object $60A4_h$: 01_h~ 04_h ($041E_h$ ~0424_h) take effect.

- **Impact unconstrained mode**

  In this mode, there is no shock limitation at any position of the characteristic curve. To run the "Shock unrestricted" ramp: Set the entry in object 6086_h (0402_h) to "0".

5.3.6 Configuration Example

- Configuration mode:
  2002:01_h ($00B1_h$) = 0, Run Mode 6060_h ($03C2_h$) = 0x01 to 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 Current Segment Displacement Command Uniform Running Speed 6081_h ($03F8_h$) (User Units/s);

  Sets the acceleration 6083_h ($03FC_h$) (user units/s^2) and deceleration 6083_h ($03FE_h$) (user units/s^2) of the displacement;

  Send: 01 10 03 E7 00 02 04 00 00 27 10 (set target location $607A_h$ ($03E7_h$) 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 (sets 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 to enable motor
  6040_h (0380_h) = 0x06→0x07→ 0x(n)F, motor enable:
  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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
- Make the motor run
  6040_h (0380_h) = 0x(n)F → 0x(n+1)F, motor is running:
  The different types of instructions are shown in the table below:

Table5-11

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

Send: 01 10 03 80 00 01 02 00 5F (set 6040_h (0380_h) to 0x5F, relative positional motion, not immediate update)

Send: 01 10 03 80 00 01 02 00 1F (set 6040_h (0380_h) to 0x1F, absolute positional motion, not immediate update)

Send: 01 10 03 80 00 01 02 00 6F (set 6040_h (0380_h) to 0x6F, relative positional motion, update immediately)

Send: 01 10 03 80 00 01 02 00 3F (set 6040_h (0380_h) to 0x3F, absolute position motion, update immediately)

- Monitoring Parameters:

  Actual position feedback: 6063_h ($03C6_h$) (encoder units), 6064_h ($03C8_h$) (user units)

5.4 Velocity Mode (VM)

5.4.1 Structure Plan

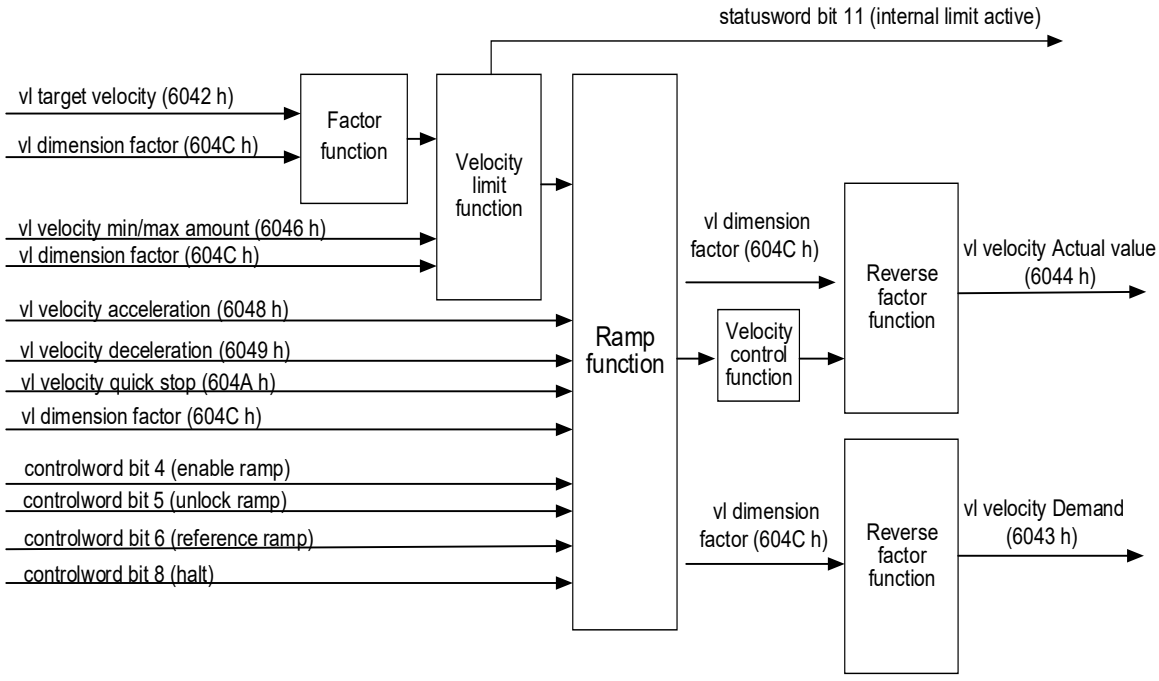


Figure5-14

5.4.2 Relevant Object

The following objects require attention in this mode:

Table5-12

Object Index	Description
6042_h (0382_h)	Target speed for VM mode (default unit: rpm, related to $604C_h$ (0394_h and 0396_h))
6043_h (0383_h)	Target speed command in effect for 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 VM mode speed units, default speed units are rpm (revolutions per minute). $604C_h:01_h$ (0394_h) contains the numerator (multiplier) used to calculate speed and $604C_h:02_h$ (0396_h) contains the denominator (divisor). Speed units: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: 2rpm. If either $604C_h:01_h$ (0394_h) and $604C_h:02_h$ (0396_h) is 0, the speed unit is rpm.
$6048_h:01_h$ (0389_h) and $6048_h:02_h$ ($038B_h$)	VM mode acceleration. $6048_h:01_h$ (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 $6049_h:02_h$ ($038E_h$)	Deceleration of the VM mode. $6049_h:01_h$ ($038C_h$) contains the speed change in rpm, $6049_h:02_h$ ($038E_h$) contains the corresponding time in s (seconds) The two together calculate the acceleration 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)	Deceleration speed at fast stop for VM mode. $604A_h:01_h$ ($038F_h$) contains the speed change (in rpm) and $604A_h:02_h$ (0391_h) contains the corresponding time (in s (seconds)). The two together calculate the acceleration 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)	Limits for speed in VM mode (default unit: rpm, related to $604C_h:01_h$ (0394_h) and $604C_h:02_h$ (0396_h))

Object Index	Description
	It is described as follows: 6046_h : 01_h (0385_h) Sets the minimum speed. If the target speed (6042_h (0382_h)) is lower than the minimum speed, its value will be set to the minimum speed 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, its value will be set to the maximum speed 6046_h : 02_h (0387_h).

The figure below illustrates the role of acceleration, deceleration on the running process:

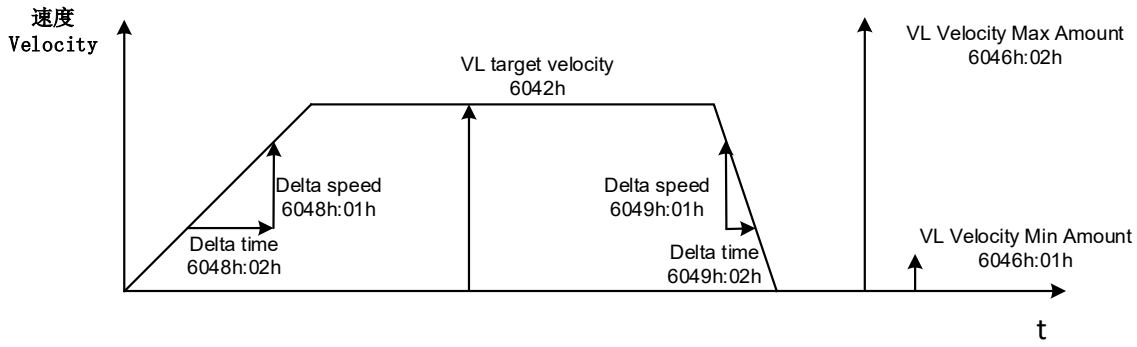


Figure5-15

5.4.3 Control Commands and Status Messages

start

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

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 speed mode, the following bits in object **6040**$_h$ (0380_h) have a special function

Bit	(be) worth	Description
2	0	When a quick stop is triggered, the motor applies the quick brake according to the quick stop deceleration speed set in object $604A_h$. The controller then switches to "Switch on disabled".
	1	no movement

Bit	(be) worth	Description
4	0	No acceleration or deceleration process, output changes immediately to the given speed
	1	Speed regulation according to acceleration and deceleration settings
5	0	No longer following the output of the planner for speed regulation, the speed output value should be locked to the current speed value
	1	Speed regulation according to the speed output of the planner
6	0	Velocity profile planner input is 0
	1	The speed profile planner input is a given target speed
8	0	no movement
	1	Motor Suspension

Table5-13

status word

15 14	13 12	11	10	9 0
(see 3.1.3)	reserved	(see 3.1.3)	reserved	(see 3.1.3)
MSB				LSB

In speed mode, the bits in object **6041**$_h$ (0381$_h$) have no special function; see section 4.1.3 status word for the definition of the general-purpose bits.

5.4.4 Functional Description

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

5.4.5 Configuration Example

- Configuration mode:
  Write 2002$_h$: 01$_h$ (00B1$_h$) = 0, 6060$_h$ (03C2$_h$) = 0x02, configured for 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 speed mode)
- Parameter Configuration:
  Write target speed: 6042$_h$ (0382$_h$) = 1000;
  Send: 01 10 03 82 00 01 02 03 E8 (set target speed to 1000 rpm)
  Write acceleration and deceleration:
  6048$_h$: 01$_h$ (0389$_h$) = 500, 6048$_h$: 02$_h$ (038B$_h$) = 1; and
  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 the word control:

  6040_h (0380_h) = 0x06→0x07→ 0x7F, motor enable:

  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 to disable motor)

  Send: 01 10 03 80 00 01 02 00 7F (set 6040_h (0380_h) to 0x7F to enable motor)

- Parameter monitoring:

  Current actual speed $606C_h$ ($03D5_h$) (in rpm)

5.5 Profile Velocity Mode (PV)

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

5.5.1 Structure Plan

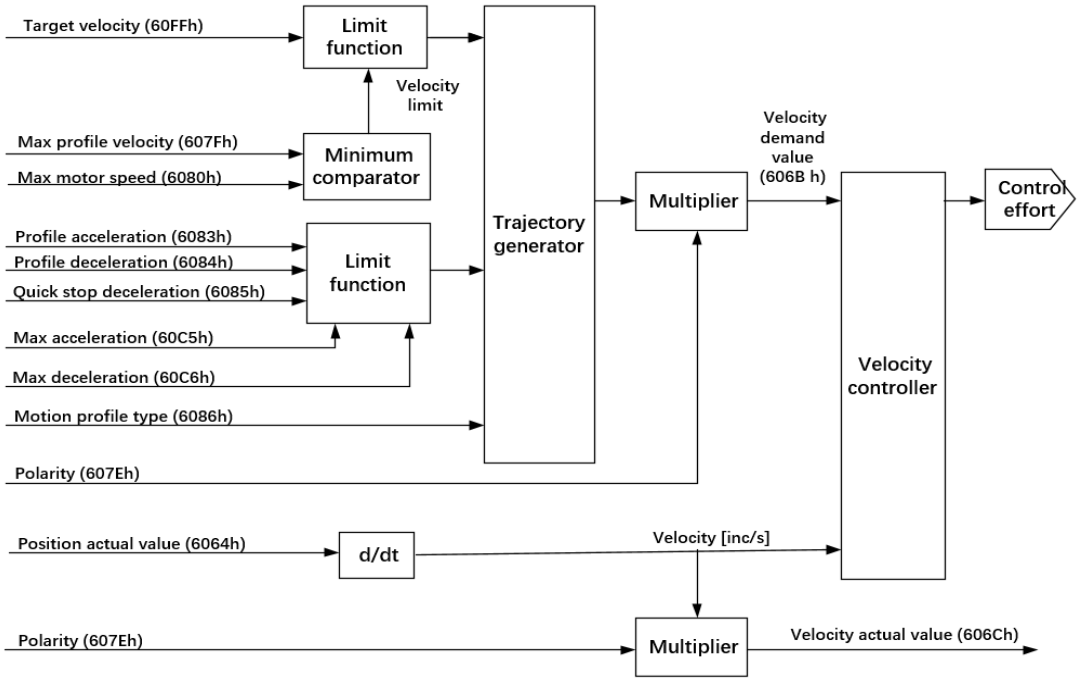


Figure5-16

5.5.2 Related Objects

The following objects require attention in this mode:

Table5-14

Object Index	Description
6063_h ($03C6_h$)	Feedback of the current absolute position of the motor (encoder unit)
6064_h ($03C8_h$)	Feedback on the current absolute user position of the motor (user units)
$606B_h$ ($03D3_h$)	Speed command value validated within the controller
$606C_h$ ($03D5_h$)	Current actual speed feedback value (rpm)
$607E_h$ ($03F3_h$)	Direction of rotation (polarity), see "4.3 $607E_h$: Polarity".
$607F_h$ ($03F4_h$)	Maximum profile speed during operation (user unit/s), which acts as a speed limiter.
6080_h ($03F6_h$)	Maximum speed of motor (rpm)
6083_h ($03FC_h$)	Profile acceleration during operation (user unit/s^2), size limited by $60C5_h$.
6084_h ($03FE_h$)	Contour deceleration during the run (user unit/s^2), size limited by $60C6_h$.
6085_h (0400_h)	Stopping deceleration rate when performing a "quick stop" (user unit/s^2)
6086_h (0402_h)	Select the type of curve, i.e. slope. If the value is "0", the shock (plus acceleration) is not limited, i.e., the trapezoidal curve; If the value is "3", the shock (plus acceleration) will be limited by the value in $60A4_h$: 01_h -02_h , i.e. the S-curve, only the 01_h and 02_h indexes are used in this device.
$60C5_h$ ($043B_h$)	Maximum acceleration limits (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, the value "0" must be set in object 2002_h : 01_h (00B1h) and "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 speed mode, the following bits in object 6040_h (0380_h) have a special function

Table5-15

Bit	Value	Description
8	0	Motor continues to run
	1	pause (media player)

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 speed mode, the following bits in object 6041_h (0381_h) have a special function:

Table5-16

Bit	Value	Description
10	0	Non-suspended state (bit8 of 6040_h (0380_h) = 0): target speed not reached Pause state (bit8 = 1 for 6040_h (0380_h)): motor deceleration
	1	Non-suspended state (bit8 of 6040_h (0380_h) = 0): target speed reached Pause state (bit8 = 1 for 6040_h (0380_h)): motor speed 0
12	0	Velocity is not zero.
	1	Speed is zero.

5.5.4 Functional Description

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

5.5.5 Configuration Example

- Configuration Mode
  Write 2002_h : 01_h ($00B1_h$) = 0, run mode 6060_h ($03C2_h$) = 0x03 to operate in contour speed 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 profile speed mode)

- Configuration parameters:
  Write target speed: $60FF_h$ (0448_h) = 50000;
  Send: 01 10 04 48 00 02 04 00 00 C3 50 (set target speed to 300 rpm)
  Send: 01 10 04 48 00 02 04 FF FF 3C B0 (set target speed to -300 rpm)
  Write Profile Acceleration: 6083_h ($03FC_h$) = 40000.
  Send: 01 10 03 FC 00 02 04 00 00 9C 40 (sets acceleration 6083_h ($03FC_h$) to 40000)
  Write contour deceleration: 6084_h ($03FE_h$) = 40000.
  Send: 01 10 03 FE 00 02 04 00 00 9C 40 (set deceleration 6084_h ($03FE_h$) to 40000)

- Write the word control:
  6040_h = 0x06→0x07→0x0F, motor 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 to disable motor)

Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable 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 6071h (03DBh), the torque ramp constant 6087h (0403h) to the drive, the torque adjustment is performed internally, and when the speed reaches the limiting value it will enter the speed control phase.

5.6.1 Structure Plan

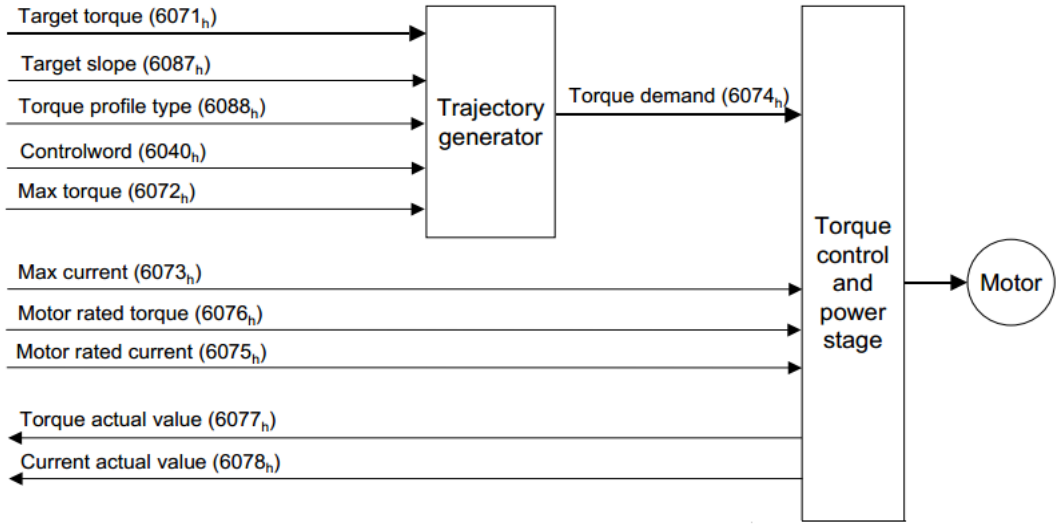


Figure5-17 Control block diagram

5.6.2 Relevant Object

The following objects require attention in this mode:

Table5-17

Object Index	Description
605Dh (03C0)h	Pause mode selection
605Ah (03BF)h	Quick stop mode selection
6071h (03DB)h	Setting target torque (unit: 0.1%)
6072h (03DC)h	Maximum torque on the entire ramp (acceleration, conservation of torque, braking) (unit: 0.1%)
2007h : 10h (016D) h	Torque control forward speed limit value (rpm)
2007h : 11h (016E)h	Torque control reverse speed limit value (rpm)
6087h (0403)h	Maximum change in torque per second (unit: 0.1%)
6088h (0405)h	Torque ramp type (0-slope, 2-none)
6077h (03E3)h	Actual value of torque (unit: 0.1%)

5.6.3 Control Commands and Status Messages

start

To enable this mode, the value "0" must be set in object 2002_h : 01_h ($00B1_h$) and "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 a special function

Table5-18

Bit	Value	Description
8	0	Motor continues to run
	1	pause (media player)

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 a special function:

Table5-19

Bit	Value	Description
10	0	Non-suspended state (bit8 = 0 for 6040_h (0380_h)): target torque not reached In the pause state (bit8 = 1 for 6040_h (0380_h)): Motor deceleration
	1	Non-suspended state (bit8 = 0 for 6040_h (0380_h)): target torque reached Pause state (bit8 = 1 for 6040_h (0380_h)): motor speed 0

5.6.4 Functional Description

Speed Limit Function

Running an integrated low-voltage servo motor in contour torque mode automatically switches to speed mode control and controls the current speed to be within the speed limit 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 ($016E_h$)). When the given target torque is detected to be less than the average torque at the current speed, the motor exits speed control and reverts to torque mode control.

5.6.5 Configuration Example

- Configuration Mode
  Write 2002_h : 01_h ($00B1_h$) = 0, run mode 6060_h ($03C2_h$) = 0x04 to operate 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 profile torque mode)
- Configuration parameters:
  Write target torque: 6071_h ($03DB_h$) = 500 (unit: 0.1%);
  Send: 01 10 03 DB 00 01 02 01 F4 (set target torque 6071_h ($03DB_h$) to 50%)
  Write Torque Ramp Type 6088_h (0405_h)=2;
  Write torque ramp 6087_h (0403_h)=10.
  Send: 01 10 04 05 00 01 02 00 02 (sets ramp type 6088_h (0405_h) to no ramp)
  Send: 01 10 04 03 00 02 04 00 00 00 0A (set torque ramp 6087_h (0403_h) to 10)
- Write the word control:
  6040_h (0380_h) = 0x06→0x07→0x0F, motor 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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
- Parameter monitoring:
  Actual torque 6077_h ($03E3_h$) (unit 0.1%)

5.7 Homing Mode (HM)

Homing mode is used to move from the current position to the home position of the device. During the movement, maximum acceleration, minimum deceleration, maximum velocity, minimum velocity, etc. are taken into account.

5.7.1 Structure Plan

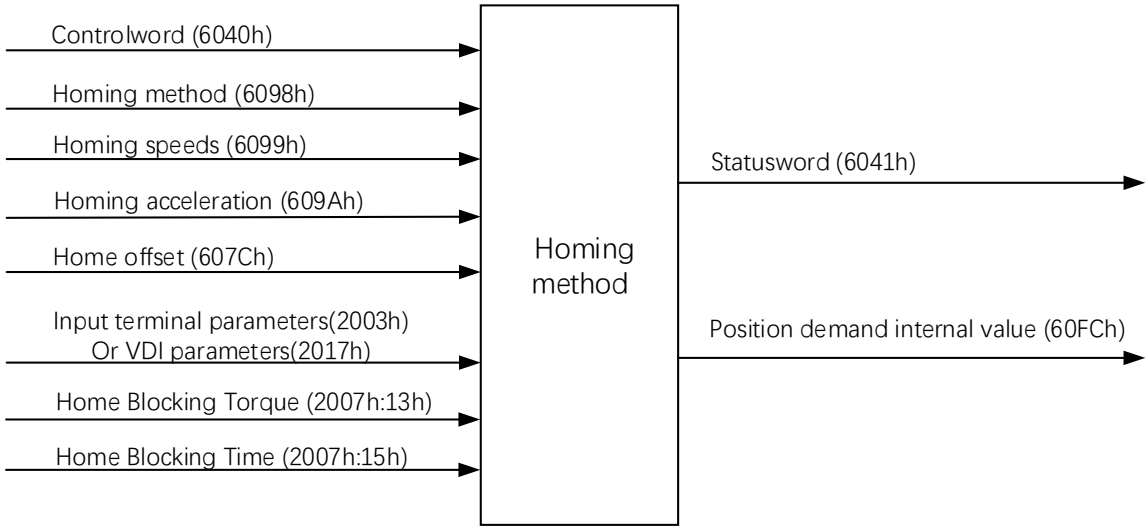


Figure5-18

5.7.2 Relevant Object

The following objects require attention in this mode:

Table5-20

Object Index	Description
607C$_h$ (03ED)$_h$	Specific description: Set the physical position of the mechanical zero point deviation from the motor home position in the home return mode; takes effect when bit15=1 of status word 6041$_h$ (0381$_h$) at the completion of the home return operation
6098$_h$ (0416)$_h$	Home return method (see "6.7.4 Functional description")
6099$_h$:01$_h$ (0417)$_h$ 6099$_h$:02$_h$ (0419)$_h$	Running speed (user unit/s) Sub-index: 6099$_h$: 01$_h$ (0417$_h$): speed to find the switch; 6099$_h$: 02$_h$ (0419$_h$): speed to find the origin
609A$_h$ (041B)$_h$	Acceleration of the origin return run (user units/s^2)
60FC$_h$ (0446$_h$)	Output of the trajectory generator, i.e. internally planned real-time position commands (encoder units)
6063$_h$ (03C6)$_h$	Feedback of the current absolute position of the motor (encoder unit)
2003$_h$ (00D5$_h$ ~00E6)$_h$	Functional configuration of physical input terminals (see "6.1.1 Digital Input" for details)

Object Index	Description
	Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.
2017_h (0326_h ~0345_h)	Configuration of the virtual terminal input function (see: "6.1.1 Digital input")
2007_h :13_h (0170_h)	Detecting torque while searching for home position in blocked rotation
2007_h :15_h (0172_h)	Detection time when blocking rotation to find the home position

Examples of the meaning of speed

The following graph shows the change in velocity when using the origin regression method 18:

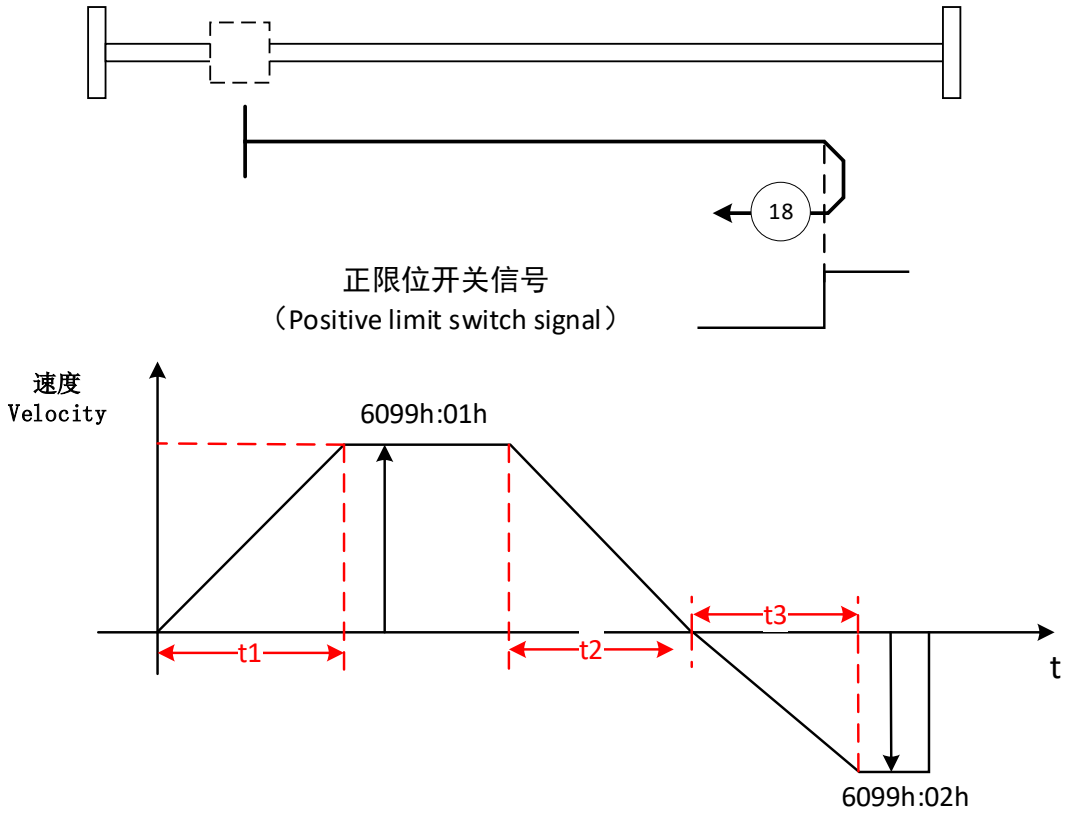


Figure 5-19

Among them:t1=$\frac{6099_h:\ 01_h(0417_h)}{609A_h(041B_h)}$;t2=$\frac{6099_h:\ 01_h(0417_h)}{609A_h(041B_h)}$;t3 =$\frac{6099_h:\ 02_h(0419_h)}{609A_h(041B_h)}$

5.7.3 Control Commands and Status Messages

start

To enable this mode, the value "0" must be set in object 2002_h : 01_h ($00B1_h$) and "6" in object 6060_h ($03C2_h$), and the multifunction terminal must be configured with the function of home switch or limit switch. For details of the multifunction terminal configuration, see section 7.1.

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 home return mode, the following bits in object 6040_h (0380_h) have a special function

Table5-21

Bit	Value	Description
4	0	Home return mode not activated
	1	Initiation or continuation of home return mode
8	0	The motor decides whether to start the home return or not by setting bit4.
	1	The motor is suspended at 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

The following bits in object 6041_h (0381_h) have a special function in the home return mode:

Table5-22

Bit	Value	Description
10	0	Target position not reached
	1	Target position reached
12	0	Return of origin incomplete
	1	Return of origin complete

5.7.4 Functional Description

5.7.4.1 Finding the home switch mode

The home return method using a switch currently supports 17~30 in CiA402. For home return to zero, we often need to start running from zero, and generally speaking, we need to run to the zero position command after home return to zero. For the convenience of users, this series of integrated motors will automatically return to the zero position after the home return.

Configuration of the input terminal DI function number associated with the HM mode:

Table5-23

Function Number	Functional Definition
14	Positive Limit Switches
15	Negative Limit Switches
31	Home switch

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

Table5-24

Homing Method	Switch Related
17	Negative Limit Switches
18	Positive Limit Switches
19	Home switch
20	Home switch
21	Home switch
22	Home switch
23	Home Switch, Positive Limit Switch
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
35	-

- Homing Method 17:

Mechanical Home: Negative Limit Switch

When the negative limit switch is invalid , start back to zero at high speed in reverse, decelerate, reverse, and run at low speed in forward direction when the negative limit switch is valid (along the change), and stop when the negative limit switch is invalid (along the change).

When the negative limit switch is valid, the direct positive low speed starts back to zero and stops when it encounters the negative limit switch as invalid (along the change).

The motion trajectory is shown below:

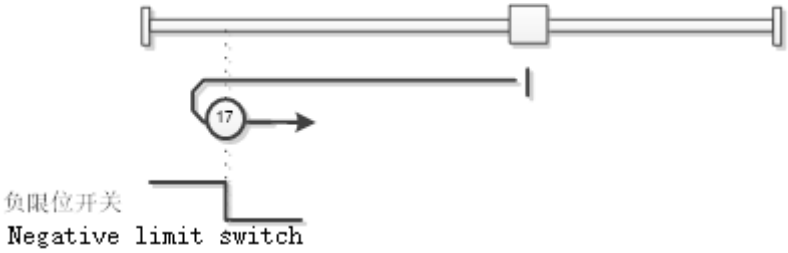


Figure5-20

- Homing Method 18:

Mechanical Home: Positive Limit Switch

When the positive limit switch is invalid, start back to zero at high speed in the positive direction, decelerate, reverse, reverse low speed when the positive limit switch is valid (along the change), and stop when the positive limit switch is invalid (along the change).

When the positive limit switch is valid, it starts back to zero at low speed in direct reverse, and stops when it encounters a positive limit switch that is invalid (along the change).

The motion trajectory is shown below:

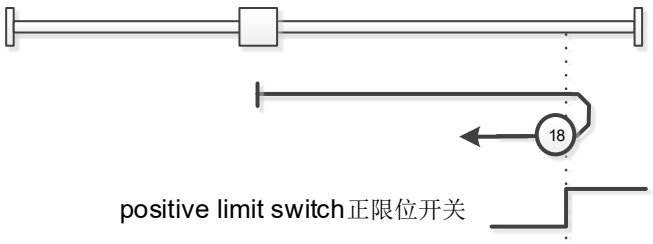


Figure5-21

- Homing Method 19 and 20:

Mechanical Home: Home Switch

Homing Method 19:

When the home switch is invalid , start back to zero with forward high speed, encounter the home switch is valid (along the change), decelerate, reverse, reverse low speed, encounter the home switch is invalid (along the change) to stop; when the home switch is valid, start back to zero with direct reverse low speed, encounter the home switch is invalid (along the change) to stop.

Homing Method 20:

When the home switch is invalid, start back to zero at low speed directly in the forward direction, and stop when the home switch is valid (along the change); when the home switch is valid, start back to zero at high speed in the reverse direction, and stop when the home switch is invalid (along the change), decelerate, reverse, and run at low speed in the forward direction, and stop when the home switch is valid (along the change).

The motion trajectory is shown below:

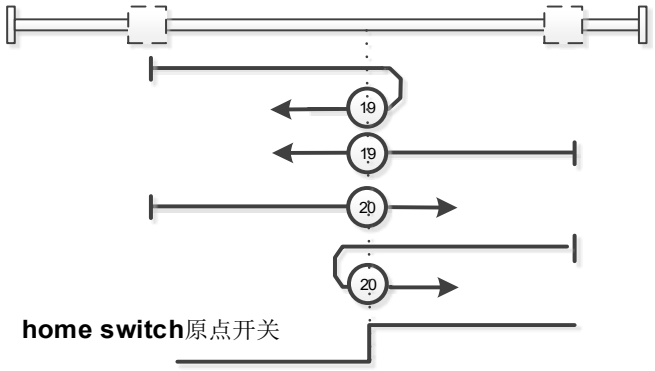


Figure5-22

- Homing Method 21 and 22:

Mechanical Home: Home Switch

Homing Method 21:

When the home switch is invalid, start back to zero at high speed in reverse, decelerate, reverse, and run at low speed in forward direction when the home switch is invalid (along the change), and stop when the home switch is invalid (along the change); When the home switch is valid, start back to zero at low speed in forward direction directly, and stop when the home switch is invalid (along the change).

Homing Method 22:

When the home switch is invalid, start back to zero at low speed in direct reverse, and stop when the home switch is valid (along the change); when the home switch is valid, start back to zero at high speed in forward direction, and stop when the home switch is invalid (along the change), decelerate, reverse, and run at low speed in reverse direction, and stop when the home switch is valid (along the change).

The motion trajectory is shown below:

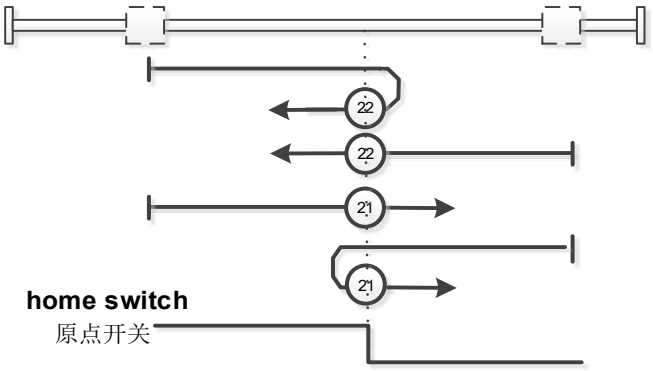


Figure5-23

- Homing Method 23-26:

Mechanical Home: Home Switch

These are actually motors moving in one direction first to scan the home switch. The only time the home switch is active is when it is a shorter search trajectory.

Homing Method 23:

a. When the home switch is invalid, start back to zero at high speed in the positive direction without encountering the positive limit switch, decelerate, reverse, reverse low speed when the home switch is valid (along the change), and stop when the home switch is invalid (along the change);
b. When the home switch is invalid, start back to zero with positive high speed, encounter positive limit switch, reverse, reverse high speed, encounter home switch valid (along the change), decelerate, continue to reverse low speed, encounter home switch invalid (along the change) stop;
c. When the home switch is active, it starts back to zero at low speed in direct reverse and stops when it encounters the home switch is not active (along the change).

Homing Method 24 :

a. When the home switch is invalid, start back to zero at high speed in the forward direction without encountering a positive limit switch, when the home switch is valid (along the change), decelerate, reverse, run at low speed in the reverse direction, when the home switch is invalid (along the change), reverse, run at low speed in the forward direction, and stop when the home switch is valid (along the change);
b. When the home switch is invalid, start back to zero with positive high speed, encounter positive limit switch, reverse, reverse high speed, encounter home switch valid (along the change), decelerate, reverse low speed, encounter home switch invalid (along the change), reverse, positive low speed, encounter home switch valid (along the change) and stop;
c. When the home switch is valid, start back to zero at low speed directly in reverse, reverse when the home switch is invalid (along the change), run at low speed in forward direction, and stop when the home switch is valid (along the change).

Homing Method 25:

a. When the home switch is invalid, start back to zero at high speed in the forward direction without encountering a positive limit switch, decelerate and run at low speed in the forward

direction when encountering a home switch that is valid (along the change), reverse and run at low speed in the reverse direction when encountering a home switch that is invalid (along the change), and stop when encountering a home switch that is valid (along the change);

b. When the home switch is invalid, start back to zero with positive high speed, encounter positive limit switch, reverse, reverse high speed, encounter home switch valid (along the change), decelerate, positive low speed, encounter home switch invalid (along the change), reverse, reverse low speed, encounter home switch valid (along the change) stop;
c. When the home switch is valid, direct forward low speed start back to zero, encounter the home switch is invalid (along the change), reverse, reverse low speed, encounter the home switch is valid (along the change) stop.

Homing Method 26:

a. When the home switch is invalid, start back to zero at a positive high speed without encountering a positive limit switch, decelerate and run at a positive low speed when the home switch is valid (along the change), and stop when the home switch is invalid (along the change);
b. When the home switch is invalid, start back to zero with positive high speed, encounter positive limit switch, reverse, reverse high speed, encounter home switch valid (along the change), decelerate, positive low speed, encounter home switch invalid (along the change) stop;
c. When the home switch is active, direct forward low speed starts back to zero and stops when it encounters a home switch that is not active (along the change).

The motion trajectory is shown below:

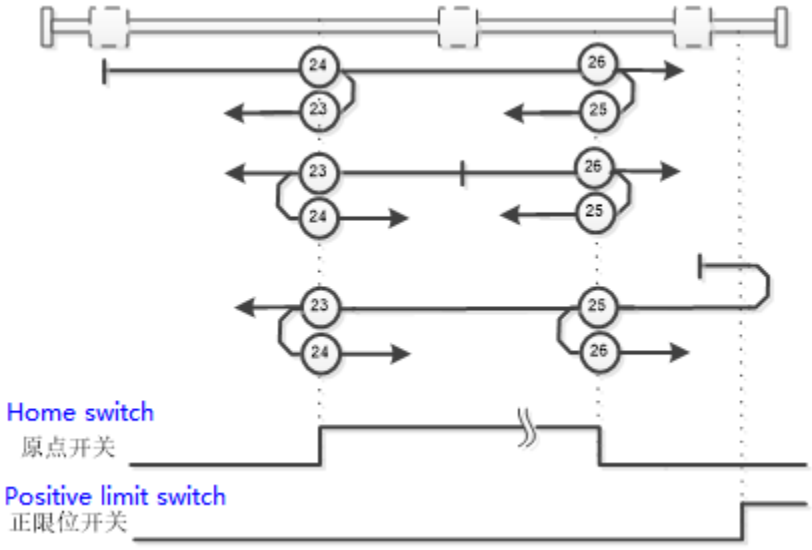


Figure5-24

- Homing Method 27-30:

Mechanical Home: Home Switch

These are actually motors moving in one direction first to scan the home switch. The only time the home switch is active is when it is a shorter search trajectory.

Homing Method 27:

a. When the home switch is invalid, start back to zero at high speed in reverse without encountering a negative limit switch, decelerate, reverse, and run at low speed in forward direction when the home switch is valid (along the change), and stop when the home switch is invalid (along the change);
b. When the home switch is invalid, start back to zero at high speed in reverse, encounter the negative limit switch, reverse, run at high speed in the positive direction, encounter the home switch valid (along the change), decelerate, continue to run at low speed in the positive direction, and stop when encountering the home switch invalid (along the change);
c. When the home switch is active, direct forward low speed starts back to zero and stops when it encounters a home switch that is not active (along the change).

Homing Method 28:

a. When the home switch is invalid, start back to zero at high speed in reverse without encountering a negative limit switch, when the home switch is valid (along the change), decelerate, reverse, and run at low speed in the positive direction, when the home switch is invalid (along the change), reverse, and run at low speed in the reverse direction, and stop when the home switch is valid (along the change);
b. When the home switch is invalid, start back to zero with reverse high speed, encounter negative limit switch, reverse, positive high speed operation, encounter home switch valid (along the change), decelerate, positive low speed operation, encounter home switch invalid (along the change), reverse, reverse low speed operation, encounter home switch valid (along the change) stop;
c. When the home switch is valid, direct forward low speed start back to zero, encounter the home switch is invalid (along the change), reverse, reverse low speed, encounter the home switch is valid (along the change) stop.

Homing Method 29:

a. When the home switch is invalid, start back to zero at high speed in reverse without encountering a negative limit switch, when the home switch is valid (along the change), decelerate and run at low speed in reverse, when the home switch is invalid (along the change), reverse and run at low speed in positive direction, and stop when the home switch is valid (along the change);
b. When the home switch is invalid, start back to zero with reverse high speed, encounter negative limit switch, reverse, positive high speed operation, encounter home switch valid (along the change), decelerate, reverse low speed operation, encounter home switch invalid (along the change), reverse, positive low speed operation, encounter home switch valid (along the change) stop;
c. When the home switch is valid, start back to zero at low speed directly in reverse, reverse when the home switch is invalid (along the change), run at low speed in forward direction, and stop when the home switch is valid (along the change).

Homing Method 30:

a. When the home switch is invalid, start back to zero at high speed in reverse without encountering the negative limit switch, decelerate when the home switch is valid (along the change), run at low speed in reverse, and stop when the home switch is invalid (along the change);

b. When the home switch is invalid, start back to zero with reverse high speed, encounter negative limit switch, reverse, positive high speed operation, encounter home switch valid (along the change), decelerate, reverse low speed operation, encounter home switch invalid (along the change) stop;

a. When the home switch is active, it starts back to zero at low speed in direct reverse and stops when it encounters the home switch is not active (along the change).

The motion trajectory is shown below:

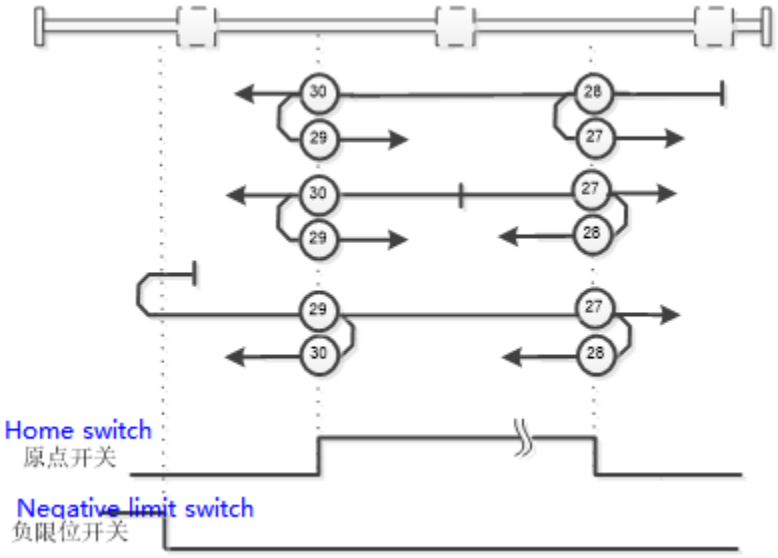


Figure5-25

- Homing Method 35:

After resetting the trigger origin to zero (with the control word 6040_h(0380_h) changing from 0x0F to 0x1F), take the current position as the mechanical origin.

5.7.4.2 Blocking the turn to find the way of origin

The blocking home return method is currently only supported in closed-loop control mode. The blocking home return method is similar to the previous method of using a switch to find the home position, but the difference is that the blocking detection torque value is used instead of a limit switch.

The way of its origin return and the direction of its operation are shown in the table below:

Table5-25

Homing Method	Running Direction
37	Running at 6099_h :02_h (0419_h) speed forward
38	Running at 6099_h :02_h (0419_h) speed reverse

In addition this method requires the following two parameters to be set:

1、Detection torque during home search in stalled rotation 2007_h :13_h (0170_h)

This blocking torque threshold should be greater than the actual operating torque during the origin finding process and should generally be less than the maximum torque to avoid triggering the blocking fault protection.

2、Detection time when blocking to find the home position 2007_h :15_h (0172_h)

This time is the time for the motor blocking to find the home position process to maintain the blocking.

The process of the specific plug-and-go home return method is schematically shown below:

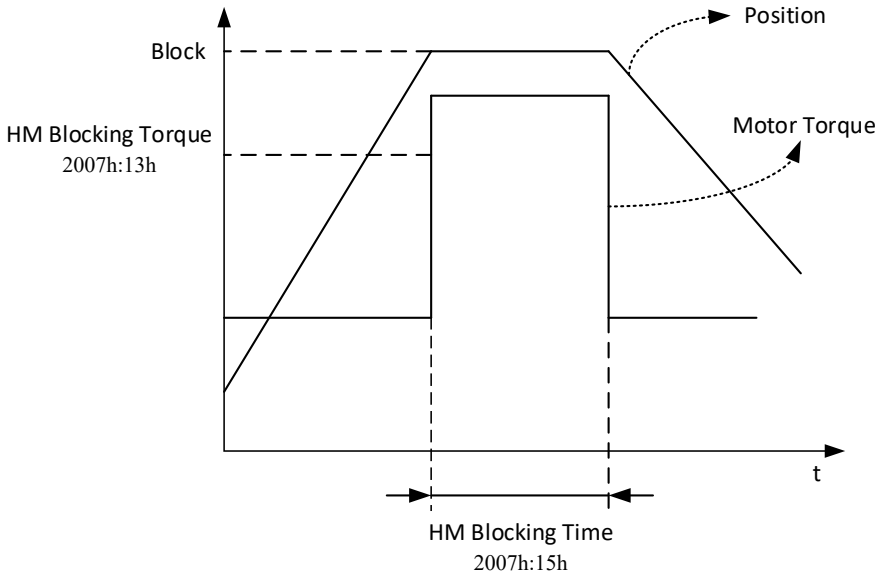


Figure5-26

5.7.5 Configuration Example

1. Negative Limit Switch Finding Home Mode

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

- Setup Mode
  Write 2002_h : 01_h ($00B1_h$) = 0, run mode 6060_h ($03C2_h$) = 0x06 to make it work in home return 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 return mode)

- Setting the origin return method
  6098_h (0416_h) = 17.
  Send: 01 10 04 16 00 01 02 00 11 (set home return mode to 17)

- Write find limit switch speed and find home signal speed
  6099_h : 01_h (0417_h)= 10000 6099_h :02_h (0419_h)=1000;
  Send: 01 10 04 17 00 02 04 00 00 27 10 (set seek limit switch speed to 10000)
  Send: 01 10 04 19 00 02 04 00 00 03 E8 (set find home signal speed to 1000)

- Setting the return-to-zero acceleration
  $609A_h$ ($041B_h$) = 200000;
  Send: 01 10 04 1B 00 02 04 00 03 0D 40 (set return to zero acceleration to 200000)

- Write control words
  6040_h (0380_h) = 0x06→0x07→0x0F→0x1F, motor 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 to disable motor)

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

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

2. Blocking the turn to find the way of origin

- Setting the origin return method
  6098_h (0416_h) = 37.
  Send: 01 10 04 16 00 01 02 00 25 (set home return mode to 37)
- 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 blocking detection torque to 30%)
  Send: 01 10 01 72 00 01 02 00 32 (set blocking detection time to 50ms)
- Setup Mode
  Write 2002_h : 01_h ($00B1_h$) = 0, run mode 6060_h ($03C2_h$) = 0x06 to make it work in home return 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 return mode)
- Setting the Find Home Signal Speed
  6099_h : 02_h = 1000.
  Send: 01 10 04 19 00 02 04 00 00 03 E8 (set find home signal speed to 1000)
- Setting the return-to-zero acceleration
  $609A_h$ ($041B_h$) = 200000;
  Send: 01 10 04 1B 00 02 04 00 03 0D 40 (set return to zero acceleration to 200000)
- Write control words
  6040_h = 0x06→0x07→0x0F→0x1F, motor 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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
  Send: 01 10 03 80 00 01 02 00 1F (set 6040_h (0380_h) to 0x1F to run the motor)

5.8 Interpolation mode (IP)

The interpolated position mode is used to synchronize multiple axes. To do this, the advanced controller performs slope and path calculations and passes to the controller the corresponding demand positions that the axes are at at a particular time. The controller interpolates between these intermediate position points.

5.8.1 Structure Plan

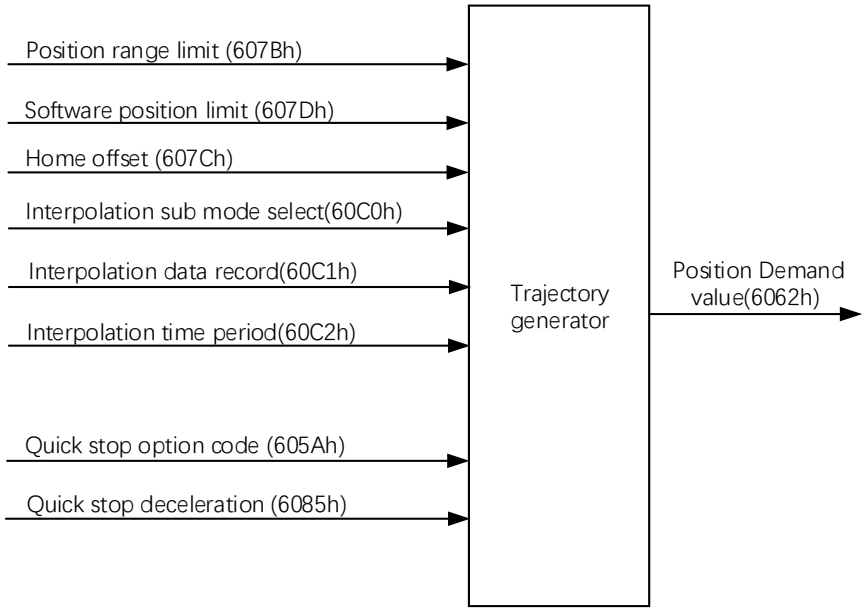


Figure5-27

5.8.2 Relevant Object

The following objects require attention in this mode:

Table5-26

Object Index	Description
$607D_h$: 01_h (03EF)$_h$ $607D_h$: 02_h (03F1)$_h$	Limit value for the target position (user units).
$60C0_h$ (042C)$_h$	Interpolation type selection, this device only supports linear interpolation, the default value is 0
$60C1_h$: 01_h (042D)$_h$	Setting the interpolation target position (user units)
$60C2_h$: 01_h (042F)$_h$ $60C2_h$: 02_h (0430)$_h$	Interpolation time (1-20ms recommended) Defined jointly by $60C2_h$: 01_h ($042F_h$) and $60C2_h$: 02_h (0430_h), interpolation period = $t*10^n$ 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)$_h$	Stopping deceleration rate when performing a "quick stop" (user unit/s^2)

Object Index	Description
6062_h (03C4)$_h$	Target position command value currently in effect within the drive (user units)
6063_h (03C6)$_h$	Feedback of the current absolute position of the motor (encoder unit)

5.8.3 Control Commands and Status Messages

start

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

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 a special function:

Table5-27

Bit	Value	Definition
4	0	disablement interpolation
	1	Enable interpolation
8	0	Execute the instruction in bit 4
	1	Configuration suspension according to 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 6041_h (0381_h) have a special function:

Table5-28

Bit	Value	Definition
10	0	Target position not reached
	1	Reach target position
12	0	The interpolation is not activated
	1	interpolation activation
13	0	No following error
	1	With following error

5.8.4 Functional Description

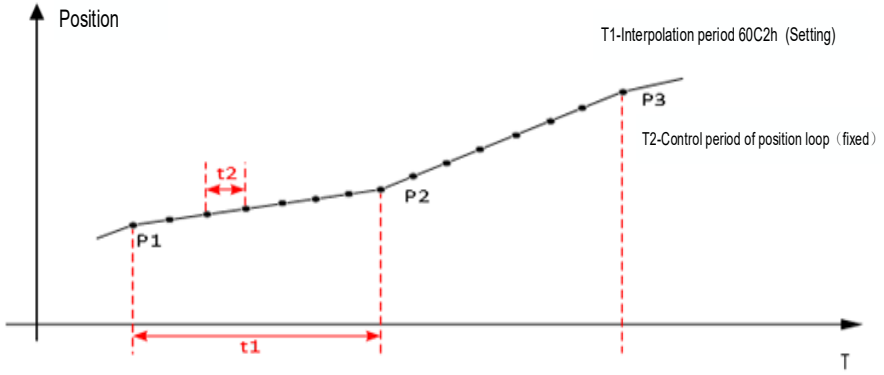


Figure5-28 Schematic diagram of interpolation

Note: t1: interpolation period, which can be set via the object dictionaries $60C2_h : 01_h$ ($042F_h$) and $60C2_h : 02_h$ (0430_h) (1-20ms recommended).

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

P1/P2/P3: Absolute Position, Absolute Position instruction is sent via object dictionary $60C1_h : 01_h$ ($042D_h$), Interpolation mode only supports Absolute Position instruction.

5.8.5 Configuration Example

- Configure to work in interpolation mode:
  Write $2002_h : 01_h$ ($00B1_h$) = 0, run 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 the interpolation time:
  The time constant $60C2_h : 01_h$ ($042F$)$_h$ = 20 with the time exponent $60C2_h : 02_h$ (0430)$_h$ = -3 and the time is 20ms;
  Send: 01 10 04 2F 00 01 02 00 14 (set time constant to 20)
  Send: 01 10 04 30 00 01 02 00 FD (set time index to -3)
- The motor runs:
  Write control word 6040_h (0380_h) = 0x06→0x07→0x0F→0x1F
  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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
  Send: 01 10 03 80 00 01 02 00 1F (set 6040_h (0380_h) to 0x1F to run the motor)
- The host computer writes the interpolation position according to the interpolation cycle $60C1_h : 01_h$ ($042D_h$) (Only absolute position instruction is supported.)

5.9 Cyclic Synchronized Position Mode (CSP)

In this mode, an absolute position preset value is sent to the controller via the fieldbus at fixed intervals (hereinafter referred to as "cycles"). At this point, the controller no longer calculates the ramp, it only follows the preset value. The target position is Sendted via the PDO and the controller reacts to it immediately. bit 4 in the Controlword does not need to be set (unlike in profile position mode). The target preset value is an absolute value and therefore independent of the number of times each cycle is sent.

5.9.1 Structure Plan

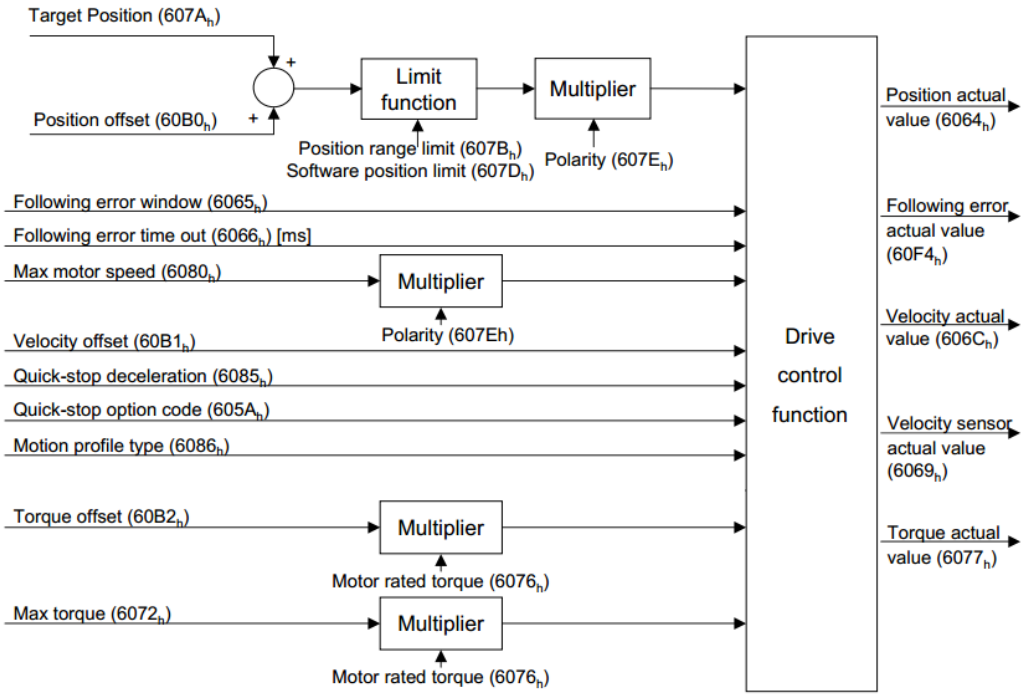


Figure5-29

5.9.2 Relevant Object

The following objects require attention in this mode:

Table5-29

Object Index	Description
607Ah (03E7)h	Preset target position (user units)
607Dh : 01h (03EF)h 607Dh : 02h (03F1)h	Limit values for target position (user units)
607Eh (03F3)h	Direction of rotation (polarity), see "4.3 607Eh (03F3h): polarity".
6065h (03CA)h	Follow deviation threshold window (user units)
6066h (03CC) h	Time range of the following error in milliseconds. If the actual position is out of the error range 6065h (03CAh) for longer than the specified time 6066h (03CCh), the follow error will be triggered and bit13 of the control word will be set to 1
6080h (03F6)h	Maximum speed of motor (rpm)

6085h (0400)h	Stopping deceleration rate when performing a "quick stop" (user unit/s^2)
605Ah (03BF)h	Quick stop mode selection
605Dh (03C0)h	Pause mode selection
6086h (0402)h	The type of ramp to be traveled, if the value is "0", the shock will not be limited, if the value is "3", the shock will be limited with the value from 60A4h : 01h - 04h (041Eh ~0424h), this device only uses 01h -02h (041Eh ~0420h).
60B0h (0426)h	Position offset (user units)
60B1h (0428)h	Speed bias (user units)
6064h (03C8)h	Feedback on the current absolute user position of the motor (user units)
606Ch (03D5)h	Current actual speed feedback value (rpm)
60F4h (0440)h	Real-time following deviation (user units)

5.9.3 Control Commands and Status Messages

start

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

control word

The bits in object 6040h (0380h) have no special function in cyclic synchronized position mode, see section 3.1.2 Control words for the definition of the general-purpose bits.

Note: CSP mode only supports absolute position commands

status word

15 14	13	12	11	10	9 0
(see 3.1.2)	Following error	Target position ignored	(see 3.1.2)	reserved	(see 3.1.2)

MSB LSB

In cyclic synchronized position mode, the following bits in object **6041**h (0381h) have a special function:

Table5-30

Bit	Value	Description
12	0	The controller does not follow the target preset, and the preset value of 607Ah (03E7h) is ignored.
	1	The controller follows the target preset and object 607Ah (03E7h) is used as position control input.
13	0	No following error
	1	With following error

5.9.4 Functional Description

In this mode, an absolute position preset value is sent to the controller via the fieldbus at fixed intervals (hereinafter referred to as "cycles").

5.9.5 Configuration Example

- Configuration mode:
  Working in cyclic synchronized position mode: write 2002_h : 01_h ($00B1_h$) = 0, run 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 synchronized position mode)
- The motor runs:
  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 to make motor ready)
  Send: 01 10 03 80 00 01 02 00 07 (set 6040_h (0380_h) to 0x7 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
- The host computer sends the target position according to the synchronization cycle $607A_h$ ($03E7$)$_h$

5.10 Cyclic Synchronized Velocity Mode (CSV)

In the cyclic synchronized speed mode, the upper controller sends the calculated target speed $60FF_h$ (0448_h) to the driver in a cyclic synchronized manner, and the speed and torque adjustments are executed internally.

5.10.1 Structure Plan

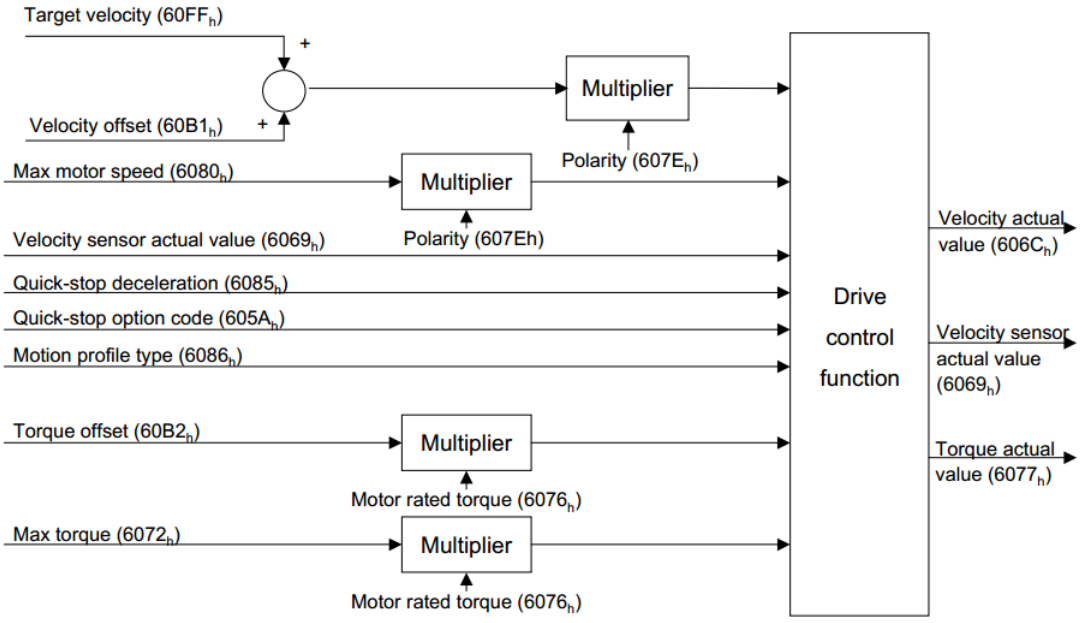


Figure5-30

5.10.2 Relevant Object

The following objects require attention in this mode:

Table5-31

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 (in rpm)
$607E_h$ (03F3)$_h$	Direction of rotation (polarity), see "4.3 $607E_h$: Polarity".
$60FF_h$ (0448)$_h$	Setting the target speed (user unit/s)
6080_h (03F6)$_h$	Maximum speed of motor (rpm)
6085_h (0400)$_h$	Stopping deceleration rate when performing a "quick stop" (user unit/s^2)
6086_h (0402)$_h$	Select the type of curve, i.e. slope. If the value is "0", the shock (plus acceleration) is not limited, i.e., the trapezoidal curve; If the value is "3", the impact will be limited by the value in $60A4_h$: 01_h - 04_h ($041E_h$ ~0424_h), only 01_h -02_h ($041E_h$ ~0420_h) is used for this device.
$60B1_h$ (0428)$_h$	Position offset (user units)
$60B2_h$ (042A)$_h$	Torque bias (0.1%)
6072_h (03DC)$_h$	Maximum torque (0.1%) on the entire ramp (acceleration, torque conservation, braking)

5.10.3 Control Commands and Status Messages

start

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

control word

In cyclic synchronized speed mode, the bits in the object 6040_h (0380)$_h$ have no special function, see section 3.1.2 Control words for the definition of the general-purpose bits.

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 synchronized speed mode, the following bits in object **6041**$_h$ (0381_h) have a special function:

Table5-32

Bit	Value	Definition
12	0	Target speed will be rounded
	1	Target speed as planner input

5.10.4 Functional Description

In the cyclically synchronized speed mode, the upper controller sends the calculated target speed $60FF_h$ (0448_h) to the motor driver in a cyclically synchronized manner, and the speed regulation is performed internally by the motor.

5.10.5 Configuration Example

- Setup Mode:
  Write 2002_h : 01_h ($00B1_h$) = 0, Run Mode 6060_h ($03C2_h$) = 0x09 to operate in Cyclic Synchronized Speed 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 synchronized speed mode)
- Write control words
  6040_h (0380_h) = 0x06→0x07→0x0F;
  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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040_h (0380_h) to 0xF to enable motor)
  The motor is running and the host computer needs to write the speed object according to the synchronization cycle $60FF_h$ (0448)$_h$

5.11 Cyclic Synchronous Torque Mode (CST)

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

5.11.1 Structure Plan

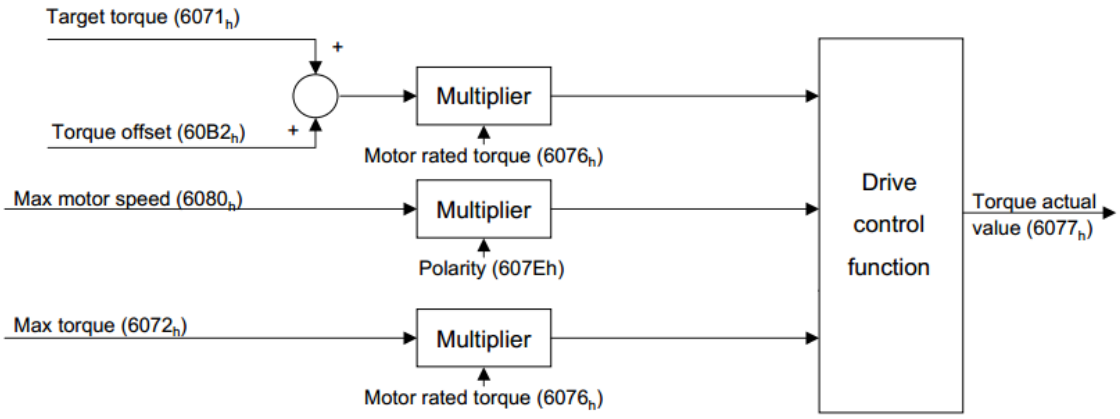


Figure5-31

5.11.2 Relevant Object

The following objects require attention in this mode:

Table5-33

Object Index	Description
$605A_h$ ($03BF$)$_h$	Quick stop mode selection
$605D_h$ ($03BE$)$_h$	Pause mode selection
6071_h ($03DB$)$_h$	Setting target torque (0.1%)
6072_h ($03DC$)$_h$	Maximum torque (0.1%) on the entire ramp (acceleration, torque conservation, braking)
6073_h ($03DD$)$_h$	Maximum allowable current (0.01A)
6080_h ($03F6$)$_h$	Maximum speed of motor (rpm)
$60B2_h$ ($042A$)$_h$	Torque bias (0.1%)
6088_h (0405)$_h$	Torque ramp type (0-slope, 2-none)
6077_h ($03E3$)$_h$	Torque actual value (0.1%)

5.11.3 Control Commands and Status Messages

start

To enable this mode, the value "0" must be set in object 2002_h : 01_h ($00B1_h$) and "10" in object 6060_h ($03C2_h$).

control word

In cyclic synchronous torque mode, the bits in object 6040_h (0380_h) have no special function; see section 3.1.2 Control words for the definition of the general-purpose bits.

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 a special function:

Table5-34

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

5.11.4 Functional Description

In this mode, the upper controller sends the calculated target torque 6071$_h$ (03DB$_h$) to the driver in a cyclically synchronized manner.

5.11.5 Configuration Example

- Setup Mode:
  Write 2002$_h$: 01$_h$ (00B1$_h$) = 0, run mode 6060$_h$ (03C2$_h$) = 0x0A to operate 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 words
  6040$_h$ (0380$_h$) = 0x06→0x07→0x0F, motor running, upper computer needs to write target torque 6071$_h$ (03DB)$_h$
  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 to disable motor)
  Send: 01 10 03 80 00 01 02 00 0F (set 6040$_h$ (0380$_h$) to 0xF to enable motor)

5.12 NiMotion mode

5.12.1 Summary

The NiMotion mode is different from the CIA402 mode in that there is no state machine management, and the position, speed and torque are controlled through the parameters configured in the multi-function terminals, including the NiMotion position mode, the NiMotion speed mode and the NiMotion torque mode. 2002_h : 01_h ($00B1_h$) (control mode selection) is used to determine which operating mode is in operation. See 5.1 Overview of Control Modes for specific correspondences.

5.13 NiMotion Position Mode

5.13.1 Structure Plan

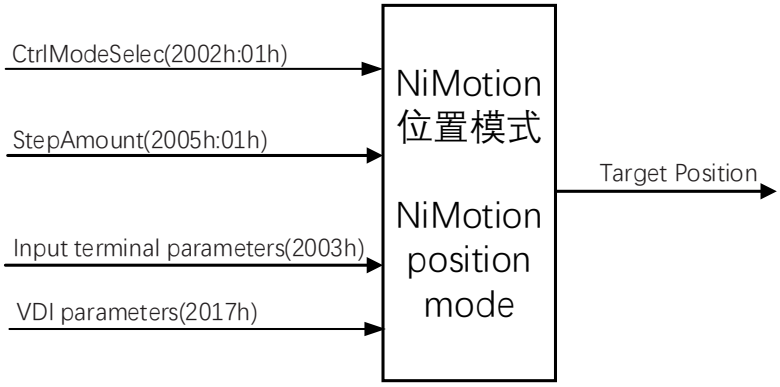


Figure5-32

5.13.2 Relevant Object

The following objects require attention in this mode:

Table5-35

Object Index	Description
2002_h :01_h ($00B1$)$_h$	Operation mode selection
2005_h :01_h ($010D$)$_h$	Position command source, 0-pulse, 1-step, 2-multi-segment position
2005_h :05_h (0112)$_h$	Step amount setting (encoder unit)
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h ($00D5_h$ ~$00E6$)$_h$	Functional configuration of the physical input terminals (see: "6.1.1 Digital input") Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.

5.13.3 Control Commands and Status Messages

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

Enable

Prerequisite for this mode of operation: the integrated low-voltage servo 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 a source of position instructions, meaning: 0-pulse, 1-step amount;

2 Setting the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on terminal settings).

If the set position instruction source is 1, configure one input terminal as motor enable and configure its valid logic level according to the actual situation, the default is low and valid; set the other input terminal as step amount enable and configure its valid logic level according to the actual situation, the default is low and valid;

If the source of the set position command is 0 (external pulse given), in addition to setting the terminals for motor enable, it is also necessary to set the input terminals DI1 and DI2 for the corresponding pulse input function.

3 Position Setting

If the source of the position instruction is selected as 1, you need to set the step amount given value to be written to object 2005_h : 05_h (0112_h);

If the source of the position command is selected as 0, a pulse signal from input terminals DI1 and DI2 is required.

4 Run Run

If you select the position command source as 1, enable the motor through the terminal, and then enable the step amount signal through the terminal, the motor travels the distance (unit: encoder unit) set by 2005_h :05_h (0112_h), and the motor travels once for each enabled step amount;

If the source of position command is selected as 0, motor enable is performed according to the terminal and the motor runs according to the pulse signal from the input terminal.

5.13.4 Functional Description

In this mode, the motor will travel a controlled distance based on the input signal from the terminal.

5.13.5 Example

Stepping volume 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 01 (set to NiMotion position mode)

- Setting the position instruction source as a step amount: 2005_h : 01_h ($010D_h$)=1

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

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

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

- Set motor enable: 2003_h : 03_h ($00D5_h$) = 1, logic active low: 2003_h : 04_h ($00D6_h$)=0

Send: 01 10 00 D5 00 01 02 00 01 (sets DI1 to step enable)

Send: 01 10 00 D6 00 01 02 00 00 (sets DI1 logic active low)

- Setting the step amount 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 (sets DI2 to step enable)

Send : 01 10 00 D8 00 01 02 00 03 (sets DI2 logic to be active on falling edge)

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

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)

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

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

- Set the input terminals:

  DI1 pulse input : 2003_h : 03_h ($00D5_h$)=42

  Send: 01 10 00 D5 00 01 02 00 2A (sets DI1 to input pulse)

  DI2 Pulse Input direction: 2003_h : 05_h ($00D7_h$)=43

  Send: 01 10 00 D7 00 01 02 00 2B (sets DI2 to input pulse direction)

  DI3 function number is set to motor enable:

  Function selection for step enable: 2003_h : 07_h ($00D9_h$) =1

  Send: 01 10 00 D9 00 01 02 00 01 (sets DI3 to step enable)

  Logic selection active low: 2003_h : 08_h ($00DA_h$) =0

  Send: 01 10 00 DA 00 01 02 00 00 (sets DI3 logic active low)

Running:

- Motor enable: DI3 input low level
- Pulse input control: Terminals DI1 and DI2 input pulses and the motor is controlled in position according to the input pulses.

5.14 NiMotion Speed Mode

5.14.1 Structure Plan

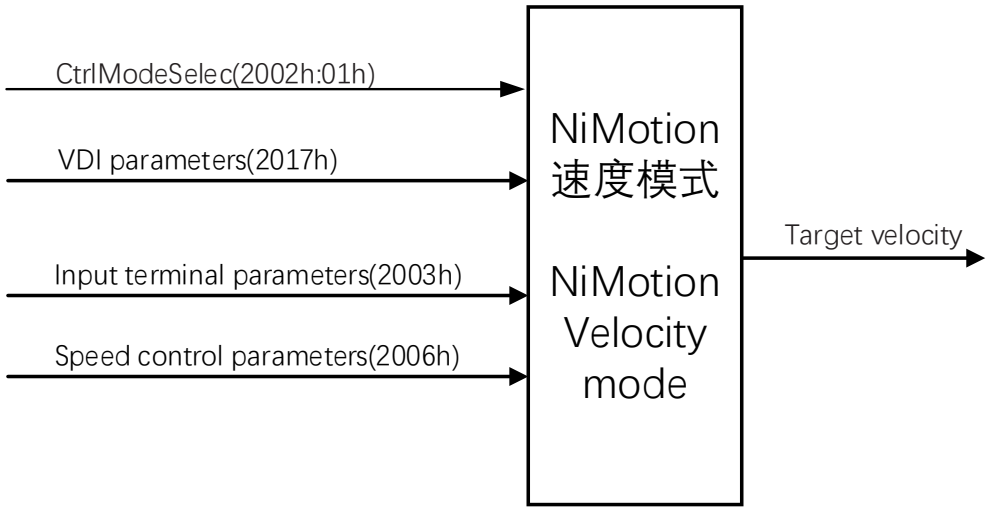


Figure5-33

5.14.2 Relevant Object

The following objects require attention in this mode:

Table5-36

Object Index	Description
2000_h : $0F_h$ (006B)$_h$	Maximum speed, rpm
2002_h : 01_h (00B1)$_h$	Operation mode selection
2006_h : 01_h (0148_h)	NiMotion speed mode master speed feed source, 0-digital feed, 3-duty cycle feed (20Hz-20kHz, 1kHz recommended)
2006_h : 02_h (0149_h)	NiMotion speed mode auxiliary speed feed source, 0 - digital feed, 3 - multispeed
2006_h : 03_h ($014A_h$)	NiMotion speed mode speed command selection, 0-primary speed given, 1-secondary speed given, 2-primary and secondary speed given (default 0-primary speed given)
2006_h : 04_h ($014B_h$)	Speed command numerical given value in rpm
2006_h : 07_h ($014E_h$)	The speed command acceleration ramp time constant, in ms, indicates the time taken to accelerate from 0 rpm to 1000 rpm.
2006_h :08_h ($014F_h$)	Speed command deceleration ramp time constant in ms
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h ($00D5_h$ ~$00E6$)$_h$	Functional configuration of the physical input terminals (see: "6.1.1 Digital input") Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.

5.14.3 Control Commands and Status Messages

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

Enable

Prerequisite for this mode of operation: the integrated low-voltage servo motor is switched to NiMotion speed mode (2002_h :01_h ($00B1_h$) set to 2).

1 Speed source selection:

The object is the speed instruction source 2006_h : 01_h (0148_h), meaning: 0 - number given, 3 - duty cycle given;

2 Set the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on 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 source of speed command set in the previous step is 3 (duty cycle given), you need to set the corresponding input terminals DI1 and DI2 for duty cycle input function.

3 Velocity, acceleration and maximum speed settings

If the source of the speed command is selected as 0-digit given, you need to set the speed given value to be written to object 2006_h : 04_h ($014B_h$);

If the source of the select speed command is 3-duty cycle given, it is necessary to input pulse signals from DI1 (duty cycle) and DI2 (direction); when the direction input is invalid, 0% duty cycle corresponds to a negative maximum speed value 2000_h : $0F_h$ ($006B_h$), and 100% duty cycle corresponds to speed 0; when the input direction is valid, 0% duty cycle corresponds to a positive maximum speed value 2000_h : $0F_h$ ($006B_h$), 100% duty cycle corresponds to speed 0;

If the source of main speed command is selected as 4~50% duty cycle speed regulation given, it is necessary to input pulse signal from DI1 (duty cycle), 0~50% corresponds to negative limiting speed 2006_h : $0B_h$ (0152_h)~0,50%-100% corresponds to 0~positive limiting speed 2006_h : $0A_h$ (0151_h)

Configuring the speed of acceleration and deceleration 2006_h : 07_h ($014E_h$) and 2006_h : 08_h ($014F_h$)

Configuration of the maximum speed 2000_h : $0F_h$ ($006B_h$), with a duty cycle regulation of 100% corresponding to the maximum speed

4 (of a computer) run

The motor is enabled according to the set motor enable terminal, and the motor runs at the set speed.

5.14.4 Functional Description

In this mode, the motor will run to the appropriate speed based on the input from the terminals or

digital givens.

5.14.5 Example

Realization of digitally given speed regulation:

1 Parameter Settings:

- Set speed mode to 2 (NiMotion speed mode): 2002_h : 01_h ($00B1_h$)=2
  Send: 01 10 00 B1 00 01 02 00 02
- Set the main speed command source to 0 (digit given): 2006_h : 01_h (0148_h)=0;
  Send: 01 10 01 48 00 01 02 00 00
- Set the auxiliary speed command source to 0 (number given): 2006_h : 02_h (0149_h)=0;
  Send: 01 10 01 49 00 01 02 00 00
- 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: 2006_h : 08_h ($014F_h$)=100
  Send: 01 10 01 4F 00 01 02 00 64
- Set the number given as 60rpm, 2006_h : 04_h ($014B_h$)=60
  Send: 01 10 01 4B 00 01 02 00 3C
- Terminal Setting
  Set the DI1 terminal:
  Function selection for step enable: 2003_h : 03_h ($00D5_h$) =1
  Send: 01 10 00 D5 00 01 02 00 01
  Logic select low active: 2003_h : 04_h ($00D6_h$)=0
  Send: 01 10 00 D6 00 01 02 00 00

2 Running:

- DI1 input is low, the motor is speed-controlled according to the digitally given 2006_h : 04_h ($014B_h$)

Duty cycle input speed regulation is realized:

1 Parameter Settings:

- Set speed mode to 2 (NiMotion speed): 2002_h : 01_h ($00B1_h$)=2
  Send: 01 10 00 B1 00 01 02 00 02
- 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 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: 2006_h : 08_h ($014F_h$)=100
  Send: 01 10 01 4F 00 01 02 00 64

Set the maximum speed to 3000rpm: 2000_h : $0F_h$ ($006B_h$)=3000
Send: 01 10 00 6B 00 01 02 0B B8

- Terminal Setting
  DI1 duty cycle input: 2003_h : 03_h ($00D5_h$)=44
  Send: 01 10 00 D5 00 01 02 00 2C

  DI2 duty cycle direction input: 2003_h : 05_h ($00D7_h$)=45
  Send: 01 10 00 D7 00 01 02 00 2D

  DI3 function number is set to step enable:
  Function selection for step enable: 2003_h : 07_h ($00D9_h$) =1
  Send: 01 10 00 D9 00 01 02 00 01

  Logic selection active low: 2003_h : 08_h ($00DA_h$) =0
  Send: 01 10 00 DA 00 01 02 00 00

2 Running:

- DI1 inputs a square wave with a certain duty cycle, DI3 inputs a low level to enable the motor, and the motor speeds up according to the input duty cycle.

5.15 NiMotion Torque Mode

5.15.1 Structure Plan

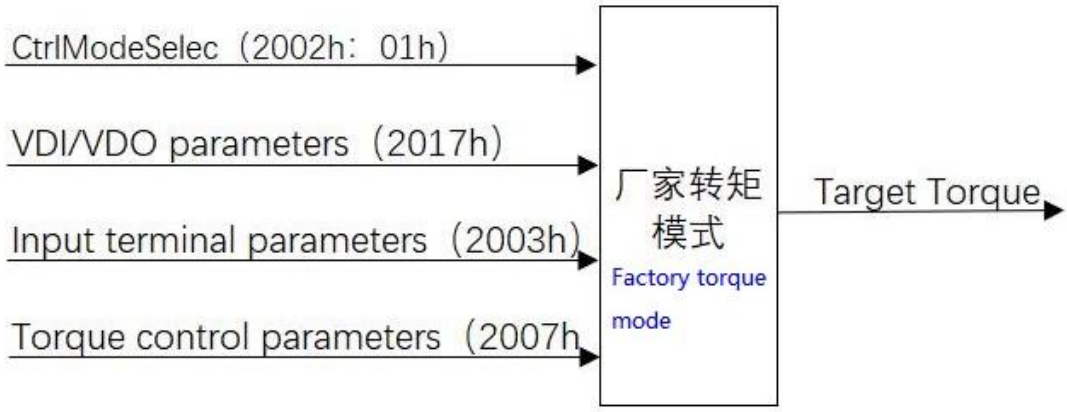


Figure5-34

5.15.2 Relevant Object

The following objects require attention in this mode:

Table5-37

Object Index	Description
2002_h : 01_h ($00B1$)$_h$	Operation mode selection
2007_h : 01_h ($015E$)$_h$	NiMotion torque mode main torque feed source, 0-digital feed
2007_h : 02_h ($015F$)$_h$	NiMotion Torque Mode Auxiliary Torque Feed Source, 0-Digital Feed

Object Index	Description
2007_h : 03_h (0160)$_h$	NiMotion torque mode torque command selection, 0-primary torque given, 1-secondary torque given, 2-primary and secondary torque given (default 0)
2007_h : 04_h (0161)$_h$	Torque command digital given value (0.1%)
2007_h : $0A_h$ (0167)$_h$	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 (0326h ~0346h)	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h (00D5h ~00E6h)	Functional configuration of physical input terminals (see "6.1.1 Digital Input" for details) Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.

5.15.3 Control commands and status messages

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

start

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

1 Torque source selection:

Object 2007_h : 01_h ($015E_h$) is the torque instruction source, meaning: 0-number given;

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

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

3 Torque setting and torque and speed limit settings

Selecting the source of the speed command as digitally given, the torque given value needs to be set and written to object 2007_h : 04_h (0161_h);

Configure positive and negative internal torque limits 2007_h : $0A_h$ (0167_h) and 2007_h : $0B_h$ (0168)$_h$

Configuring 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 according to the set motor enable terminal, and the motor operates at

the set torque.

5.15.4 Functional Description

In this mode, the motor will run to the corresponding torque by means of the enable control and torque digitalization of the physical or virtual terminals. When the motor's operating speed exceeds the value set by the speed limits (forward speed limit value 2007_h : 10_h ($016D_h$) and reverse speed limit value 2007_h : 11_h ($016E_h$)), it automatically switches to speed mode control and controls the current speed within the speed limits. When the given target torque is detected to be less than the average torque at the current speed, the motor exits speed control and reverts to torque mode control.

5.15.5 Example

1. Parameter Settings:

- Set the speed mode to 3 (NiMotion torque mode): 2002_h : 01_h ($00B1_h$)=3;
  Send: 01 10 00 B1 00 01 02 00 03
- Set the speed command source to 0 (number given): 2007_h : 01_h ($015E_h$)=0
  Send: 01 10 01 5E 00 01 02 00 00
- Setting the speed command selection as the main torque command: 2007_h : 03_h (0160_h)=0
  Send: 01 10 01 60 00 01 02 00 00
- Set the positive and negative internal torque limits to 100%: 2007_h : $0A_h$ (0167_h)=1000 and 2007_h : $0B_h$ (0168_h)=1000
  Send: 01 10 01 67 00 01 02 03 E8 (sets positive internal torque limit to 100%)

  Send:01 10 01 68 00 01 02 03 E8 (sets reverse internal torque limit to 100%)
  Set the forward and reverse speed limit values to 3000rpm: 2007_h : 10_h ($016D_h$)=3000 and 2007_h : 11_h ($016E_h$)=3000
  Send: 01 10 01 6D 00 01 02 0B B8 (set forward speed limit value to 3000rpm)

  Send: 01 10 01 6E 00 01 02 0B B8 (set reverse speed limit value to 3000rpm)
- Setting the numeric value to 100%, 2007_h : 04_h (0161_h)=1000
  Send: 01 10 01 61 00 01 02 03 E8
- Terminal Setting
  Set the DI1 terminal:
  Function selection for step enable: 2003_h : 03_h ($00D5_h$) =1
  Send: 01 10 00 D5 00 01 02 00 01

  Logic selection active low: 2003_h : 04_h ($00D6_h$) =0
  Send: 01 10 00 D6 00 01 02 00 00

2. Running:

- DI1 input is low and the motor runs at the torque digitally given 2007_h : 04_h (0161_h)

5.16 Multi-Segment Position Mode

5.16.1 Structure Plan

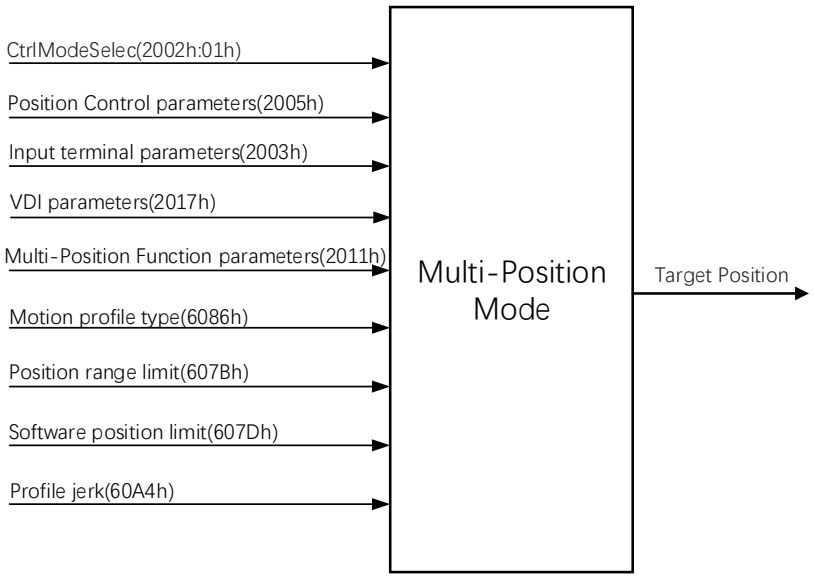


Figure5-35

5.16.2 Relevant Object

The following objects require attention in this mode:

Table5-38

Object Index	Description
2002_h : 01_h ($00B1$)$_h$	Operation mode selection
2005_h : 01_h ($010D$)$_h$	Position command source, 0-pulse, 1-step, 2-multi-segment position
2005_h : 05_h (0112)$_h$	Step amount setting (encoder unit)
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h ($00D5$h ~$00E6$)$_h$	Functional configuration of the physical input terminals (see: "6.1.1 Digital input") Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.
2011_h (0291_h ~$02E6$)$_h$	Multi-stage position function parameter set
6086_h (0402)$_h$	Selection of the type of planner curve, i.e. slope. If the value is "0", then there will be no limitation of shocks (plus acceleration), i.e. a trapezoidal curve; If the value is "3", the shock (plus acceleration) will be limited by the value of $60A4_h$: 01_h -02_h ($041E_h$

Object Index	Description
	~0420h), i.e., S-curve, and the device only uses the indexes 01h and 02h (041Eh and 0420h).
607Dh (03EFh ~03F1h)	Limit values for target position (user units)
60A4h (041Eh ~0424h)	Contour plus acceleration, see 6.3.2 for details of the implementation process.

5.16.3 Control Commands and Status Messages

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

Enable

Prerequisite for operation in this mode: Motor operation mode is switched to NiMotion position mode (2002h : 01h (00B1h) write 1).

1. Location Source Selection:

   Object 2005h :01h (010Dh) is the source of position instructions, 2005h :01h (010Dh) Meaning: 0-pulse, 1-step amount, 2-multiple positions;

2. Setting the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on terminal settings).

   Set the position command source as 2, configure one input terminal as motor enable, and configure its valid logic level according to the actual situation, with the default low level valid;

   Set the other input terminal as multi-segment position/speed enable (Function No. 28), and configure its valid logic level according to the actual situation, the default is valid low;

3. Position Setting

   The corresponding operating mode and operating parameters for each segment are set via the multi-segment position function parameter set 2011h (0291h ~02E6h).

4. (of a computer) run

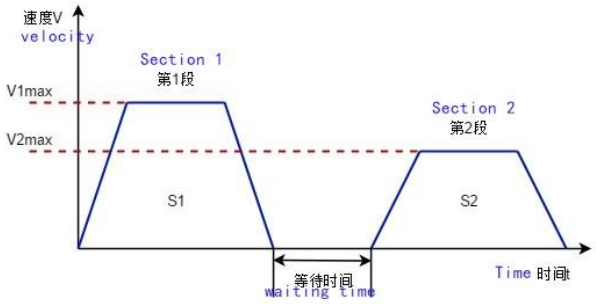
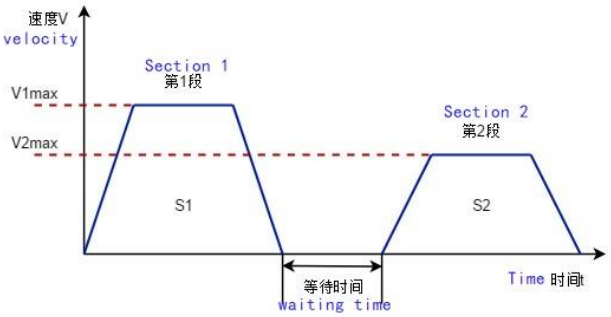
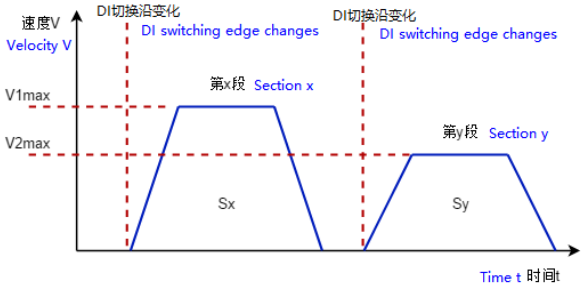
   The motor is first enabled via one terminal, and then the multi-segment position instruction is enabled via the other terminal, and the motor runs in accordance with the operation mode and the operation parameters of each segment set in the multi-segment position function parameter set 2011h (0291h ~02E6h).

5.16.4 Functional Description

The multi-segment position mode has 3 types of operation, namely, single operation, cyclic operation and DI switching operation. When using the multi-segment position function, the DI port must be set to enable (function No. 1) and multi-segment position/speed enable (function No. 28) respectively; during each segment operation, it must be ensured that the enable is valid; in the non-DI switching operation mode, in addition to the enable being valid, it must be ensured that the multi-segment position/speed enable is valid, and if the multi-segment position/speed enable is turned off, it will give up the displacement commands that have not been sent in this segment and stop the machine.

Specific descriptions of each mode of operation are shown in the table below:

Table5-39

Operation mode	Description	Running curve
single run 2011_h :01_h (0291_h)=0	Paragraphs 1 through 16 are run once; Automatic incremental switching of segment numbers; Waiting time can be set between segments; Multi-segment position/speed enable is level active.	
run in a loop 2011_h :01_h (0291_h)=1	2011_h :06_h (0296_h) The number of times the starting segment set by 2011 :06 (0296) has run the end segment set by 2011_h :02_h (0292_h) the number of times the 2011_h :04_h (0294_h) has run the 2011 :04 (0294) set by 2011 :02 (0292); Automatic incremental switching of segment numbers; Waiting time can be set between segments; Multi-segment position/speed enable is level active; Note: If the setting value of the end segment is smaller than the start segment, segments 1 through 16 cycle once.	
DI switching operation 2011_h :01_h (0291_h)=2 **Note: Physical and virtual terminals can be used in combination and the combination must be 4-digit; for example: Physical terminal selection**	Position segment number switching is determined by the decimal 0 to 15 corresponding to the 4-digit binary number consisting of multifunction numbers 6, 7, 8 and 9; The wait time set between segments is invalid; Multi-segment position/speed enable is level active; The relationship between segment number and multifunction number corresponds as follows:	

<table>
<tr><th>Operation mode</th><th>Description</th><th>Running curve</th></tr>
<tr><td>DI1=6,DI2=7,
Virtual terminals
VI1=8,VI2=9.</td><td>
<table>
<tr><th>Fun 9</th><th>Fun 8</th><th>Fun 7</th><th>Fun 6</th><th>section number</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td colspan="5">.....</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>16</td></tr>
</table>
</td><td></td></tr>
</table>

5.16.5 Configuration Example

Multi-stage position-single run control

- 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 the source of the position command to a multi-segment position: 2005_h :01_h ($010D_h$)=2;

Send: 01 10 01 0D 00 01 02 00 02

Sets the multi-segment position function parameter group:

Segment position run mode: 2011_h :01_h (0291_h) = 0 (single run);

Send: 01 10 02 91 00 01 02 00 00

Position instruction type selection: 2011_h :05_h (0295_h) = 0 (relative displacement);

Send: 01 10 02 95 00 01 02 00 00

Operational parameters for segment 1 displacements:

Moving Displacement 2011_h :07_h (0297_h) = 10000 (user units), Maximum Allowable Speed 2011_h :08_h (0299_h) = 500 (in rpm).

Send: 01 10 02 97 00 02 04 00 00 27 10 (set first segment displacement to 10000)

Send: 01 10 02 99 00 01 02 01 F4 (set the maximum operating speed of the first displacement to 500 rpm)

Acceleration and deceleration time 2011_h :09_h ($029A_h$) = 500 (in ms), waiting time after displacement completion 2011_h :$0A_h$ ($029B_h$) = 1000 (in ms)

Send: 01 10 02 9A 00 01 02 01 F4 (set acceleration/deceleration time to 500ms)

Send: 01 10 02 9B 00 01 02 03 E8 (set the wait time after the first displacement is completed to 1000ms)

Paragraphs 2 to 16 are set with reference to the operational parameters of paragraph 1 displacement;

Set motor enable: 2003_h :03_h ($00D5_h$)=1, logic active low: 2003_h :04_h ($00D6_h$)=0;

Send: 01 10 00 D5 00 01 02 00 01 (set DI1 to enable)

Send: 01 10 00 D6 00 01 02 00 00 (sets DI1 logic active low)
Setting Multi-Segment Position/Speed Enable: 2003_h :05_h ($00D7_h$)=28, Logic is active low: 2003_h :06_h ($00D8_h$)=0;
Send: 01 10 00 D7 00 01 02 00 1C (sets DI2 to multisegment position enable)
Send: 01 10 00 D8 00 01 02 00 00 (sets DI2 logic active low)

- run

Motor enable: DI1 input low;
Multi-segment position/speed enable, DI2 input low, the motor runs in accordance with the operation mode and operation parameters set in the 2011_h (0291_h ~$02E6_h$) Multi-segment position function parameter set.

5.17 Multi-Speed Mode

5.17.1 Structure Plan

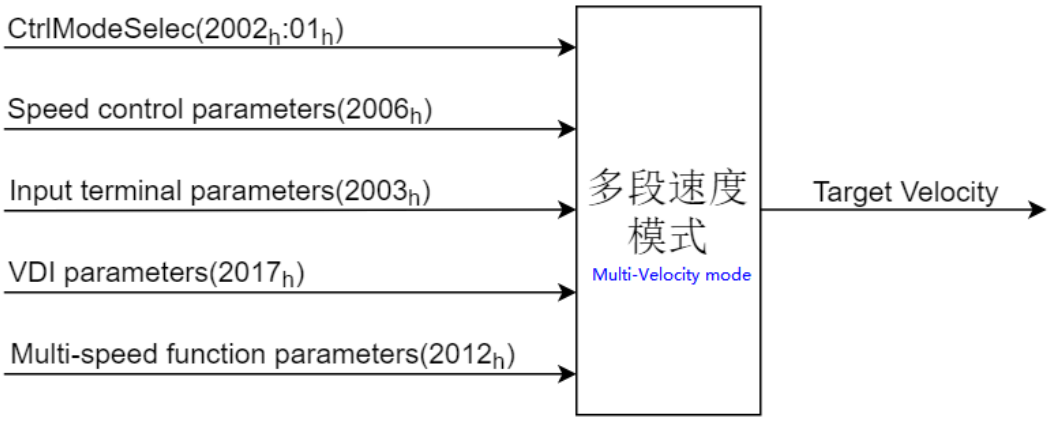


Figure5-36

5.17.2 Relevant Object

The following objects require attention in this mode:

Table5-40

Object Index	Description
2002h : 01h (00B1)h	Operation mode selection
2006h : 01h (0148)h	NiMotion speed mode master speed feed source, 0-digital feed, 3-duty cycle feed (20Hz-20kHz, 1kHz recommended)
2006h : 02h (0149)h	NiMotion speed mode auxiliary speed feed source, 0 - digital feed, 3 - multispeed
2006h : 03h (014A)h	NiMotion speed mode speed command selection, 0 - primary speed given, 1 - secondary speed given, 2 - primary and secondary speed given (default 0)
2006h : 04h (014B)h	Speed command numerical given value in rpm
2006h : 07h (014E)h	The speed command acceleration ramp time constant, in ms, indicates the time taken to accelerate from 0 rpm to 1000 rpm.
2006h : 08h (014F)h	Speed command deceleration ramp time constant in ms
2017h (0326h ~0346)h	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003h (00D5h ~00E6)h	Functional configuration of the physical input terminals (see: "6.1.1 Digital input") Note: You cannot have multiple input terminals configured as the same switch at the same time, or you will report a terminal setup fault.
2012h (02E9h ~0323)h	Multi-Segment Speed Function Parameter Set

5.17.3 Control Commands and Status Messages

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

Enable

Prerequisite for operation in this mode: Motor operation mode is switched to NiMotion speed mode (2002_h : 01_h ($00B1_h$) set to 2).

1. Speed source selection:

   The object is the auxiliary speed command source 2006_h :02_h (0149_h), meaning: 0-number given, 3-multiple speeds,;

   With the speed command selection 2006_h :03_h ($014A_h$) it is the main speed command, the auxiliary speed command or the main and auxiliary speed commands that actually take effect.

2. Set the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on 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.

   Configure the other input terminal as multi-segment enable (function No. 28), and configure its effective logic level according to the actual situation, the default is effective low level.

3. Speed Setting

   Setting of the corresponding operating mode and operating parameters for each segment via the multi-segment speed function parameter group 2012_h ($02E9_h$ ~0323_h)

4. (of a computer) run

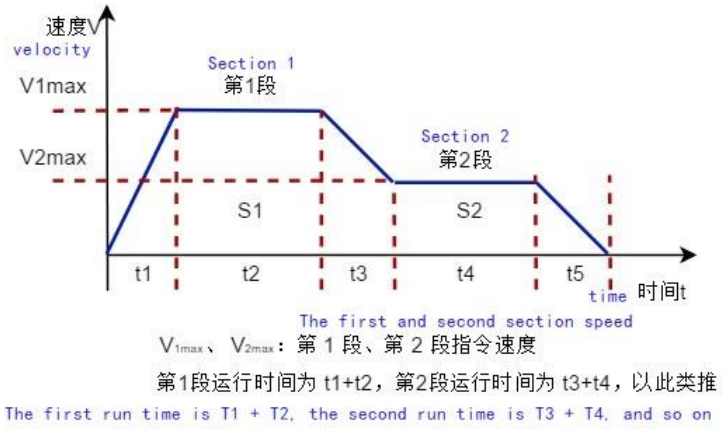
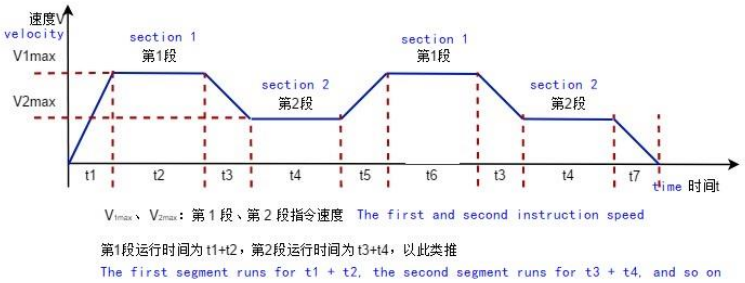
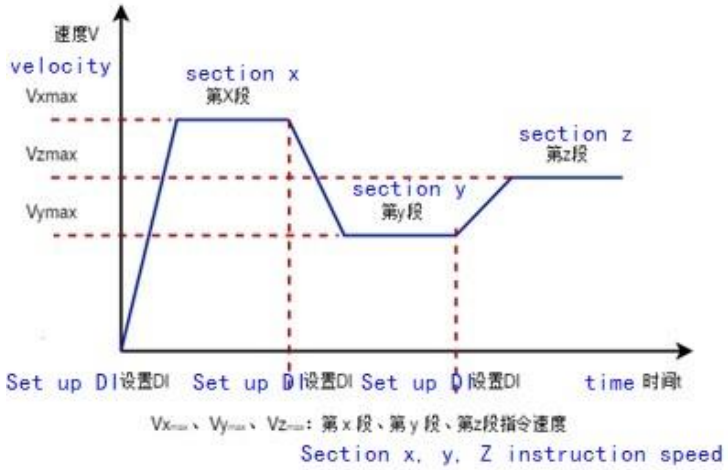
   If the motor is enabled via one of the terminals, the motor operates in accordance with the operation mode set in the multispeed function parameter set 2012_h ($02E9_h$ ~0323_h) and the operation parameters for each segment.

5.17.4 Functional Description

The multi-segment speed mode has 3 types of operation, namely single run, cycle run and DI switching run. When using multi-segment position/speed function, one DI port must be set as motor enable (function No. 1) and multi-segment enable (function No. 28); during the operation of each segment speed command, it must be ensured that the motor enable is valid, if the motor enable is turned off, the motor will be shut down according to the deactivation mode. In the non-DI switching operation mode, in addition to the motor enable being valid, the multi-segment enable must be guaranteed to be valid; if the multi-segment enable is turned off, the motor will not be able to switch between multi-segment speeds.

Specific descriptions of each mode of operation are shown in the table below:

Table 5-41

<table>
<tr><th>Operation mode</th><th>Description</th><th>Running curve</th></tr>
<tr><td>single run
2012h :01h
(02E9h) = 0</td><td>Paragraphs 1 through 16 were run once;
Automatic incremental switching of segment numbers.
Multi-segment enable is level active.</td><td> 
</td></tr>
<tr><td>run in a loop
2012h :01h
(02E9h) = 1</td><td>Segment 1 to the end segment set by 2012h :02h (02EAh);
Run 2012h :03h (02EBh) the number of times set;
Automatic incremental switching of segment numbers.
Multi-segment enable is level active.</td><td> 
</td></tr>
<tr><td>DI switching operation
2012h :01h
(02E9h)= 2
Note: Physical and virtual terminals can be used in combination and the combination must be 4-digit; for example: Physical terminal selection DI1=6,DI2=7,Virtual terminal VI1=8,VI2=9</td><td>The speed segment number switching is determined by the decimal 0 to 15 corresponding to the 4-digit binary number consisting of multifunction numbers 6, 7, 8 and 9;
Enabling is valid for continuous operation;
Each speed command run time is determined only by the speed segment number switching interval.
Multi-segment enable is active along the change.</td><td> 
</td></tr>
<tr><td>Correspondence table</td><td colspan="2"><table>
<tr><td></td><td>Fun9</td><td>Fun8</td><td>Fun7</td><td>Fun6</td><td>section</td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Operation mode</th><th>Description</th><th>Running curve</th></tr>
<tr><td></td><td>between segment number and multifunction number</td><td>
<table>
<tr><td></td><td></td><td></td><td></td><td>number</td></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 Example

Multi-Speed-Single Run Control

- Parameter Settings:

Set to NiMotion speed mode 2002_h :01_h ($00B1_h$)=2;
Send: 01 10 00 B1 00 01 02 00 02
Set the speed command source to multispeed: 2006_h :02_h (0149_h)=3, 2006_h :03_h ($014A_h$)=1;
Send: 01 10 01 49 00 01 02 00 03 (set speed command source to multispeed)
Send : 01 10 01 4A 00 01 02 00 01 (set speed source to auxiliary speed command source)
Sets the multispeed function parameter group:
Multi-speed operation mode 2012_h :01_h ($02E9_h$) = 0 (single run);
Send: 01 10 02 E9 00 01 02 00 00
The operating parameters for the 1st speed:
Speed command 2012_h :$0C_h$ ($02F4_h$) = 500 (in rpm), command runtime 2012_h :$0D_h$ ($02F5_h$) = 2000 (in ms), the
Send: 01 10 02 F4 00 01 02 01 F4 (sets first segment speed to 500 rpm)
Send: 01 10 02 F5 00 01 02 07 D0 (set first speed run time to 2000ms)
Acceleration and deceleration time 2012_h :$0E_h$ ($02F6_h$) = 500 (in ms);
Send: 01 10 02 F6 00 01 02 01 F4
The operating parameters for segments 2 through 16 are set with reference to the operating parameters for segment 1 speeds;
Set motor enable: 2003_h :03_h ($00D5_h$)=1, logic active low: 2003_h :04_h ($00D6_h$)=0;
Send: 01 10 00 D5 00 01 02 00 01 (set DI1 to enable)
Send: 01 10 00 D6 00 01 02 00 00 (sets DI1 logic active low)
Setting Multi-Segment Position/Speed Enable: 2003_h :05_h ($00D7_h$)=28, Logic is active low: 2003_h :06_h ($00D8_h$)=0;
Send: 01 10 00 D7 00 01 02 00 1C (sets DI2 to multisegment position enable)
Send: 01 10 00 D8 00 01 02 00 00 (sets DI2 logic active low)

- run

Motor enable: DI1 input low;
Multi-segment position/speed enable, DI2 input low, the motor runs in accordance with the operation mode set in the 2012_h ($02E9_h$ ~0323_h) Multi-Segment Speed Function Parameter Set and the operation parameters of each segment.

6 Special Features

6.1 Digital Inputs and Outputs

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

6.1.1 Digital Inputs

Related Object Settings

Table6-1

Object Index	Description
2003_h (00D5h ~00E6)$_h$	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
$200C_h$: 07_h (0235)$_h$	Virtual digital input enable, 0-disable, 1-enable
2031_h : 01_h (0373)$_h$	Communication Given VDI Virtual Levels
$200B_h$: 05_h (01E2)$_h$	Actual level status corresponding to entity terminals (bit0-DI1~bit3-DI4)

control command

This device has up to 3 digital input entity terminals: fixed input DI1~3;

Communication virtual input terminals, set for use via the 2017_h (0326h ~ 0346h) parameter set.2031_h : 01_h (0373_h) sets the state of the corresponding level of the virtual terminal for communication (bit0-VDI1 ~ bit15-VDI16).

6.1.1.1 Physical Terminal

Special Function Options

The specific values are taken from the table below:

Table6-2

function number	functional definition
0	undefined
1	enable
2	Alarm reset
6	Multi-segment operation command switching1
7	Multi-segment operation command switching 2
8	Multi-segment operation command switching3
9	Multi-segment operation command switching4
12	pause (media player)
14	Positive limit switch
15	Reverse limit switch
20	Step amount enable
21	Nimotion Speed Mode RPM Direction Commands
28	Multi-segment position/speed enable
31	Home switch
33	Setting the origin
36	Analog input position/speed control direction
38	Clearing Fault History
39	Clear power-up time
40	Quadrature pulse input A (fixed DI1)
41	Quadrature pulse input B (fixed DI2)
42	Pulse input (fixed DI1)
43	Pulse input direction (fixed DI2)
44	Duty cycle input (fixed DI1)
45	Duty cycle input direction (fixed DI2)

Terminal Logic Selection

Multi-function terminal logic selection is used through the 2003_h (00D5h ~$00E6_h$) (Entity Input Terminal) parameter group setting. The specific values are shown in the table below:

Table6-3

Set Value	Logical Definition
0	Active Low

1	Active High
2	Rising edge valid
3	Effective along the descending edge
4	Valid on both rising and falling edges

Notes:

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

2, different terminals can not be set the same function number, if set incorrectly will report parameter error fault.

3. If the working mode is set to HM mode of CiA402, without setting 14, 15 and 31 function numbers will also report parameter failure.

4. Terminal function and logic selection are set to shutdown effective.

5. The digital inputs are detected only once every millisecond. Cannot handle signal changes of less than 1 ms at the input.

example

The following configuration of the entity input terminal DI1 is 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
- Logic selection is active low, 2003_h : 04_h ($00D6_h$) =0
  Send: 01 10 00 D6 00 01 02 00 00

6.1.1.2 Virtual Terminal

Virtual VDI Terminal Special Function Definition

The specific values are taken from the table below:

Table6-4

Function Number	Functional Definition
0	undefined
1	enable
2	Alarm reset
6	Multi-segment operation command switching1
7	Multi-segment operation command switching 2
8	Multi-segment operation command switching3
9	Multi-segment operation command switching4
12	pause (media player)
14	Positive limit switch
15	Reverse limit switch
20	Step amount enable

Function Number	Functional Definition
21	Nimotion Speed Mode RPM Direction Commands
28	Multi-segment position/speed enable
31	Home switch
33	Setting the zero point
36	Analog input position/speed control direction
38	Clearing Fault History
39	Clear power-up time

Notes:

1、"1-Enable""""12-Suspend "is only valid in NiMotion mode, which is equivalent to writing enable control word to 6040_h (0380_h) in 402 mode.

Virtual VDI Terminal Logic Selection

Multi-function terminal logic selection is used through the 2017_h (0326_h ~0346_h) (Virtual Input Terminal) parameter group setting. The specific values are shown in the table below:

Table6-5

Preset Value	Logic Definition
0	Active High
1	Rising edge valid

start

The Integrated Low Voltage Servo Motor Virtual VDI Terminal is functionally active when index $200C_h$: 07_h (0235_h) is 1, and a value of 0 for $200C_h$: 07_h (0235_h) disables the Virtual DI Terminal function. Enable is performed by entering the terminal function number and logic selection via the 2017_h (0326_h ~0346_h) sub-index, and when the logic selection is valid, the function of the corresponding function number is valid. Index 2031_h : 01_h (0373_h) is the state corresponding to the virtual input terminal (bit0-VDI1 ~ bit15-VDI16), which can only be given by communication, as defined below:

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

Example

The following virtual input terminal VDI1 is configured as a positive limit switch

- Configure VDI1 as a positive limit switch, 2017_h : 01_h (0326_h)=14
  Send: 01 10 03 26 00 01 02 00 0E
- Logic selection is active low, 2017_h : 02_h (0327_h) =0
  Send: 01 10 03 27 00 01 02 00 00

6.1.2 Digital Outputs

Related Object Settings

Table6-6

2004_h ($00F8_h$ ~0101_h)	Output Terminal Configuration Parameters
2031_h : 02h (0374_h)	Communication given VD0 output state (bit0~DO)

control command

This device has 2 digital output entity terminals, which need to be configured with the object 2004_h ($00F8_h$ ~0101_h) to select the output terminal function and logic level. The output status can also be given by communication through object 2031_h : 02h (0374_h).

Output terminal function selection:

Table6-7

Function Number	Function Definition
0	Configure DX1 as the input
1	Configure DX1 as an output
2	Motor running stop
3	target met
4	Alarm output
5	Brake output
6	External braking resistor function
30	Brake power saving drive output

Note: One of the DO ports is dedicated to the external brake resistor function and the brake power saving drive output function;

Output terminal logic selection:

Table6-8

Set Value	Logical Definition
0	Active Low
1	Active High

give an example

Example 1: Configure DO1 as a normal DO function and output the DO1 port to a conductive state:

- Configure DO1 as an output, 2004_h : 01_h ($00F8_h$) = 1;
  Send: 01 10 00 F8 00 01 02 00 01
- Configure DO1 logic to be active high, 2004_h : 02_h ($00F9_h$) = 1;
  Send: 01 10 00 F9 00 01 02 00 01
- Communication is given that DO1 output is on, 2031_h : 02_h (0374_h) = 1, at this time DO1 to GND output state on.
  Send: 01 10 03 74 00 01 02 00 01

6.2 Analog Inputs

The implementation of the analog control mode requires voltage input to the analog input port, and the voltage input range is 0-10V; through the configuration parameters, the range of controlled quantity corresponding to 0-10V is set; the analog input control includes analog speed control, analog torque

control and analog position control;

6.2.1 Analog position control

Table6-9

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-band judgment time window, unit 1ms
2005_h :$2C_h$ (0142)$_h$	Analog 10V corresponding position value (encoder unit)
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h :17_h ($00E9$)$_h$	AI1 bias in 1mV
2003_h :18_h ($00EA$)$_h$	AI1 input filter time constant in 1ms
2003_h :19_h ($00EB$)$_h$	AI1 Dead-band in 0.01mV
2003_h :$1A_h$ ($00EC$)$_h$	AI1 Magnification in 0.001

Enable

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

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

1. Location Source Selection:

   Set Position Command Source 2005_h :01_h ($010D_h$), Meaning: 3-Mode 1 analog position control, 4-Mode 2 analog position control;

2. Setting the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on 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, the corresponding input terminal DI1 or DI2 needs to be set for analog input position control direction function (multifunction number 36).

3. Position Setting

   Configure the analog 10V to correspond to the position value 2005_h :$2C_h$ (0142_h), which is the absolute motor position.

4. run

   The motor is enabled via a terminal, and the motor operates in the corresponding position according to the analog input.

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

Analog position control mode

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

1. Model 1:
   Given position = Symbol (multifunction number 36 level corresponds to) * Analog 10V corresponds to position value * (Input voltage * AI1 multiplier + AI1 bias)/10000, multifunction number 36 is valid for negative direction, invalid for positive direction.

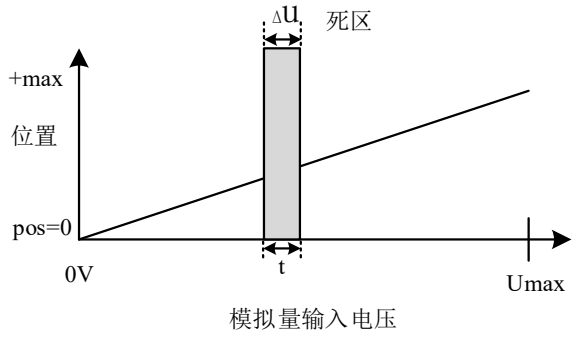


Figure6-1

2. Mode 2:
   Input voltage 5~10V: given position value = Symbol (multifunction number 36 level corresponds to) * Analog 10V corresponding position value * (Input voltage * AI1 multiplier + AI1 bias)/10000;

   Input voltage 0~5V: given position value = -symbol (multifunction number 36 level corresponds) * analog 10V corresponding position value * (input voltage * AI1 multiplier + AI1 bias)/10000;

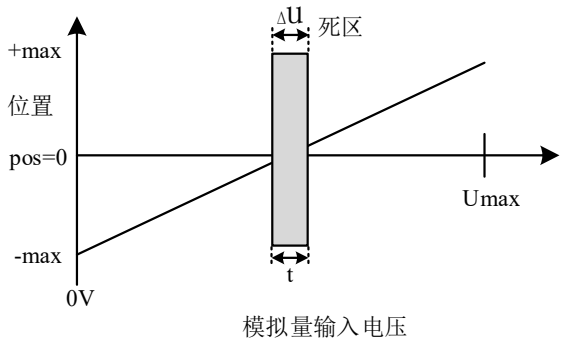


Figure6-2

Note: The given position before dead-band is the analog voltage (△u/t) after input filtering and multiplicative bias in the AI1 dead-band judgment time window 2005_h :$2A_h$ (0140_h) within the change of continuous greater than the AI1 dead-band 2003_h :19_h ($00EB_h$) setting value, otherwise the given position maintains the value of the previous moment, the setting can eliminate the position jitter brought about by the jitter of the analog voltage.

Configuration Example

- Parameters setting

  Set to NiMotion position mode: 2002_h :01_h ($00B1_h$)=1;
  Send: 01 10 00 B1 00 01 02 00 01

Set the location command source to 3: 2005_h :01_h ($010D_h$)=3;
Send: 01 10 01 0D 00 01 02 00 03
Set motor enable: 2003_h :03_h ($00D5_h$)=1, logic active low: 2003_h :04_h ($00D6_h$)=0;
Send: 01 10 00 D5 00 01 02 00 01 (set DI1 to enable)
Send: 01 10 00 D6 00 01 02 00 00 (sets DI1 logic active low)
Set the analog input position control direction: 2003_h :05_h ($00D7_h$)=36, Logic is falling edge valid: 2003_h :06_h ($00D8_h$)=3;
Send:01 10 00 D7 00 01 02 00 24 (set DI2 to analog input position control direction 36)
Send:01 10 00 D8 00 01 02 00 03 (sets DI2 logic to be active on falling edge)
2005_h :$2C_h$ (0142_h)=10000;(position value corresponding to 10V)
Send: 01 10 01 42 00 02 04 00 00 27 10
The AI interface is connected to the analog voltage value (be careful not to reverse the positive and negative), and the position is given by adjusting the voltage;

- run

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

6.2.2 Analog speed control

relevant object

Table6-10

Object Index	Description
2002_h :01_h ($00B1$)$_h$	Operation mode selection
2006_h :01_h (0148) $_h$	NiMotion speed mode master speed feed source, 1-mode 1 analog speed control
2006_h :02_h (0149)$_h$	NiMotion speed mode auxiliary speed feed source, 1-mode 2 analog speed control
2006_h :03_h ($014A$)$_h$	NiMotion speed mode speed command selection, 0 - primary speed given, 1 - secondary speed given
2006_h :04_h ($014B$)$_h$	Speed command numerical given value in rpm
2006_h :07_h ($014E$)$_h$	The speed command acceleration ramp time constant, in ms, indicates the time taken to accelerate from 0 rpm to 1000 rpm.
2006_h :08_h ($014F$)$_h$	Speed command deceleration ramp time constant in ms
2006_h :$0A_h$ (0151)$_h$	Forward speed threshold in rpm
2006_h :$0B_h$ (0152)$_h$	2006_h :$0B_h$: Forward speed threshold, in rpm
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h :17_h ($00E9$)$_h$	AI1 bias in 1mV
2003_h :18_h ($00EA$)$_h$	AI1 input filter time constant in 0.01ms
2003_h :19_h ($00EB$)$_h$	AI1 Dead-band in 1mV
2003_h :$1A_h$ ($00EC$)$_h$	AI1 Magnification in 0.001
2003_h :$1F_h$ ($00F1$)$_h$	Analog 10V corresponds to the speed value in rpm

Enable

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

Operating prerequisites: The integrated low-voltage servomotor is switched to NiMotion speed mode (2002_h :01_h ($00B1_h$) set to 2).

1. Speed source selection:

Set the primary speed command source 2006_h :01_h (0148_h) and the secondary speed command source 2006_h :02_h (0149_h) to 1. Meaning: 1 - Analog speed regulation;

By means of 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 given, 1 - auxiliary speed given;

2. Set the multi-function terminals (both physical and virtual terminals are applicable, see "7.1 Digital Inputs and Outputs" for details on 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 ($014A_h$) set in the previous step is 0, you need to set the corresponding input terminal DI1 or DI2 for the analog input speed control direction function (multifunction number 36);

3. Velocity, acceleration and maximum speed settings

If the speed command selection 2006_h :03_h ($014A_h$) is 0, you need to configure the analog 10V corresponding to the speed value 2003_h :$1F_h$ ($00F1_h$);

If the speed command selection 2006_h :03_h ($014A_h$) is 1, the forward and reverse speed thresholds 2006_h :$0A_h$ (0151_h) and 2006_h :$0B_h$ (0152_h) need to be configured;

Configure the plus and minus speeds 2006_h :07_h ($014E_h$) and 2006_h :08_h ($014F_h$);

4. run

The motor is enabled via a terminal and runs at the corresponding speed according to the analog input.

Analog speed control mode

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

1. Model 1:

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

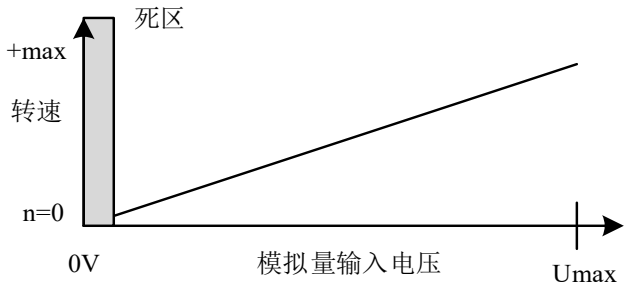


Figure6-3

2. Mode 2:

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

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

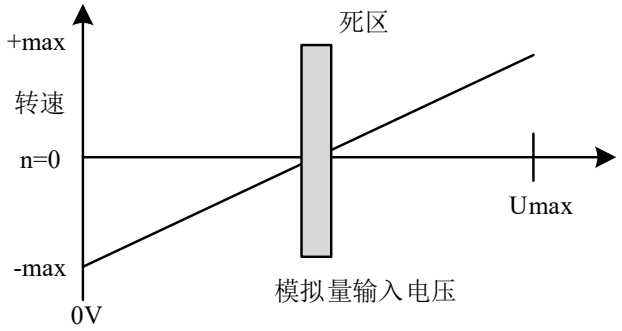


Figure6-4

Configuration Example

- Parameters setting

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

Send: 01 10 00 B1 00 01 02 00 02

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

Send: 01 10 01 48 00 01 02 00 01 (set primary speed source to 1)

Send: 01 10 01 4A 00 01 02 00 00 (set speed command selection as main speed command source)

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

Send: 01 10 01 4E 00 01 02 00 64

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

Send: 01 10 01 4F 00 01 02 00 64

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

Send: 01 10 00 F1 00 01 02 0B B8

Set the DI1 and DI2 terminals:

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

Send: 01 10 00 D5 00 01 02 00 01

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

Send: 01 10 00 D6 00 01 02 00 00

DI2 function selected for analog input speed control direction: 2003_h :05_h ($00D7_h$) = 36;

Send: 01 10 00 D7 00 01 02 00 24

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

Send: 01 10 00 D8 00 01 02 00 03

The AI interface accesses the analog voltage value (be careful not to reverse the positive and negative), which is used to adjust the voltage speed regulation;

- Running:

  DI1 inputs low, the motor runs at the speed given by the analog 2006_h :06_h ($014D_h$), and DI2 controls the direction of motor operation.

6.2.3 Analog torque control

Table6-11

Object Index	Description
2002_h :01_h ($00B1$)$_h$	Operation mode selection
2007_h :01_h ($015E$)$_h$	NiMotion torque mode main torque feed source, 1 - Mode 1 analog torque control;
2007_h :02_h ($015F$)$_h$	NiMotion torque mode auxiliary torque feed source, 1-mode 2 analog torque control;
2007_h :03_h (0160)$_h$	NiMotion torque mode torque command selection, 0 - primary torque given, 1 - secondary torque given;
2007_h :04_h (0161)$_h$	Torque command digital given value (0.1%)
2007_h :$0A_h$ (0167)$_h$	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 ($016E$)$_h$	2007_h :11_h : Torque control reverse speed limit value (rpm)
2017_h (0326_h ~0346)$_h$	Functional configuration of the virtual input terminals (see: "6.1.1 Digital input")
2003_h :17_h ($00E9$)$_h$	AI1 bias in 1mV
2003_h :18_h ($00EA$)$_h$	AI1 input filter time constant in 0.01ms
2003_h :19_h ($00EB$)$_h$	AI1 Deadband in 1mV
2003_h :$1A_h$ ($00EC$)$_h$	AI1 Magnification in 0.001
2003_h :20_h ($00F2$)$_h$	Analog 10V corresponds to the torque value, unit 0.1%.

Enable

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

Operating prerequisites: The integrated low-voltage servomotor is switched to NiMotion torque mode (2002_h :01_h ($00B1_h$) set to 3).

1. Torque source selection:

   Set the primary torque command source 2007_h :01_h ($015E_h$) and the secondary torque command source 2007_h :02_h ($015F_h$) to 1. Meaning: 1 - Analog torque control;

   By means of the torque command selection 2007_h :03_h (0160_h) the actual effective is either a

main torque command (mode 1 analog torque control) or an auxiliary torque command (mode 2 analog torque control), meaning: 0 - main torque given, 1 - auxiliary torque given;

2. Set the multi-function terminals (both physical and virtual terminals are applicable, see "6.1 Digital Inputs and Outputs" for details on 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 for analog input torque control direction function (multifunction number 36);

3. Torque setting and torque and speed limit settings

If the torque command selection 2007_h :03_h (0160_h) is 0, you need to configure the analog 10V corresponding to the torque value 2003_h :20_h ($00F2_h$);

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 via one terminal and the motor is controlled according to the analog input.

Analog torque control mode

The analog input torque control has two modes: mode 1 and mode 2.

1. Model 1:

Given torque = Symbol (Multi-function No. 36 level corresponds to) * Analog 10V corresponds to torque value * (Input voltage * AI1 multiplier + AI1 bias)/10000, Note: Multi-function No. 36 is valid for negative direction, invalid for positive direction.

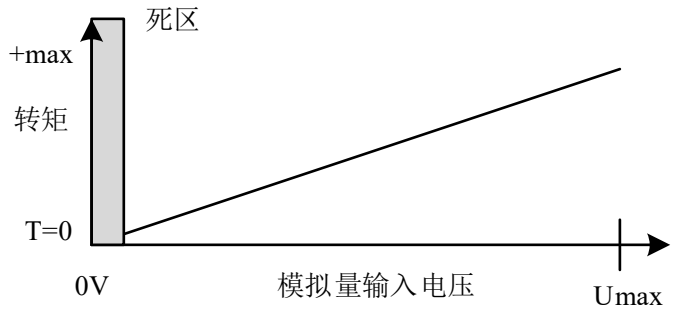


Figure6-5

2. Mode 2:

Input voltage 5~10V: given torque = Symbol (multifunction number 36 level corresponds to) * Analog 10V corresponding torque value * (Input voltage * AI1 multiplier + AI1 bias)/10000;

Input voltage 0~5V: given torque = -symbol (multifunction number 36 level corresponds to) * analog 10V corresponds to torque value * (input voltage * AI1 multiplier + AI1 bias)/10000;

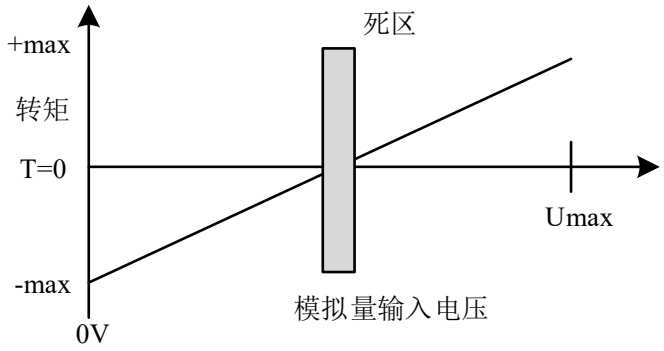


Figure6-6

Configuration Example

- Parameters setting

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

  Send: 01 10 00 B1 00 01 02 00 03

  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

  Setting torque command selection is for the main torque command: 2007_h :03_h (0160_h)=0;

  Send: 01 10 01 60 00 01 02 00 00

  Set the positive and negative internal torque limits to 100%: 2007_h :$0A_h$ (0167_h)=1000 and 2007_h :$0B_h$ (0168_h)=1000 (unit 0.1%);

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

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

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

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

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

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

  Send: 01 10 00 F2 00 01 02 03 E8

  Set the DI1 and DI2 terminals:

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

  Send: 01 10 00 D5 00 01 02 00 01

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

  Send: 01 10 00 D6 00 01 02 00 00

  Set DI2 function selection to analog input position control direction: 2003_h :05_h ($00D7_h$)=36, logic to falling edge valid: 2003_h :06_h ($00D8_h$)=3;

  Send: 01 10 00 D7 00 01 02 00 24 (set DI2 to analog input position control direction 36)

  Send: 01 10 00 D8 00 01 02 00 03 (sets DI2 logic to be active on falling edge)

- run

  DI1 input is low, the motor runs according to the torque given by the analog 2007_h :04_h (0161_h), and DI2 controls the direction of motor operation.

6.3 Holding Brake Settings

A holding brake is used to prevent the motor shaft from moving when the motor is in a non-enabling state, thus locking the motor in position. The holding brake is categorized into two ways: external holding brake and internal holding brake. The integrated motor automatically switches the state of the holding brake output according to the CiA402 state machine as shown below:

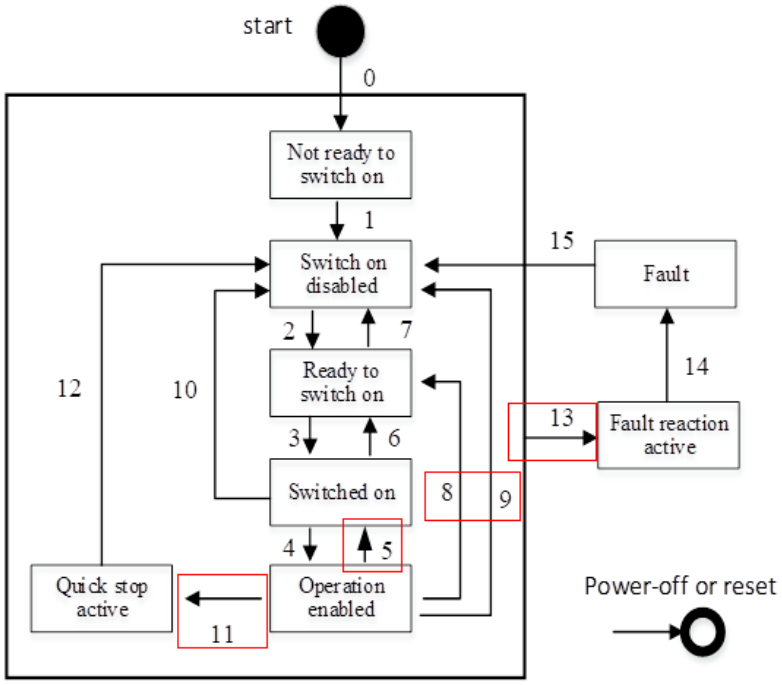


Figure6-7

Note: When the state machine leaves the operation state (Operation enable), the holding brake output is closed.

Table6-12

Bit	Name	Description
0	Motor ready.	When both status words Bit0~ bit3 are 1, the holding brake output is turned on.
1	Turn on the main circuit power	
2	Quick Stop	
3	Motor operation	

Table6-13

Object Index	Description
2002_h : $0A_h$ (00BA)$_h$	Holding brake output off to motor not energized delay time, unit: ms
$200A_h$: $1F_h$ (01D8)$_h$	Brake detection enable, 0-no gravity preset, 1-enable gravity preset
$200A_h$: 20_h (01D9)$_h$	Holding brake load gravity (gravity preset value, when the holding brake is open, the electric machine output torque), 0.1% rated torque
$200D_h$: $0B_h$ (0254)$_h$	Holding brake load recognition, 0-does not enable gravity recognition, 1-enables in the linear gravity recognition

Notes:

1. When the motor is used in vertical axis, it is necessary to turn on the gravity preset and online gravity recognition function, through the gravity preset value to prevent the load from self-weight or external force caused by the load.

into the holding brake release and load drop during motor startup.

2. when the motor is used for vertical axis, it is necessary to set the holding brake output to motor not energized delay time appropriately, to prevent the motor from running to stop or fault stopping when the

Load drops due to self-weight or external forces.

3. Enable online real-time identification of load gravity and automatically update the gravity preset value $200A_h$:20_h ($01D9_h$) when the motor speed is below 300rpm.

Enable and disable

a) The holding brake output control can be enabled via the object dictionary 2004_h :01_h ($00F8_h$) and the holding brake function is automatically enabled when its function number is set to 30; When the terminal function 2004_h :01_h ($00F8_h$) of DX1 is configured with multifunction No. 5 (motor holding state output) and the logic is active high, DX1 outputs a high level when the holding brake is opened.

b) The hold output control can be switched off using the object dictionary 2004_h :01_h ($00F8_h$), and the hold function is automatically switched off when its function number is set to 0. When the terminal function 2004_h :01_h ($00F8_h$) of DX1 is configured with multifunction No. 5 (motor holding state output) and the logic is active high, DX1 outputs a low level when the holding brake is closed.

duty cycle

The brake outputs trigger a PWM signal with adjustable duty cycle. The duty cycle (ratio of pulse time to cycle time) is set in 2002_h :04_h ($00B4_h$) and is a percentage value that can be selected between 1 and 100 (a configuration above 90 is recommended). When the value is 100, the brake output will be on continuously.

6.4 I^2t Overload Protection of Motor

After an integrated low-voltage servo motor is energized, it will continuously 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 heat released, the motor temperature rises, and an excessively high temperature will cause the motor to burn out.I^2t The purpose of the motor overload protection is to ensure that the motor operates normally under the temperature limit and to prevent damage to the motor.

relevant object

Table6 -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: 0.1s)

trigger mechanism

To enable this protection function, the three object items mentioned above must be correctly specified. When the motor is running above the rated current, the drive detects and calculates the time integral of I^2 When the integral value is greater than the product of the set maximum current 2000_h : $0A_h$ (0066_h) and the maximum current duration 2000_h : $0B_h$ (0067_h), the motor reports an overload warning and restricts the torque output to the rated value. If these conditions are not met, the I^2 t overload protection function will not be triggered.

6.5 Brake Settings

When the torque and speed of the motor are in the opposite direction, the energy is transferred back from the motor end to the driver of the servo motor, which makes the bus voltage value rise, and when it rises to the braking point, the energy needs to be consumed through the braking resistor, otherwise the driver of the servo motor will be damaged. The braking resistor of the integrated servo motor is divided into two types: built-in and external (2002_h : 13_h ($00C3_h$) settings, 0~internal (factory default), 1~external, valid for power-on again, factory default 55V trigger), but cannot be used at the same time. The built-in braking resistor specifications are shown in the table below:

Table 6- 15

Capacity	Servo Motor Model	Built-in braking resistor specifications	Handling power
100W	PMM4010B	50Ω, 10W	3W
200W	PMM6020B	15Ω, 25W	8W
400W	PMM6040B	15Ω, 25W	8W
750W	PMM8075B	15Ω, 35W	11W

6.5.1 Torque without external loads

Motor to do back and forth reciprocating action, brake kinetic energy will be converted into electrical energy back to the bus capacitor, to be the bus voltage over the braking voltage 2001_h :$0F_h$ (0093_h), the braking resistor will consume the excess feed-back energy. The following table shows the braking energy E generated by the servomotor from 3000rpm to standstill :

Table 6- 16

power (output)	Servo motor model	Rotor inertia (10^{-4} kgm^2)	Braking energy E (J)
200W	PMM6020B	0.19	0.94
400W	PMM6040B	0.3	1.48
750W	PMM8075B	1.2	5.94

The selection of braking resistor power P is related to the period of reciprocating motion T (unit S), motor speed and the ratio of inertia of the load to the motor N. The specific calculation formula is as follows:

$$P = \frac{2\times(N+1)\times E}{T}$$

The specific speed profile of the servo motor from 3000rpm at no load to standstill is as follows:

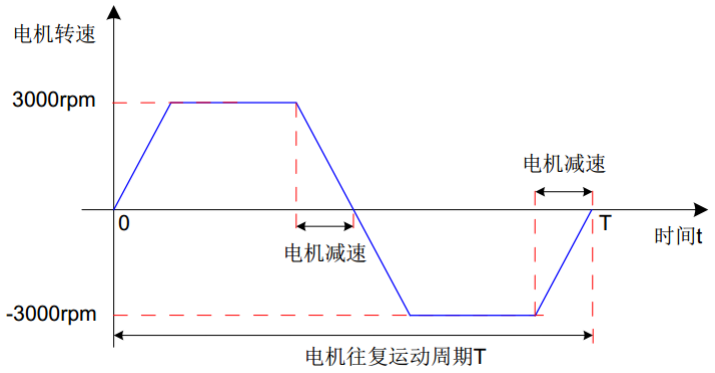


Figure 6-8

Taking 200W capacity as an example, assuming the reciprocating motion period T=2s, the maximum speed of 3000rpm, and the load inertia is 10 times of the motor inertia, the braking resistor power is required:

$$Pa = \frac{2\times(N+1)\times E}{T} = \frac{2\times(10+1)\times 0.94}{2} = 10.34W$$

Pa is less than the power 12.5W that can be handled by the built-in braking resistor, so the requirement can be met by using the built-in braking resistor.

If the load inertia in the above assumptions is changed from 10 times to 30 times, and other conditions remain unchanged, the braking resistor power Pb is required:

$$Pb = \frac{2\times(N+1)\times E}{T} = \frac{2\times(30+1)\times 0.94}{2} = 29.14W$$

Pb is greater than the power that can be handled by the built-in braking resistor, 12.5 W. Therefore, an external braking resistor is required. When using an external braking resistor, it is necessary to use 30% of the rated power, i.e., Pb/(1-70%) = 97.13W.

6.5.2 With external load torque

The direction of rotation of the motor is the same as the direction of rotation, and the motor outputs energy to the outside. However, there are some special occasions when the motor torque output is opposite to the direction of rotation, at which time the motor performs negative work, and the external energy is fed back to the driver through the motor to generate electrical energy.

Taking the 400W (rated torque 1.27Nm) capacity as an example, when the external load torque is 60% of the rated torque and the rotation speed reaches 1,000rpm, the power fed back to the servo motor driver is calculated as follows:

$$P = \frac{2\pi\times n\times M}{60} \times 60\% = \frac{2\pi\times 1000\times 1.27}{60} \times 60\% = 79.76W$$

Considering that the braking resistor needs to be used at 30% of the rated value, the power of the external braking resistor $P_{实际}$ is 79.76/(1-70%) = 265.85W, with a resistance value of 15Ω.

Note: n is rotational speed (rpm), M is torque (N.m)

Note: This product is not suitable for use as a generator unless the DC side power supply allows energy return.

6.6 Parameter Saving and Restoration

Parameter management is written according to the CiA301 standard, 1010_h (0026_h ~$003A_h$) is the save parameter function object, 1011_h ($003C_h$ ~0050_h) is the read parameter function object.

6.6.1 Parameter Saving

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

For example: 1010_h : 01_h (0026_h) write 0x65766173 for saving the parameters to the user area, 1010_h :$0B_h$ ($003A_h$) write 0x65766173 for saving the NiMotion 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 the recovery parameter command word to the corresponding sub-index of 1011_h , and a return of 1 indicates that the recovery parameter is successful, and a return of non-1 indicates that it is unsuccessful.

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

6.7 Position Recovery Function

The motor is able to regain its absolute position after being powered down and then powered back up.

By setting the position recovery method 2005_h :18_h ($012A_h$) to a value of 1, it applies to single-turn round-trip motions in which the motor is in the negative direction with respect to the zero point and the return to zero point requires positive operation; and in which the motor is in the positive direction with respect to the zero point and the return to zero point requires negative operation.

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

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

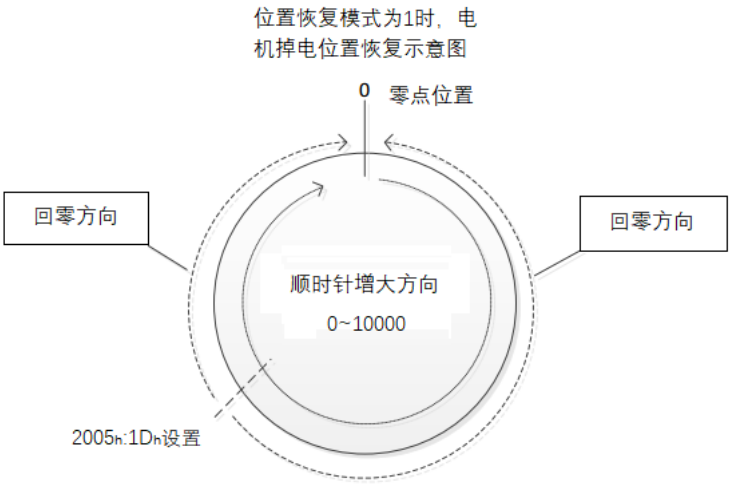


Figure6-9

When the value of the position recovery mode 2005_h :18_h ($012A_h$) is 0, which applies in multi-turn motion trajectories, the motor is able to recover the current absolute position.

6.8 Absolute Encoders

The integrated motor detects the battery voltage once per power-up and updates it in parameter $200F\text{-}0B_h$. When the integrated motor detects that the battery voltage drops below a specified value during operation, the integrated motor issues a warning, fault message.

Table -617 Related Information

Battery Voltage	warning message	Alarm Recovery	Battery Replacement Method
2.6V＞DC＞2.4V	0x7302	Multi-turn encoder counting errors may occur; battery replacement recommended	1. The integrated motor is powered up and in a non-operational state; 2. Replace the batteries in the battery mounting box; 3. The alarm is automatically disarmed.
DC≤2.4V	0x7301	Please replace the battery immediately	
DC≥2.6V	0x7309	Contact technical support to recalibrate the multiturn encoder	-

6.9 STO Safety Functions

Safe Torque Off (STO) is a safety function.STO function is triggered by the external hardware terminals STO1 and STO2 of the drive.After the STO function is triggered, the drive will turn off the input signal of the drive chip, and at the same time turn off the drive control signal to cut off the input current of the motor, preventing the drive from generating torque in the shaft end of the motor, so as to make the motor stop running.

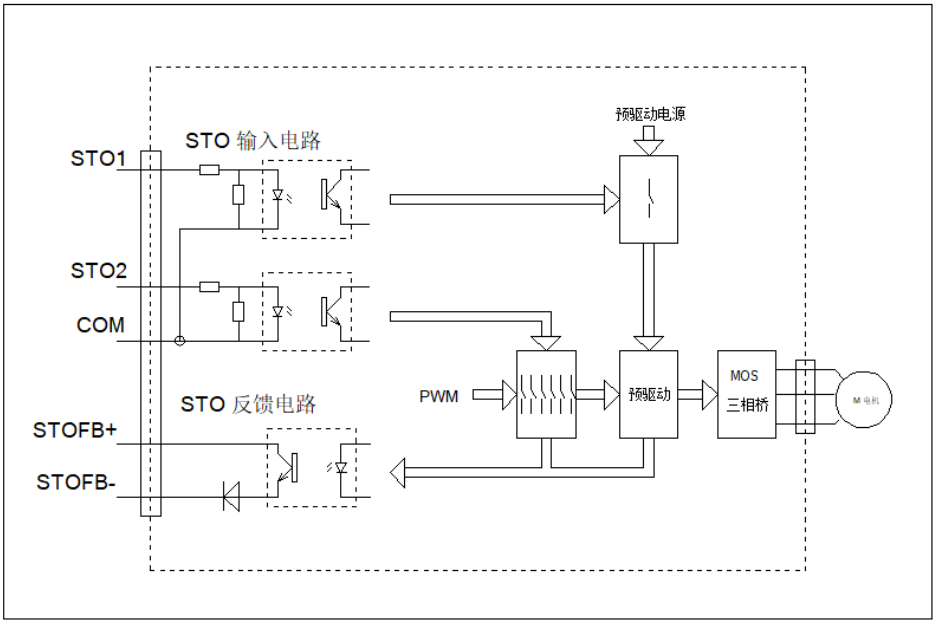


Figure6- 10 STO function wiring diagram

STO1/STO2 must be connected to the +24VDC signals at the same time so that they are at a high level to

enable normal operation of the driver. If either or both of them are placed low at the same time, then the PWM signal will be blocked.

Table 6-18 STO Function Table

STO1 Input	STO2 Input	Motor Input Current
H	H	normalcy
L	H	interdict
H	L	interdict
L	L	interdict

Table 6-19 STO (Safety Torque) Related Descriptions

Event	Clarification
Definition	Only cuts power to the motor; it does not cut power to the drive.
Description	The STO function allows the servomotor to safely enter a torque-free state and prevents accidental startup. When the STO function is activated, if the motor is running, the drive immediately blocks the output and the servomotor enters a free stop state. If the servo motor has a holding brake, the brake closes immediately and the motor stops immediately.
STO Enabled State	When at least one of the input signals of STO_1 or STO_2 becomes low, the STO enable state is entered. The power drive power supply and drive signal are cut off, and the servo enable command is invalid.
Non-Safe State	When switching from the STO enable state to the normal operation state, it will first enter an unsafe state. In this state, the power drive part will no longer work, and the STO_1/STO_2 input signals and servo enable instructions will not take effect. The servo will start self-checking. If the self-check is passed, the non-safe state will be exited and switched to the normal operating state. If the self-check fails, it will remain in a non-safe state and alarm for the detected faults. After clearing the faults, the self-check will be conducted again until it passes and the non-safe state will be exited. 2.After the servo motor is powered on, it will first enter an unsafe state. In this state, the power drive part will no longer work, and the STO_1/STO_2 input signals and servo enable instructions will not take effect. The servo will start a self-check. If the self-check is passed, it will exit the unsafe state and switch to the normal operation state or the STO enable state. If the self-check fails, it will remain in a non-safe state and alarm for the detected faults. After clearing the faults, the self-check will be conducted again until it passes and the non-safe state will be exited.

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.

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 Adjustments

7.1 Overview

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

Note: Before proceeding with the gain adjustment, it is recommended to carry out a spot test run to confirm that the motor can operate normally; the adjustment process may be accompanied by vibration of the motor, so please pay full attention to safety.

7.2 Manual Gain Adjustment

7.2.1 Relevant Object

Table7 -1

Object Index	Description
2008_h : 01_h (0178)$_h$	Velocity ring gain (unit 0.1Hz), in the case of no noise and vibration, increase this parameter, can speed up the positioning time, bring better speed stability and followability; generate noise, then reduce the parameter setting value;
2008_h : 02_h (0179)$_h$	Speed ring integral time constant (unit 0.01ms), reduce the set value can strengthen the integral effect, speed up the positioning time, but the set value is too small easy to cause mechanical vibration.
2008_h : 03_h ($017A$)$_h$	Position loop gain, increase this parameter to speed up the positioning time and improve the ability to resist external disturbances when the motor is stationary. Too high a setting may lead to system instability and oscillation.
2008_h : $0F_h$ (0186)$_h$	Speed feed-forward filtering time constant (in 0.01ms)
2008_h : 10_h (0187)$_h$	Velocity Feed Forward Gain
2008_h : 11_h (0188)$_h$	Torque feed-forward filtering time constant (in 0.01ms)
2008_h : 12_h (0189)$_h$	Torque Feed Forward Gain
2007_h : 06_h (0163)$_h$	Torque command filtering time constant to eliminate high frequency noise and suppress mechanical resonance. Increase 2008_h : 01_h (0178_h) When vibration occurs, it can be suppressed by adjusting 2007_h : 06_h (0163_h). Too large a setting will result in reduced response of the current loop

7.2.2 Basic Gain Adjustment Instructions

The integrated servo motor consists of three control loops, current loop, speed loop and position loop in order from the inside out, with the current loop having the highest response frequency. The factory default current loop gain parameter of the servo motor has ensured sufficient responsiveness and generally does not need to be adjusted, so the only things that need to be adjusted are the position

loop gain, the speed loop gain and the speed loop integration time constant. In order to ensure the stability of the system, improve the position loop gain at the same time, need to improve the speed loop gain.

The basic gain adjustment is as follows:

Figure 7-1 Velocity loop gain

Figure 7-2 Velocity loop integration time constant

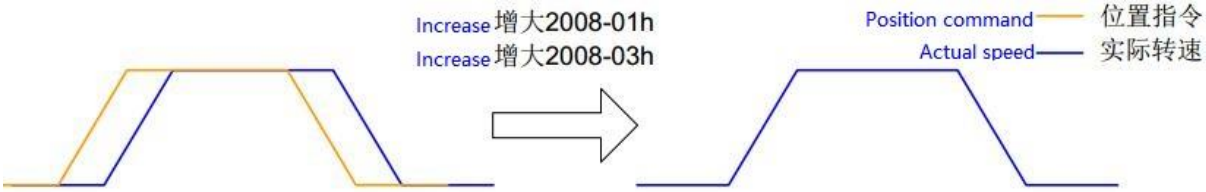


Figure 7-3 Position loop gain

Figure 7-4 Torque command filtering time constant

7.2.3 Parameter tuning based on rigid tables

Based on the rigidity table parameter adjustment refers to the rigidity table provided by the manufacturer (the following table) to set the PID parameters corresponding to different rigidity levels, to realize the purpose of quickly adjusting the stability of the integrated motor.

Table 7-2 Parameters updated by rigid table tuning mode

rigidity	2008-03h (017Ah) Position Gain (Hz)	2008-01h (0178h) Speed loop gain (0.1Hz)	2008-02h (0179h) Speed Integration Time Constant (0.01ms)
1	2	15	37000
2	3	20	28000
3	3	25	22000
4	4	30	19000
5	5	35	16000

6	6	45	12000
7	8	60	9000
8	10	75	7000
9	12	90	6000
10	14	110	5000
11	18	140	4000
12	32	180	3100
13	39	220	2500
14	48	270	2100
15	63	350	1600
16	72	400	1400
17	90	500	1200
18	108	600	1100
19	135	750	900
20	162	900	800
21	206	1150	700
22	251	1400	600
23	305	1700	500
24	377	2100	400
25	449	2500	400
26	500	2800	350
27	560	3100	300
28	610	3400	300
29	660	3700	250
30	720	4000	250
31	810	4500	200
32	900	5000	200

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

When it is necessary to adjust the parameter, it is recommended to prioritize the PID parameter with low rigidity level, and then increase the rigidity level step by step until it meets the usage requirements.

The PID parameters can be fine-tuned when the desired effect is not achieved by following the parameter settings in the rigidity table. The effect is optimized by more detailed adjustments.

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

7.2.4 Feedforward Gain Adjustment Instructions

Speed Feed Forward

Velocity feedforward improves the velocity command response and reduces the position deviation at fixed velocity, and is mainly used in the fully 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 the speed feedforward achieves an effect at a certain setting value. When adjusting, the 2008_h : $0F_h$ (0186_h) and 2008_h : 10_h (0187_h) settings should be adjusted repeatedly to find a well-balanced setting.

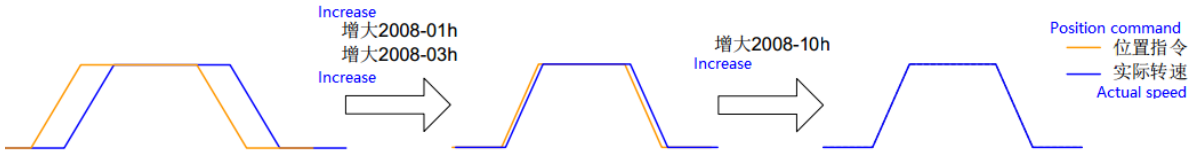


Figure 7-5 Velocity feedforward

Torque Feed Forward

Position control mode, using torque feed-forward, can improve the torque command response and reduce the position deviation at fixed acceleration and deceleration; speed control mode, using torque feed-forward, can improve the torque command response and reduce the speed deviation at fixed speed.

The basic adjustment method is as follows:

Set 2008_h ： 11_h (0188_h) to a fixed value; then, gradually increase the 2008_h ： 12_h (0189_h) setting from 0 until the speed feedforward is effective at a certain setting. When adjusting, the 2008_h : 11_h (0188_h) and 2008_h : 12_h (0189_h) settings should be adjusted repeatedly to find a well-balanced setting.

7.2.5 Gain Switching Adjustment Instructions

The gain switching function of the motor is triggered by the external DI or internal state, when the switching condition is satisfied, the motor switches to the group 1 or 2 gain according to the validity of the relevant parameter settings to meet the actual use requirements, the specific role is as follows:

- The motor running state is switched to a higher gain for better command tracking performance;
- The motor standstill is switched to a higher gain to shorten the positioning time;
- The motor quiescent (enabled) state is switched to a lower gain to suppress vibration;
- Different gain settings can be switched by external signals depending on the load equipment;

Gain Switching Flowchart

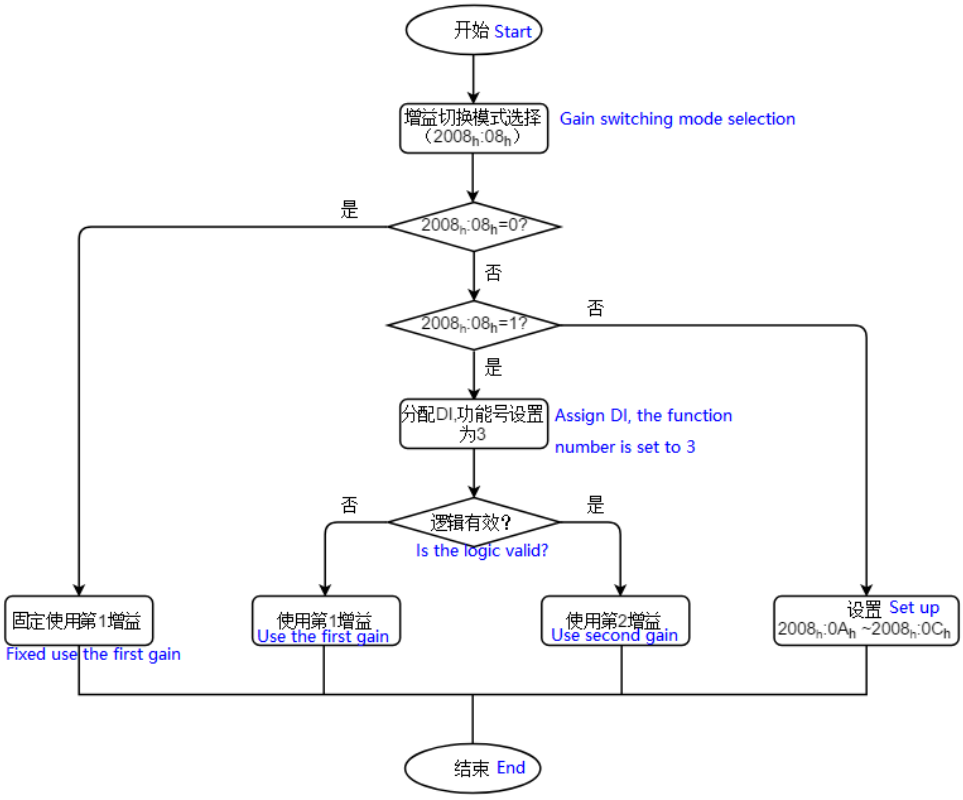


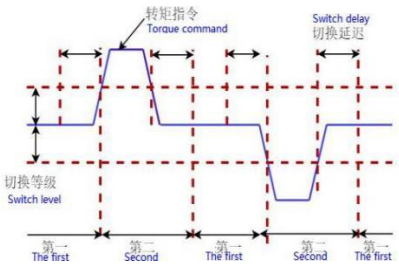
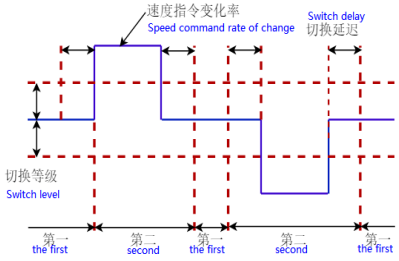
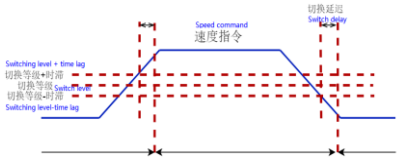
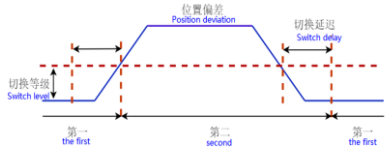
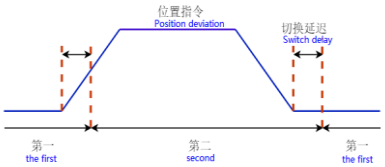
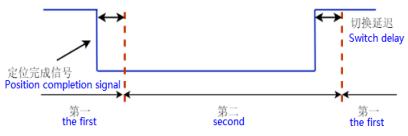
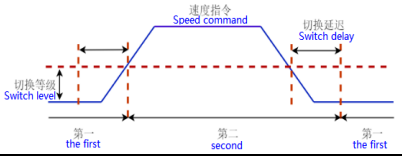
Figure7-6

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 ($017D_h$)) and there are 10 modes of switching between the different modes as shown in the following table for the schematic diagrams and the associated parameters:

Table7-1

Gain switching mode			Associated parameters		
2008_h :08_h ($017F_h$)	model name	schema	latency 2008_h :$0A_h$ (0181_h)	Toggle Level 2008_h :$0B_h$ (0182_h)	time lag of switching 2008_h :$0C_h$ (0183_h)
0	Fixed use of 1st gain	-	null	null	null
1	Switching with external DI	-	null	null	null

Gain switching mode			Associated parameters		
2	Torque command		validity (ms)	validity (%)	validity (%)
3	speed command	Internally validated speed command value (606B)h 25ms constant fixed using gain 1, change using gain 2	validity (ms)	null	null
4	Speed command rate of change		validity (ms)	validity (rpm/ms)	validity (rpm/ms)
5	speed command		validity (ms)	validity (rpm)	validity (rpm)
6	positional deviation		validity (ms)	validity (Encoder unit)	validity (Encoder unit)
7	position command		validity (ms)	null	null
8	Positioning complete.		validity (ms)	null	null
9	Actual speed		efficiently (ms)	validity (rpm)	validity (rpm)

7.3 Vibration Suppression

7.3.1 Mechanical Resonance Suppression

Servo motors may resonate near the mechanical resonance frequency while the basic gain is increased due to the inherent resonance characteristics of the mechanical system during actual use, which affects the performance of the servo motor. The trap can achieve the purpose of suppressing the resonance by reducing the gain at a specific frequency.

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

Table7-2

name (of a thing)	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 ($01A7_h$)
Trap 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 manual traps, you need to set the trap frequency to the actual center resonance frequency, and enter the width level (center resonance frequency range) and depth level of the trap, if the resonance is suppressed, it means that the traps are effective.

Trap Depth

At trap depth level 100, the input is completely suppressed at the center frequency; at trap depth level 0, the trap is ineffective. Therefore, the larger the trap depth level setting, the deeper the trap depth and the stronger the suppression of mechanical resonance, but it may lead to system instability and care should be taken when using it.

The principle of trap suppression is shown below:

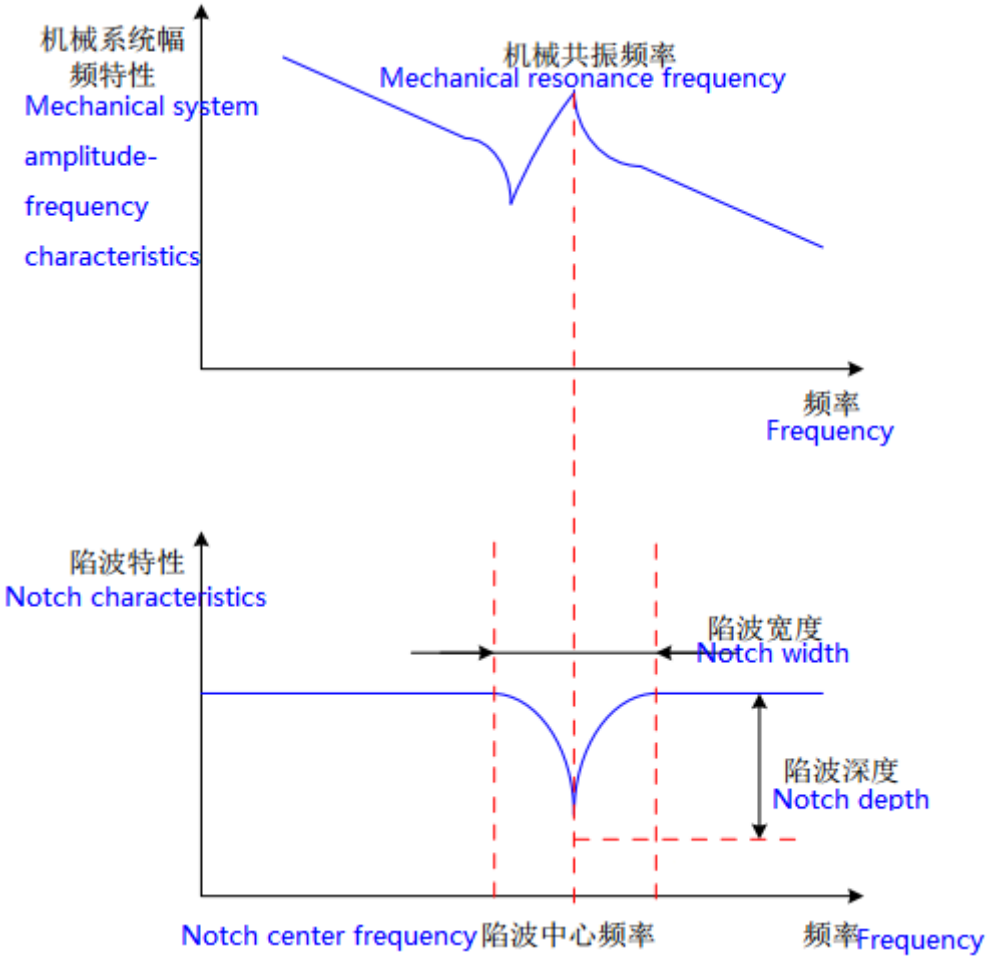


Figure7-7

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, affecting the positioning effect, and the frequency of this vibration is generally within 100Hz, which can be reduced by the low-frequency vibration suppression function.

relevant object

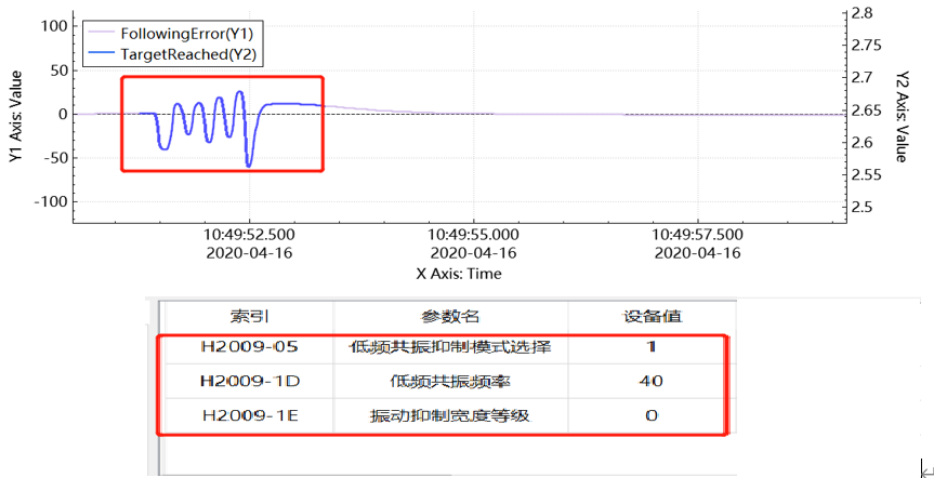
Table7-3

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, limit range 10~1000
2009_h :$1E_h$ ($01AF$)$_h$	Vibration suppression width level (unit %), set the effective frequency range of vibration suppression, 0 represents only suppression of vibration at the center frequency; set value is

object index	Description
	adjusted large, can increase the frequency range of low-frequency vibration suppression, but it will lead to longer positioning time; set value is too small, can not complete the suppression of vibration in the load vibration frequency will be changed occasions (such as belt loads), according to the actual situation of debugging settings. Limit range 0~100
2009_h :23_h ($01B5_h$)	Damping ratio coefficient, set the system vibration damping ratio coefficient, the value of the default is 5, generally do not modify, limit the range of 0 ~ 50

Setting low-frequency vibration suppression parameters

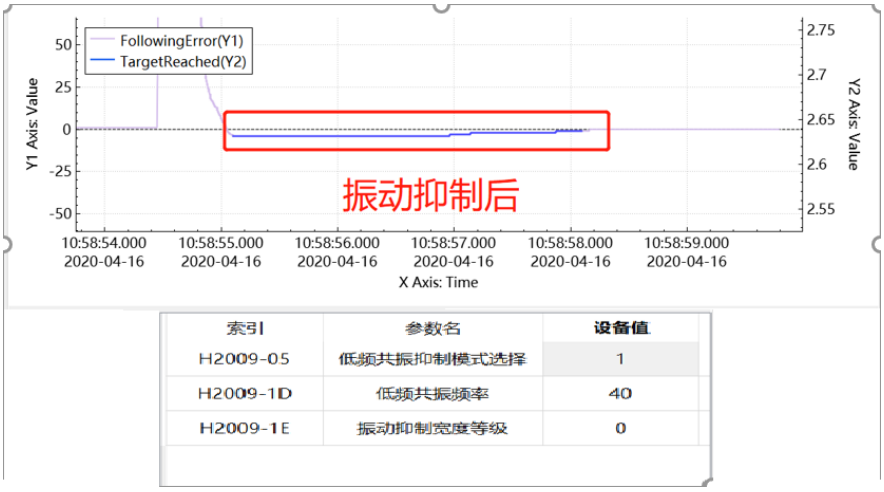
- Turn on the low-frequency vibration suppression mode, 2009_h :05_h Set 1;
  Send: 01 10 01 96 00 01 02 00 01
- The NiMStudio host computer software was used to capture the waveforms when the motor was in positional localization and to automatically calculate the frequency of the low-frequency resonance after the motion was stopped;



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

Figure7-8

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



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

Figure7-9

8 Fault Management

8.1 Alarm Code

When the integrated servo motor has a warning or fault, the motor will take the initiative to alarm (Indicator light - yellow blinking, 0.5s blinking frequency means warning, 0.25s blinking frequency means fault; Indicator light - green 0.5s blinking means communication alarm, constant light means communication is normal), and perform the corresponding action. The corresponding alarm code will be stored in objects 1003_h ： 01_h (0002_h)~1003_h ： 10_h (0020_h), and 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) that follow the FIFO rule. In the case that all the historical alarm stores are occupied, generating a new alarm will delete the earliest error that occurs, and the previous errors will move down in order. The specific alarm codes are shown in the following table:

Table8-1 List of alarm codes

Alarm Code	Alarm content	Default Alarm Type	Default Fault Response Code	Self-reset or not	$037F_h$ (error code)
0x2300	Motor overcurrent	malfunctions	0	No	0x2300
0x4012311	Motor overload	warnings	4	Yes	0x2311
0x3002312	motor stalling	Pause and lock the shaft	3	No	0x2312
0x13210	Power overvoltage	malfunctions	0	Yes	0x3210
0x1013220	Power supply undervoltage	malfunctions	1	Yes	0x3220
0x14210	Over Temperature Alarm	malfunctions	0	Yes	0x4210
0x14220	Low temperature alarm	malfunctions	0	Yes	0x4220
0x5080	Drive failure*	malfunctions	0	No	0x5080
0x5540	Flash operation malfunction*	malfunctions	0	No	0x5540
0x5541	Flash initialization failure*	malfunctions	0	No	0x5541
0x4005542	Flash Checksum Error Warning*	warnings	4	No	0x5542
0x5543	Flash user area without parameters*	malfunctions	0	No	0x5543
0x4005544	Abnormal power-down position storage	warnings	4	No	0x5544
0x4005545	Power-down save data not stored	warnings	4	No	0x5545
0x6000	Hardware initialization failure*	malfunctions	0	No	0x6000

Alarm Code	Alarm content	Default Alarm Type	Default Fault Response Code	Self-reset or not	037F$_h$ (error code)
0x16320	Parameter setting error	malfunctions	0	Yes	0x6320
0x6321	Registration failures*	malfunctions	0	No	0x6321
0x7301	Encoder Battery Low Voltage Failure	malfunctions	0	No	0x7301
0x4017302	Encoder battery undervoltage alarm	warnings	4	Yes	0x7302
0x1017304	Encoder multiturn count error	warnings	1	Yes	0x7304
0x7305	Z-pulse fault*	malfunctions	0	No	0x7305
0x7306	Encoder malfunction*	malfunctions	0	No	0x7306
0x4017307	Encoder warning*	warnings	4	Yes	0x7307
0x1007309	Encoder battery failure fault	Fault	1	No	0x7309
0x17310	speeding	malfunctions	0	Yes	0x7310
0x3017501	Illegal Function Codes in Modbus Communication	warnings	3	Yes	0x7501
0x4017502	Illegal addresses in Modbus communication	warnings	4	Yes	0x7502
0x4017503	Illegal data values in Modbus communication	warnings	4	Yes	0x7503
0x4017505	Confirmation in Modbus communication	warnings	4	Yes	0x7505
0x4017506	Slave device busy in Modbus communication	warnings	4	Yes	0x7506
0x401750C	Synchronization message in Modbus communication requesting data data is greater than the total mapped data	warnings	4	Yes	0x750C
0x401750D	The number of synchronization messages requesting data in Modbus communication does	warnings	4	Yes	0x750D

Alarm Code	Alarm content	Default Alarm Type	Default Fault Response Code	Self-reset or not	037F$_h$ (error code)
	not correspond to the mapping data equally.				
0x401750E	Wrong node address for unicast message under synchronization function in Modbus communication	warnings	4	Yes	0x750E
0x4018311	Torque limiting protection	warnings	4	Yes	0x8311
0x8610	Home return timeout	malfunctions	0	No	0x8610
0x8611	The location is terrible.	malfunctions	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	Errors in curve planning calculations	warnings	4	Yes	0x8615
0x18616	Target position overflow	malfunctions	0	No	0x8616
0x5018617	Curve planning parameters too small	neglect	5	Yes	0x8617
0xFF01	Motor Parameter Recognition Fault*	malfunctions	0	No	0xFF01
0x0401FF02	Parameter Save Failure	warnings	4	Yes	0xFF02
0xFF05	STO_1 failed to close	Fault	0	No	0xFF05
0xFF06	STO_1 failed to enable	Fault	0	No	0xFF06
0xFF07	STO_2 failed to close	Fault	0	No	0xFF07
0xFF08	STO_2 failed to enable	Fault	0	No	0xFF08
0xFF09	STO input abnormal fault	Fault	0	No	0xFF09
0xFF0A	STO enabled state	Fault	0	No	0xFF0A
0xFF0B	Non-safe state	Fault	0	No	0xFF0B
0xFF0C	Encoder calibration timeout	Fault	0	No	0xFF0C

Definition of the corresponding bit of the alarm code

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Malfunction 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 Codes
  - 0- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if 605E>=0, then follow the fault shutdown mode selected by 605E; otherwise: the power tubes are fully turned off, free shutdown.
  - 1- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if 605E>=0, then the fault shutdown mode is selected according to $605E_h$ ($03C1_h$); otherwise: the output is shut down according to the slow curve stop.
  - 2- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if 605E>=0, then the fault shutdown mode is selected according to $605E_h$ ($03C1_h$); otherwise: the outputs are shut down according to the fast curve stop.
  - 3- Pressing the quick curve suspends the locking axis without switching to the fault state and is not affected by object $605E_h$ ($03C1_h$).
  - 4- Warning and does not change the current operating state, does not switch to faulty state and is not affected by object $605E_h$ ($03C1_h$).
  - 5- Ignored and logged in 1003_h (0002_h ~0020_h), not switched to the fault state and not affected by object $605E_h$ ($03C1_h$).
  - 6- Ignored but not logged in 1003_h (0002_h ~0020_h), not switched to fault state and not affected by object $605E_h$ ($03C1_h$).
- Self-reset allowed
  - 0- impermissible
  - 1- Allowed (reset automatically when warning or fault is removed)

Notes:

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

8.2 Fault Action Setting

Alarm code consists of fault code (16 bits), fault response code (8 bits) and self-reset allow (8 bits) totaling 32 bits. Fault code indicates that the motor in the abnormal state of the fault phenomenon, through the fault code to determine the current state of the motor, generally and the object $603F_h$ ($037F_h$) (error code) corresponds to 0 indicates that the motor is currently no fault; fault response code indicates

that the motor after the failure of the corresponding mode of action and the treatment of the status word fault bit, such as free stop - fault state, according to a certain speed reduction stop - fault state, pause to lock the axis -The self reset is allowed to indicate whether the motor is allowed to automatically reset the fault bit of the current status word after the fault of the motor is eliminated, 1-allowed, 0-disallowed. The order of the corresponding bits is as follows:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Malfunction 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 Definitions of Fault Response Codes

0- The state machine switches to the fault state (refer to "4.1 CiA402 state machine"), if $605E_h$ ($03C1_h$) >= 0, then the fault shutdown mode selected by $605E_h$ ($03C1_h$) is followed; otherwise: power tubes are fully switched off, free shutdown.

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

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

3- Pressing the quick curve suspends the locking axis without switching to the fault state and is not affected by object $605E_h$ ($03C1_h$).

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

5- Ignored and logged in 1003_h (0002_h ~0020_h), not switched to the fault state and not affected by object $605E_h$ ($03C1_h$).

6- Ignored but not logged in 1003_h (0002_h ~0020_h), not switched to fault state and not affected by object $605E_h$ ($03C1_h$).

Alarm Code Configuration

Alarm codes that can be freely configured by the user are exemplified in fault code group $200E_h$ ($025B_h$ ~$027B_h$) as follows:

Table8-2

Fault Code Group $200E_h$ ($025B_h$ ~$027B$)$_h$	**hidden meaning**	**Default Failure Action**
0x4012311	Motor overload	No
0x3002312	motor stalling	No
0x13210	Power overvoltage	Yes
0x1013220	Power supply undervoltage	Yes
0x14210	Over Temperature Alarm	Yes
0x14220	Low temperature alarm	Yes
0x16320	Parameter setting error	Yes
0x17310	speeding	Yes

Fault Code Group 200E$_h$ (025B$_h$ ~027B)$_h$	hidden meaning	Default Failure Action
0x17501	Illegal Function Codes in Modbus Communication	No
0x17502	Illegal addresses in Modbus communication	No
0x17503	Illegal data values in Modbus communication	No
0x17505	Confirmation in Modbus communication	No
0x17506	Slave device busy in Modbus communication	No
0x1750C	Synchronization message in Modbus communication requesting data data is greater than the total mapped data	No
0x1750D	The number of synchronization messages requesting data in Modbus communication does not correspond to the mapping data equally.	No
0x1750E	Wrong node address for unicast message under synchronization function in Modbus communication	No
0x8610	Home return timeout	Yes
0x8611	The location is terrible.	Yes
0x3018613	Software limit error	No
0x3018614	Limit switch error	No
0x3285	Output phase failure	Yes
0x4018615	Errors in curve planning calculations	No
0x18616	Target position overflow	Yes
0x5018617	Curve planning parameters too small	No

Note: The "Yes (No)" for the fault action in the above list is an indication of whether or not the motor entered a fault state after this fault was generated.

Alarm Code Configuration Examples

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 is shut down according to the shutdown mode set by 605E$_h$ (03C1$_h$) (refer to the order of the corresponding bits, the fault response code corresponds to 1); when the power supply voltage is restored to normal, it is necessary to operate the motor by manual reset When the power supply voltage returns to normal, the motor needs to be reset manually before it can be operated normally (refer to the order of the corresponding bits, the self-reset permission corresponds to 0).

8.3 Fault Reset

After the integrated fault recovery, the fault codes that are not allowed by the self-reset need to be operated by manual reset operation before the motor can be operated for normal running operation. There are 3 ways of manual reset operation as follows:

- After the motor is de-enabled, the fault state is cleared by sending the control word 0x80 (control word bit7 rising edge active)
- Through the entity input terminal 2003_h (00D5h ~$00E6_h$) one of the functions is configured as the No. 2 function code (alarm reset), and set the corresponding logic selection, according to the logic selection of its setup terminal to make it effective can be reset!
- Configure one of the functions of virtual input terminal 2017_h (0326_h ~0346_h) as function code No. 2 (alarm reset) and set the corresponding logic selection, and then reset the alarm by making it valid according to the logic selection of the setup terminal.

8.4 Fault Detection Instructions

- Drive Failure Detection

  This fault is triggered by the drive chip fault signal interrupt response, triggering is reported as a drive fault.

- I^2T overload detection

  When the motor is running above the rated current, the fault detection will calculate the time integral of I^2 . 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), it will report an overload warning and limit the torque output to the rated value, and then unrestrict it after 30s.

- Blocking fault detection

  When the motor is running at a speed lower than 5 rpm and above the rated current, the fault detection will calculate the time integral of I2 and report a blocking fault and stop the output 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).

- High and low temperature fault detection

  When the drive temperature is greater than 105 ° C, report high temperature fault, below 70 ° C, the fault is lifted; when the drive temperature is lower than -25 ° C, report low temperature fault, higher than -20 ° C, the fault is lifted.

- Over-voltage and under-voltage fault detection

  An overvoltage fault is reported when the motor's supply voltage is higher than the voltage set at 2001_h : $0E_h$ (0092_h), and an undervoltage fault is reported when the motor's supply voltage is lower than the voltage set at 2001_h : 10_h (0094_h).

- Parameter setting fault detection

  Parameter Setting Fault Detection Errors in the setting of the multifunction terminals, including the same function number being set for different terminals (physical and virtual) or the 14, 15 and 31 function numbers not being set for the CiA402 home return mode.

- Over-speed fault detection

  When the motor speed is higher than the speed set by $200A_h$: 06_h ($01BC_h$) and maintained for 10ms, an overspeed fault is reported, and lower than that, the fault state is released.

- Illegal address fault detection in Modbus communication

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

- Illegal data value fault detection in Modbus communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Illegal Function Code Fault Detection in Modbus Communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Fault Detection of Unicast Message Node Address Error under Synchronization Function in Modbus Communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Slave device busy fault detection in Modbus communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Synchronization message in Modbus communication requesting data data greater than mapped total data fault detection
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Fault detection of unequal mapping data corresponding to the number of data requested in synchronization messages in Modbus communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Fault Detection of Unicast Message Node Address Error under Synchronization Function in Modbus Communication
  This error is identified by the slave protocol error flag bit under Modbus communication.
- Hardware initialization failure detection

If the hardware (driver, encoder) is not initialized successfully when the motor is powered up, a hardware initialization fault is reported.

- Flash Fault Detection
  When the motor's flash is operated, a flash failure is reported when the saved parameters or power-on reading of the stored parameters do not pass the test.
- Home Return Timeout Fault Detection
  When the motor is in CiA402 home return mode enable, if the home return action cannot be completed within the limited time 2005_h : $1C_h$ ($012E_h$), the home return timeout fault is reported.
- Tracking Fault Detection
  The motor reports a tracking fault when the position deviation exceeds ±6065_h ($03CA_h$) and the time reaches 6066_h ($03CC_h$) (ms) under position mode control.
- Software overrun fault detection
  When the motor is in position control mode, it is judged according to the software limit value (user unit) set by $607D_h$ ($03EF_h$ ~$03F1_h$), and a software overrun fault is reported if the actual position exceeds the set value.
- Overrun Switch Fault Detection
  According to the function numbers 14 and 15 set in the multi-function terminal, it detects whether the positive or negative limit switch is triggered or not, and if it is triggered, it reports an over-limit switch fault.
- Target position overflow fault detection
  The motor is in position control mode, the given position $607A_h$ ($03E7_h$) after the gear ratio 6091_h ($040E_h$ ~0410_h) exceeds the data range of data type int32 (-2147483648~+2147483647) then the

target position overflow fault is reported.

- Fault detection of too small curve planning parameters
  If the motor is in position control mode and the given impact or acceleration fails to realize S-curve in the given displacement interval, the curve planning parameter is reported to be too small.
- Flash initialization fault (0x5541)
  Controller is not saving Nimotion zone parameters, contact factory personnel.
- Commands that cannot be executed
  When the motor is in the enable state, or when the supply voltage is less than 80% of the rated voltage 2000_h :07_h (0063_h), the motor reports a fault of not being able to execute commands when operating the save parameter and the restore parameter.
- Encoder low voltage fault detection
  When the servo motor is powered on, the encoder battery voltage is detected to be ≤2.4V, and a fault 0x1007301 is reported.
- Encoder low voltage alarm detection
  When the servo motor is powered on, the encoder battery voltage is detected to be between 2.4V and 2.6V, and a alarm 0x4017302 is reported.
- Encoder communication fault detection
  The servo motor fails to communicate with the encoder and reports a 0x7303 fault.
- Encoder multi-turn counting error detection
  If the encoder is not calibrated or the calibration fails, a 0x7304 fault is reported.
- Encoder battery failure fault
  ①During power off, if no battery is connected, it will report a 0x7309 fault.
  ②The charged hot-swappable IO battery extension cable will report a 0x7309 fault.
  When this fault is reported, connect the battery, restart the power supply of the servo motor, and reset the position to zero.
- STO_1 failed to close detection
  STO_1 The input signal is high, the power drive power supply is disconnected, and the 0XFF05 fault is reported. The STO_2 input signal is high, the power drive power supply is disconnected, and a 0XFF07 fault is reported.
- STO_1 failed to enable detection
  The input signal of STO_1 is low, the power drive power supply is connected, and the 0XFF06 fault is reported. The input signal of STO_2 is low, the power drive power supply is connected, and the 0XFF08 fault is reported.
- STO input abnormal fault detection
  If either of the input signals STO_1 or STO_2 becomes high and the other remains low for more than 6ms, a 0xFF09 fault is reported.
- STO enabled state detection
  When at least one input signal of STO_1 or STO_2 drops, the STO enable state is entered. The power drive power supply and drive signal are cut off, and the servo enable command becomes invalid, reporting 0xFF0A.
- Non-safe state
  Each time the power is turned on or the STO is restored, the motor and encoder are not inspected completely or fail the inspection.

9 NiMotion Protocol

The NiMotion protocol optimizes the Modbus protocol and creates two new message formats (process data message and synchronization message) on top of Modbus. Through these two telegrams can achieve multi-command control, greatly reduce the data processing time, improve the coordination of multi-motor control, and further enable the motor to achieve two-two synchronization function. The process data message is used for the free combination of multiple parameters into a frame message to send and receive processing at the same time, and the synchronization message is used to control the timing of the effective command.

R_PDO: Receive process data;

T_PDO: Send process data.

9.1 Master Slave Timing Diagram

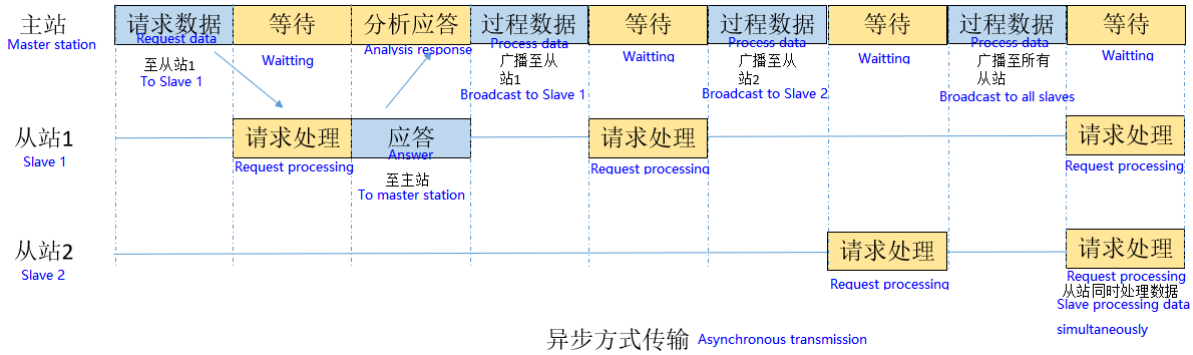


Figure9-1

In the asynchronous transmission process, the unicast process data for slave 1 sends an answer to the master after being processed by slave 1, and the master receives an error-free answer and sends the broadcast process data for slave 1, which is only processed by slave 1, but there is no answer; the master sends the broadcast process data for slave 2, which is processed by slave 2, but there is no answer; and the master sends the broadcast process data for all the slaves The master sends broadcast process data for all the slaves, and all the slaves process it, but there is no answer.

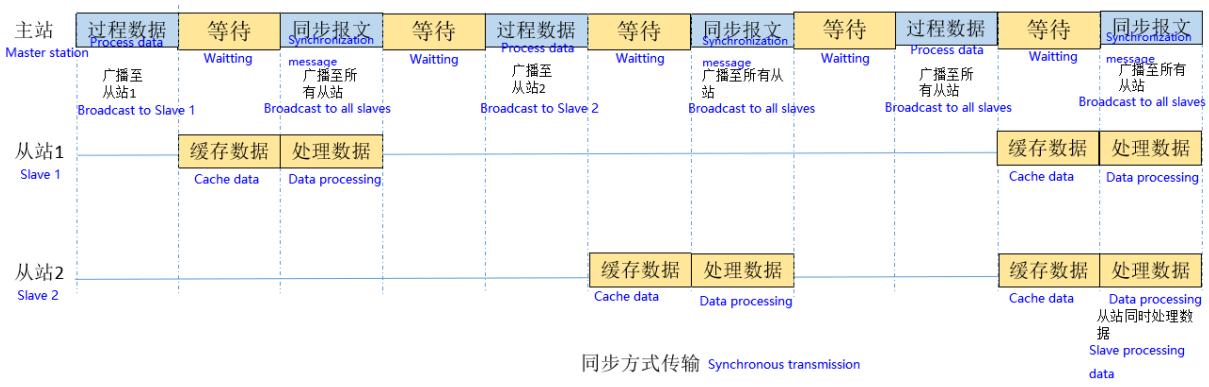


Figure9-2

During synchronous transmission, the master sends broadcast process data for slave 1, slave 1 receives the message and just caches the data and does not process it, the master broadcasts and

sends the synchronization message again, and slave 1 starts processing the data; the master sends broadcast process data for slave 2, slave 2 receives the message and just caches the data and does not process it, the master broadcasts and sends the synchronization message again, and slave 2 starts processing the data; the master sends broadcast process data for all slaves, slave 1 and slave 2 receive the message and both cache the data and do not process it, the master broadcasts and sends the synchronization message again, and both slave 1 and slave 2 start processing the data. The master sends broadcast process data for all slaves, slave 1 and slave 2 receive the message and just cache the data and do not process it, the master broadcasts a synchronization message again and slave 1 and slave 2 start to process the data at the same time.

9.2 Message Format

9.2.1 synchronization message

A synchronization message is a broadcast message used to determine when process data is updated to a register in synchronization mode by redefining the meaning of the register address.

The function of the synchronization telegram is to update the data stored in the buffer to the registers in the synchronization mode and thus control the slave changes.

Synchronous broadcast messages are specified as follows:

Table 9-1 Master synchronization request messages

Serial Number	Field Name	Byte Length (bytes)	Element	Note
1	address field	1	0x00	broadcast mode
2	function code	1	0x10	Write multiple registers
3	Register Address High Byte	1	0x80	Synchronization message identifier 0x8000
4	Register Address Low Byte	1	0x00	
5	Number of registers high byte	1	0x00	1 register
6	Number of registers low byte	1	0x01	
7	byte count	1	0x02	2 bytes of data portion
8	Register Value High Byte	1	0x66	Synchronized message data value flag Fixed to 0x6688
9	Register Value Low Byte	1	0x88	
10	CRC16 Low Bit	1	Calculate CRC low byte	16-bit CRC checksum
11	CRC16 High	1	Calculate CRC high byte	

Synchronization message is to write data to the slave, considering the process data message is to write multiple registers, so in order to reduce the change of the existing program, so the uniform selection of 0x10 function code.

In order to develop a specific synchronization message, the register address is changed to the identification of the synchronization message, and the highest bit is set to 1, 0x8000, considering the need to distinguish it from the existing register address.

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

9.2.2 Process Data Write Operations

9.2.2.1 request message

Support modbus protocol message format

Read content, length, set according to registers (0x6000~0x6021).

Table 9-2 Process data requests

Serial Number	Field Name	Byte Length (bytes)	Element	Note
1	address field	1	0x00	Broadcast/unicast mode
2	function code	1	0x10	Write multiple registers
3	Register Address High	1	0x83	message mode
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	byte count	1	0x0A	Number of registers*2 (2~66)
8	Register Value High Byte	1	0x00	Motor address (0~247) 0 for sending to all motors
9	Register Value Low Byte	1	0x01	
10	Register Value High Byte	1	0x00	The value of the register (set up 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	1	0x03	

Serial Number	Field Name	Byte Length (bytes)	Element	Note
	Byte			
17	Register Value Low Byte	1	0x84	
18	CRC16 Low Bit	1	Calculate CRC low byte	16-bit CRC checksum
19	CRC16 High	1	Calculate CRC high byte	

The process data is written to the slave for multiple registers, so the 0x10 function code is used.

In order to develop a specific process data message, the register address is changed to the identification of the process data message. Considering the need to distinguish it from the existing register address, the highest bit is set to 1. Referring to the CAN customized process data format, the process data identification is +0x300, i.e. 0x8300.

The mapping has a maximum of 16 entries, and each entry has a maximum of two registers, so the mapping has a maximum of 32 registers, plus a motor address accounts for one register, so the number of registers ranges from 1 to 33.

Each register occupies two byte counts, so the range of byte counts is 2 to 66.

The node address occupies one register, and there can only be a maximum of 247 slaves on the modbus bus. So the range is 1~247. if it is 0, it means send to all slaves.

9.2.2.2 Reply Message

Refer to the 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 content, length, set according to registers (0x6000~0x6021).

Table 9-3 Process data read request messages

Serial Number	Field Name	Byte Length (bytes)	Element	Note
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)

Serial Number	Field Name	Byte Length (bytes)	Element	Note
6	Number of registers low byte	1	0x02	
13	CRC16 Low Bit	1	Calculate CRC low byte	16-bit CRC checksum
14	CRC16 High	1	Calculate CRC high byte	

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

The process data is read from the cache, so the address is also the address of the cache.

There are up to 16 mapped entries in the cache, and each entry has at most two registers, so the number of registers ranges from 1 to 32.

9.2.3.2 Reply Message

Refer to the modbus protocol 0x04 function code response message.

9.3 Exception Handling

Refer to Chapter 8, Fault Management.

9.4 Example

1. Set the Slave Buffer Mapping Entry Count Register (0x5000) to determine the number of entries as required.
2. Depending on the number of entries, the setting starts from the R_PDO mapped 1 register and can be set up to a maximum of mapped 12 registers, with the register high byte determining the start address of the register to be mapped to, and the register low byte determining the number of registers to be mapped.
3. Set the Slave Buffer Mapping Entry Number Register (0x6000) to determine the number of entries as required.
4. Depending on the number of entries, the setting starts from T_PDO mapping 1 register and can be set up to a maximum of mapping 12 registers, with the register high byte determining the start address of the register to be mapped to and the register low byte determining the number of registers to be mapped.
5. Set Synchronization Mode Enable $200C_h$: 18_h (0246_h) to enable (0x01).
6. The master sends a process data message with register address 0x8300 and the slave executes the function according to the function code.
7. The master sends a synchronization message with register address 0x8000 and data value 0x6688, and the slave executes the function code function. The data update is triggered and the motor status is changed accordingly.

Realize the process:

1. The number of RPDO mapping entries for motor #1 is 2.

Table9-4

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
01	0x10	50 00	00 02	04	00 00 00 02	-

2. No. 1 motor RPDO mapping control word 6040_h : 0_h (0380_h), target position $607A_h$: 0_h ($03E7_h$).

Table9-5

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
01	0x10	50 02	00 04	08	03 80 00 01 03 E7 00 02	-

3. No. 1 motor on synchronization mode $200C_h$: 18_h (0246_h).

Table9-6

Slave Address	Function Code	Register Address	Register Value	CRC Value
01	0x06	02 46	00 01	-

4. Motor No. 1 saves the parameters and restarts with power off.

Table9-7

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value

01	0x10	00 26	00 02	04	65 76 61 73	-

5. The master broadcast sends process data (control word 6, target position 1000) to motor No. 1.

Table9-8

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
00	0x10	83 00	00 04	08	00 01 00 06 00 00 03 E8	-

6. The master broadcast sends a synchronization message to update the parameters.

Table9-9

Slave Address	Function Code	Register Address	Number of Registers	Byte Count	Register Value	CRC Value
00	0x10	80 00	00 01	02	66 88	-

9.5 Mapping Parameter Settings

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

Example: the value of R_PDO map 1 is 0x02310001

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

The low 16 bits 0x0001 means that the parameters of this map occupy a register.

Table9-10

Modbus address	Name (of a thing)	Description	Number of Registers	Data Range
5000	R_PDO Number of mapping entries	Number of entries in the process data message mapping registers	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

Modbus address	Name (of a thing)	Description	Number of Registers	Data Range
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
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

Modbus address	Name (of a thing)	Description	Number of Registers	Data Range
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 device network state. Each object contains an index, subindex, name, description, data type, data range, accessibility, whether it can be mapped, factory settings, units, and how it takes effect.

interpretation of nouns

"Index": specifies the position of objects of the same class in the object dictionary, expressed in hexadecimal

"Sub-index": under the same index, which contains multiple objects, the bias of each object under the class

"Data types and ranges": please refer to the following table for details

Table10-1

Data Type	Data Range	Data Length
Int8	-128~127	1 byte
Int16	-32768~32767	2 bytes
Int32	-2147483648~2147483647	4 bytes
Uint8	0~255	1 byte
Uint16	0~65535	2 bytes
Uint32	0~4294967295	4 bytes
String	ASCII	~

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

Table 10-2

Accessibility	Clarification
RW	readable
WO	write only
RO	read-only (computing)

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

Table10-3

Mappable	Clarification
NO	Not mappable in PDO
RPDO	Can be used as an RPDO
TPDO	Can be used as a TPDO

"Mode of entry into force": please refer to the following table for details

Table10-4

Mode of entry into force	Clarification
immediate effect	The set values take effect immediately after the parameter setting is completed.
Suspension in effect	After the parameter setting is completed, wait until the drive is not in the operation state and the setting value takes effect.
Re-energized.	After the parameter setting is completed, turn on the power of the drive again and the set value takes effect

10.1 Communication Parameters 1000h Group

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

Table10-5

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
1001h	-	00h	error register	Uint8	-	RO	NO	0	VAR	-
1003h	00h	01h	Number of alarms present in the current device	uint8	-	RO	NO	-	VAR	-
	01h	02h	History Alarm Cache	uint32	-	RO	NO	-	VAR	-
	02h	04h		uint32	-	RO	NO	-	VAR	-
	03h	06h		uint32	-	RO	NO	-	VAR	-
	04h	08h		uint32	-	RO	NO	-	VAR	-
	05h	0Ah		uint32	-	RO	NO	-	VAR	-
	06h	0Ch		uint32	-	RO	NO	-	VAR	-
	07h	0Eh		uint32	-	RO	NO	-	VAR	-
	08h	10h		uint32	-	RO	NO	-	VAR	-
	09h	12h		uint32	-	RO	NO	-	VAR	-
	0Ah	14h		uint32	-	RO	NO	-	VAR	-
	0Bh	16h		uint32	-	RO	NO	-	VAR	-
	0Ch	18h		uint32	-	RO	NO	-	VAR	-
	0Dh	1Ah		uint32	-	RO	NO	-	VAR	-
	0Eh	1Ch		uint32	-	RO	NO	-	VAR	-
	0Fh	1Eh		uint32	-	RO	NO	-	VAR	-
	10h	20h		uint32	-	RO	NO	-	VAR	-
1010h	01h	26h	Save all current parameters	uint32	-	RW	NO	1	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate 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 Manufacturer Defined Parameters 2000h Group Description

Table10-6

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2000h (motor parameters)	06h	62h	Bus Encoder Type	uint16	0~1	RW	NO	1	-	Suspension in effect
	07h	63h	rated voltage	uint16	24~48	RW	NO	48	1V	Suspension in effect
	08h	64h	rating	uint16	-	RW	NO	40	0.01KW	Suspension in effect
	09h	65h	rated current	uint16	-	RW	NO	1250	0.01A	Suspension in effect
	0Ah	66h	Maximum current	uint16	-	RW	NO	2250	0.01A	Suspension in effect
	0Bh	67h	Maximum current duration	uint16	-	RW	NO	3000	0.1s	Suspension in effect
	0Ch	68h	Rated torque	uint16	-	RW	NO	127	0.01Nm	Suspension in effect
	0Dh	69h	Maximum torque	uint16	-	RW	NO	2000	0.01Nm	Suspension in effect
	0Eh	6Ah	rated speed	uint16	-	RW	NO	3000	1rpm	Suspension in effect
	0Fh	6Bh	Maximum speed	uint16	-	RW	NO	6000	1rpm	Suspension in effect
	10h	6Ch	Moment of inertia Jm	uint16	-	RW	NO	30	$0.01kgcm^2$	Suspension in effect
	11h	6Dh	Permanent magnet	uint16	-	RW	NO	4	1	Suspension

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			synchronous motor pole pair number							in effect
	12_h	$6E_h$	Stator resistance	uint16	-	RW	NO	115	0.001Ω	Suspension in effect
	13_h	$6F_h$	Stator inductance Lq	uint16	-	RW	NO	25	0.01mH	Suspension in effect
	14_h	70_h	Stator inductance Ld	uint16	-	RW	NO	25	0.01mH	Suspension in effect
	16_h	72_h	Torque coefficient Kt	uint16	-	RW	NO	11	0.01Nm/Arms	Suspension in effect
	19_h	75_h	Encoder absolute digits	uint32	-	RW	NO	14	1bit	Suspension in effect
	$1A_h$	77_h	Number of encoder poles	uint16	-	RW	NO	1	1	Suspension in effect
	$1C_h$	$7A_h$	Position sensor selection	uint16	-	RW	NO	2	-	Suspension in effect
	$1D_h$	$7B_h$	Z signal corresponds to the electrical angle	uint16	-	RW	NO	0	1/4096	Suspension in effect
	$1E_h$	$7C_h$	Rising edge of U-phase corresponds to the diagonal angle	int16	-	RW	NO	0	1	Suspension in effect
2001_h (drive parameters)	07_h	$8B_h$	Rated output current	Uint16	0~65535	RO	NO	1200	0.01A	Suspension in effect
	08_h	$8C_h$	Maximum Output Current	uint16	0~65535	RO	NO	1500	0.01A	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	0D$_h$	91$_h$	Switching Dead Time	uint16	0~65535	RW	NO	100	us	Suspension in effect
	0E$_h$	92$_h$	DC bus overvoltage protection point	uint16	0~65535	RW	NO	100	1V	Suspension in effect
	0F$_h$	93$_h$	DC bus voltage relief point	uint16	0~65535	RW	NO	55	1V	Suspension in effect
	10$_h$	94$_h$	DC bus voltage undervoltage point	uint16	0~65535	RW	NO	24	1V	Suspension in effect
	11$_h$	95$_h$	Driver overcurrent protection point	uint16	0~65535	RW	NO	10	1A	Suspension in effect
	18$_h$	9C$_h$	Current Sampling Filter Parameters	uint16	0~65535	RW	NO	100	0.01	Suspension in effect
	1C$_h$	A0$_h$	Current loop cutoff frequency	uint16	0~65535	RW	NO	800	1Hz	Suspension in effect
	1D$_h$	A1$_h$	Open loop operating current	uint16	0~65535	RW	NO	625	0.01A	Suspension in effect
2002$_h$ (basic control parameters)	01$_h$	B1$_h$	Control mode selection	uint16	0~5	RW	NO	4	-	The shutdown setting takes effect immediatel y
			Specific descriptions: 0~ CIA402 mode, 1~ NiMotion position mode, 2~ NiMotion speed mode, 3~ NiMotion torque mode, 4~ NiMotion open-loop mode, 5~ offline identification of motor parameters							
	04$_h$	B4$_h$	Duty cycle of output pulse	uint16	0~100	RW	NO	50	1%	Running setup stop

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
										go into effect
	13h	C3h	Internal and External Suppression Resistor Selection	uint16	0~2	RW	NO	0	1	Re-energized.
	1Ch	CDh	firmware code	uint32	-	RO	NO	0	-	-
	1Eh	CFh	IAP software version	uint32	-	RO	NO	0	-	-
	1Fh	D1h	hardware version	uint32	-	RO	NO	0	-	-
2003h (Entity terminal input parameters)	03h	D5h	DI1 terminal function selection	uint16	0~55	RW	NO	1	1	Suspension in effect
			Specific description: see section 6.1 for details							
	04h	D6h	DI1 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
			Specific description: see section 6.1 for details							
	05h	D7h	DI2 terminal function selection	uint16	0~48	RW	NO	2	1	Suspension in effect
	06h	D8h	DI2 terminal logic selection	uint16	0~4	RW	NO	2	1	Suspension in effect
	07h	D9h	DI3 terminal function selection	uint16	0~48	RW	NO	12	1	Suspension in effect
	08h	DAh	DI3 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	09h	DBh	DI4 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	0Ah	DCh	DI4 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2003h (Entity terminal input parameters)	0Bh	DDh	DI5 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	0Ch	DEh	DI5 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	0Dh	DFh	DI6 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	0Eh	E0h	DI6 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	0Fh	E1h	DI7 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	10h	E2h	DI7 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	11h	E3h	DI8 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	12h	E4h	DI8 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	13h	E5h	DI9 terminal function selection	uint16	0~48	RW	NO	0	1	Suspension in effect
	14h	E6h	DI9 terminal logic selection	uint16	0~4	RW	NO	0	1	Suspension in effect
	15h	E7h	Power-Up Effective Function Assignment	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: bit0-DI1, bit1-DI2, bit2-DI3; a corresponding bit of 1 indicates that the corresponding DI port is at a high level of 3.0V							
	16h	E8h	Power-Up Effective Function Assignment	uint16	0~65535	RW	NO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			Specific description: bit0-DI1, bit1-DI2, bit2-DI3; a corresponding bit of 1 indicates that the corresponding DI port is at a high level of 3.0V							
	17h	E9h	AI1 bias	int16	-32768~32767	RW	NO	0	1mV	Suspension in effect
	18h	EAh	AI1 input filter time constant	int16	0~65535	RW	NO	0	-	Suspension in effect
	19h	EBh	AI1 dead zone	uint16	0~65535	RW	NO	0	1mV	Suspension in effect
	1A h	ECh	AI1 multiplier	int16	-32768~32767	RW	NO	0	0.001	Suspension in effect
	1Bh	EDh	AI2 bias	int16	-32768~32767	RW	NO	0	1mV	Suspension in effect
	1Ch	EEh	AI2 input filter time constant	int16	0~65535	RW	NO	0	-	Suspension in effect
	1Dh	EFh	AI2 dead zone	uint16	0~65535	RW	NO	0	1mV	Suspension in effect
	1E h	F0h	AI2 multiplier	int16	-32768~32767	RW	NO	0	0.001	Suspension in effect
	1F h	F1h	Analog 10V corresponds to the speed value	uint16	0~4000	RW	NO	0	rpm	Suspension in effect
	20 h	F2h	Analog 10V corresponds to the torque value	uint16	0~1000	RW	NO	0	0.1%	Suspension in effect
2004h (terminal output parameters)	01h	F8h	DO1 terminal function selection	uint16	0~30	RW	NO	0	1	Suspension in effect
			Specific description: see section 6.1 for details							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	02h	F9h	DO1 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
			Specific description: see section 6.1 for details							
	03h	FAh	DO2 terminal function selection	uint16	0~30	RW	NO	0	1	Suspension in effect
	04h	FBh	DO2 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	05h	FCh	DO3 terminal function selection	uint16	0~30	RW	NO	0	1	Suspension in effect
	06h	FDh	DO3 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	07h	FEh	DO4 terminal function selection	uint16	0~30	RW	NO	0	1	Suspension in effect
	08h	FFh	DO4 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	09h	100h	DO5 terminal function selection	uint16	0~30	RW	NO	0	1	Suspension in effect
	0Ah	101h	DO5 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
2005h (position control parameters)	01h	10Dh	Location command source	uint16	0~65535	RW	NO	1	1	Suspension in effect
			Specific description: 0-pulse, 1-step amount							
	05h	112h	step size	int16	-32768~32767	RW	NO	100	inc	Suspension in effect
	1C	12Eh	Origin return time limit	uint16	0~65535	RW	NO	10000	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2006_h (speed control parameters)	01_h	148_h	Main Speed Command A Source	uint16	0~65535	RW	NO	0	1	Suspension in effect
	02_h	149_h	Auxiliary Speed Command B Source	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0-digit given (2006_h : 04_h set value)							
	03_h	$14A_h$	Speed command selection	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0-A (only 2006_h : 01_h is in effect),1-B (only 2006_h : 02_h is in effect),2-A+B (2006_h : 01_h and 2006_h : 02_h are in effect at the same time)							
	04_h	$14B_h$	Speed command keypad setpoint	int16	-32768~32767	RW	NO	10	1rpm	immediate effect
	06_h	$14D_h$	Effective speed setting value	int16	-32768~32767	RW	NO	0	1rpm	immediate effect
	07_h	$14E_h$	Speed command acceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	immediate effect
	08_h	$14F_h$	Speed command deceleration ramp time constant	uint16	0~65535	RW	NO	10	1ms	immediate effect
	09_h	150_h	Maximum RPM Threshold	uint16	0~65535	RW	NO	6000	1rpm	immediate effect
	$0A_h$	151_h	Forward velocity threshold	uint16	0~65535	RW	NO	4000	1rpm	immediate effect
	$0B_h$	152_h	Reverse velocity threshold	uint16	0~65535	RW	NO	4000	1rpm	immediate effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	12_h	159	Select $606C_h$ Units for Speed Feedback	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0 ~ $606C_h$ unit for rpm, 1 ~ $606C_h$ unit for user unit							
2007_h (torque control parameters)	01_h	$15E_h$	Main Torque Command A Source	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0~digit given (set value of 2007_h : 04_h)							
	02_h	$15F_h$	Auxiliary Torque Command B Source	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0-digit given (2007_h : 04_h set values)							
	03_h	160_h	Torque command selection	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 0-A (only 2007_h : 01_h is in effect),1-B (only 2007_h : 02_h is in effect),2-A+B (2007_h : 01_h and 2007_h : 02_h are in effect at the same time)							
	04_h	161_h	Torque command keypad set value	int16	-32768~32767	RW	NO	10	0.1%	Suspension in effect
	05_h	162_h	Effective torque command setting	int16	-32768~32767	RW	NO	10	0.1%	Suspension in effect
	06_h	163_h	Torque command filter time constant	uint16	0~65535	RW	NO	0	0.01ms	immediate effect
			Too large a setting will reduce answerability.							
	$0A_h$	167_h	Positive internal torque limit	uint16	0~65535	RW	NO	1000	0.1%	immediate effect
	$0B_h$	168_h	Reverse internal torque limitation	uint16	0~65535	RW	NO	1000	0.1%	immediate effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	10_h	$16D_h$	Torque control forward speed limit value	uint16	0~65535	RW	NO	3000	1rpm	immediate effect
	11_h	$16E_h$	Torque control reverse speed limit value	uint16	0~65535	RW	NO	3000	1rpm	immediate effect
	13_h	170_h	Detecting torque while searching for home position in blocked rotation	uint16	0~65535	RW	NO	500	0.1%	
	15_h	172_h	Detection time when blocking rotation to find the home position	uint16	0~65535	RW	NO	500	ms	
2008_h (Gain parameters)	01_h	178_h	Velocity Loop Gain	uint16	1~20000	RW	NO	500	0.1Hz	Running the settings takes effect immediatel y
			Description: Set the proportional gain of the speed ring. This parameter determines the response of the speed ring, the larger the response of the speed ring, the faster the response of the speed ring, 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	Velocity loop integration time constant	uint16	0~51200	RW	NO	800	0.01ms	Running the settings takes effect immediatel y

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2008h (Gain parameters)			Specific description: Set the integration time constant of the speed ring. The smaller the value set, the stronger the integration effect is, and the deviation value at stopping is closer to 0 sooner. when this parameter is set to 51200, there is no integration effect.							
	03h	17Ah	Position Loop Gain	uint16	0~20000	RW	NO	1500	1	Running the settings takes effect immediatel y
			Specific 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 velocity loop gain	uint16	1~20000	RW	NO	1	0.1Hz	immediate effect
	05h	17Ch	Second velocity loop integration time constant	uint16	0~51200	RW	NO	15	0.01ms	immediate effect
	06h	17Dh	Second position loop gain	uint16	0~20000	RW	NO	0	1	immediate effect
	08h	17Fh	Second gain mode setting	uint16	0~9	RW	NO	0	1	immediate effect
			9~Actual speed, the state in which the absolute value of the actual speed exceeds (rank + time lag) [rpm] lasts during the delay time period] switches to the second gain, and the state in which the absolute value of the actual speed is less than (rank - time lag) [rpm] lasts during the delay time period returns to the first gain.							
	09h	180h	Gain switching condition selection	uint16	0~10	RW	NO	0	1	immediate effect
			Specific description: 0~ Group 1 gain, 1~ Group 2 gain							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	$0A_h$	181_h	Gain switching delay time	uint16	0~10000	RW	NO	0	0.1ms	immediate effect
	$0B_h$	182_h	Gain switching level	uint16	0~20000	RW	NO	0	1	immediate effect
			The generation of the actual switching action is influenced by a combination of both grade and time lag conditions in the manner described in 2008_h :08_h . Depending on the gain switching mode, the units of the switching level change.							
	$0C_h$	183_h	Gain switching time lag	uint16	0~20000	RW	NO	0	1	immediate effect
			The generation of the actual switching action is influenced by a combination of both grade and time lag conditions in the manner described in 2008_h :08_h . Depending on the gain switching mode, the units of the switching level change.							
	$0F_h$	186_h	Velocity feed-forward filtering time constant	uint16	0~65535	RW	NO	0	0.01ms	immediate effect
	10_h	187_h	Velocity Feed Forward Gain	uint16	0~65535	RW	NO	0	0.1%	immediate effect
			When making adjustments, you should repeatedly adjust 2008_h : $0F_h$ and 2008_h : 10_h to find a well-balanced setting.							
	11_h	188_h	Torque feed-forward filtering time constant	uint16	0~65535	RW	NO	0	0.01ms	immediate effect
	12_h	189_h	Torque Feed Forward Gain	uint16	0~65535	RW	NO	0	0.001	immediate effect
			When making adjustments, you should repeatedly adjust 2008_h : 11_h and 2008_h : 12_h to find a well-balanced setting.							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	14_h	$18B_h$	Velocity feedback low-pass filter cutoff frequency	uint16	0~4000	RW	NO	900	1Hz	immediate effect
			Specific description: the smaller the setting, the smaller the speed feedback fluctuation, but the feedback delay is also larger, generally keep the default parameters can be.							
2009_h (gain parameters)	06_h	197_h	Parameter offline recognition status display	uint16	0~65535	RO	NO	0	1	-
	$0D_h$	$19E_h$	Group 1 Trap Frequency	uint16	0~2000	RW	NO	0	1Hz	immediate effect
			Specific description: Set the center frequency of the trap, which is the mechanical resonance frequency.							
	$0E_h$	$19F_h$	Trap width of group 1 traps	uint16	0~2000	RW	NO	0	1Hz	immediate effect
			Specific description: Setting the bandwidth on either side of the center frequency of the mechanical resonance							
	$0F_h$	$1A0_h$	Group 1 Trap Depth	uint16	0~100	RW	NO	0	%	immediate effect
			Specific descriptions: 0~ no trapping, 100~ maximum trap depth							
	10_h	$1A1_h$	Group 2 Trap Frequency	uint16	0~2000	RW	NO	0	1Hz	immediate effect
	11_h	$1A2_h$	Trap width of group 2 traps	uint16	0~2000	RW	NO	0	1Hz	immediate effect
	12_h	$1A3_h$	Group 2 trap depth	uint16	0~100	RW	NO	0	1	immediate effect
	13_h	$1A4_h$	Group 3 Trap Frequency	uint16	0~2000	RW	NO	0	1Hz	immediate effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	14_h	$1A5_h$	Trap width of group 3 traps	uint16	0~2000	RW	NO	0	1Hz	immediate effect
	15_h	$1A6_h$	Group 3 trap depth	uint16	0~100	RW	NO	0	1	immediate effect
	16_h	$1A7_h$	Group 4 Trap Frequency	uint16	0~2000	RW	NO	0	1Hz	immediate effect
	17_h	$1A8_h$	Trap width of group 4 traps	uint16	0~2000	RW	NO	0	1Hz	immediate effect
	18_h	$1A9_h$	Group 4 trap depth	uint16	0~100	RW	NO	0	1	immediate effect
200Ah (fault and protection parameters)	06h	$1BC_h$	Overspeed Fault Threshold	uint16	0~65535	RW	NO	0	1	immediate effect
$200B_h$ (monitoring parameters)	01_h	$1DE_h$	Motor Driver Internal Status	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-up time	uint32	0~4294967295	RO	NO	0	s	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	10	1F2h	AI1 Sampling Voltage Value	uint16	-1200-1200	RO	NO	-1200	mV	
	11	1F3h	AI2 Sampling Voltage Value	uint16	-1200-1200	RO	NO	-1200	mV	
	12h	1F4h	RMS value of phase A current	uint16	0~65535	RO	NO	0	0.01A	-
	13h	1F5h	RMS value of phase B current	uint16	0~65535	RO	NO	0	0.01A	-
	14h	1F6h	RMS phase C current	uint16	0~65535	RO	NO	0	0.01A	-
	15h	1F7h	Busbar voltage value	uint16	0~65535	RO	NO	0	0.1V	-
	16h	1F8h	Module Temperature Value	int16	-32768~32767	RO	NO	0	1°C	-
	2Dh	211h	Actual motor speed	uint16	0~65535	RO	NO	-	0.1rpm	-
200Ch (communication parameters)	01h	22Fh	Communication method selection	uint16	0~2	RW	NO	1	1	Re-energized.
			Specific description: 0~External pulse control (default CAN communication), 1~EtherCAT, 2~CAN							
	02h	230h	Drive Axis Address	uint16	1~247	RW	NO	1	1	Re-energized.
			1~247: When multiple motors are grouped for networking, each motor can only have a unique address, otherwise it will result in abnormal or no communication.							
	03h	231h	Serial Baud Rate Setting	uint16	0~9	RW	NO	5	1	Re-energized.
			The communication rate of the motor must be the same as the communication rate of the host computer, otherwise it cannot communicate.							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	18h	246h	Synchronization Mode Enable	uint16	0~1	RW	NO	0	1	Re-energized.
200Eh (fault code)	01h	25Bh	Motor overload	int32	-214748364~2147483647	RW	NO	67183377	1	immediate effect
	02h	25Dh	motor stalling	int32	-214748364~2147483647	RW	NO	8978	1	immediate effect
	03h	25Fh	power overvoltage	int32	-214748364~2147483647	RW	NO	78352	1	immediate effect
	04h	261h	Power supply undervoltage	int32	-214748364~2147483647	RW	NO	16855584	1	immediate effect
	05h	263h	overheating	int32	-214748364~2147483647	RW	NO	82448	1	immediate effect
	06h	265h	too low a temperature	int32	-214748364~2147483647	RW	NO	82464	1	immediate effect
	07h	267h	Parameter setting error	int32	-214748364~2147483647	RW	NO	90912	1	immediate effect
	08h	269h	Motor overspeed	int32	-214748364~2147483647	RW	NO	94992	1	immediate effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
200E$_h$ (fault code)	09$_h$	26B$_h$	communications failure	int32	-214748364~2147483647	RW	NO	95489	1	immediate effect
	0A$_h$	26D$_h$	Origin return timeout error	int32	-214748364~2147483647	RW	NO	34320	1	immediate effect
	0B$_h$	26F$_h$	position overshooting error	int32	-214748364~2147483647	RW	NO	34321	1	immediate effect
	0C$_h$	271$_h$	software overrun fault	int32	-214748364~2147483647	RW	NO	50431507	1	immediate effect
200E$_h$ (fault code)	0D$_h$	273$_h$	Overrun switch failure	int32	-214748364~2147483647	RW	NO	50431507	1	immediate effect
	OE$_h$	275$_h$	Output phase failure	int32	-214748364~2147483647	RW	NO	12933	1	immediate effect
	0F$_h$	277$_h$	Errors in curve planning calculations	int32	-214748364~2147483647	RW	NO	67208725	1	immediate effect
	10$_h$	279$_h$	Target position overflow	int32	-214748364~2147483647	RW	NO	99862	1	immediate effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	11_h	$27B_h$	Curve planning parameters too small	int32	-214748364~2147483647	RW	NO	83985943	1	immediate effect
200F (encoder parameters)	$0A_h$	$28B_h$	Multi-turn absolute encoder function test	int32	-2147483648~2147483647	RO	NO	0	1	Suspension in effect
	$0B_h$	$28C_h$	Multi-turn absolute encoder battery voltage	int32	-2147483648~2147483647	RO	NO	0	mv	Suspension in effect
	$0C_h$	$28D_h$	Multiturn Absolute Encoder Calibration Instructions	int32	-2147483648~2147483647	RW	NO	0	1	immediate effect
	$0D_h$	$28E_h$	Multiturn Absolute Encoder Calibration Completion Detection Command	int32	-2147483648~2147483647	RO	NO	0	1	immediate effect
	$0E_h$	$28F_h$	Multi-turn absolute encoder zero absolute A	int32	-2147483648~2147483647	RW	NO	0	1	immediate effect
	$0F_h$	290_h	Multiturn absolute encoder zero absolute B	int32	-2147483648~2147483647	RW	NO	0	1	immediate effect
	01_h	291_h	Multi-stage position operation mode	uint16	0~2	RW	NO	0	1	Suspension in effect
			Specific description: see section 5.16.4 for details							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (Multi-segment position function parameters)	02h	292h	Number of position instruction end segments	uint16	1~16	RW	NO	1	1	Suspension in effect
			Specific description: Setting the total number of segments for multi-segment position commands							
	04h	294h	Number of multisegment position cycles	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: number of cycles in which the multi-segment position operation mode is 1							
	05h	295h	Position command type selection	uint16	0~1	RW	NO	0	1	Suspension in effect
			Specific description: 0-relative displacement; 1-absolute displacement							
	06h	296h	Multi-segment positional loop starting segment	uint16	0~16	RW	NO	0	1	Suspension in effect
			Specific description: Starting segment of a multi-segment position operating mode 1							
	07h	297h	Paragraph 1 mobile displacement	int32	-2147483648~ 2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	08h	299h	Maximum operating speed for 1st displacement	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	09h	29Ah	1st displacement acceleration and deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
			Specific description: Setting the time for the first motor of the multi-stage position to go from 0rpm to 1000rpm.							
	0Ah	29Bh	Waiting time after completion of the first displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011_h (Multi-segment position function parameters)			Specific description: the waiting time after the 1st displacement of a multi-segment position has been run and before the next displacement is run							
	$0B_h$	$29C_h$	Paragraph 2 mobile displacement	int32	-2147483648~ 2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	$0C_h$	$29E_h$	Maximum operating speed for 2nd displacement	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	$0D_h$	$29F_h$	2nd displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$0E_h$	$2A0_h$	Waiting time after completion of the 2nd displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$0F_h$	$2A1_h$	Paragraph 3 mobile displacement	int32	-2147483648~ 2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	10_h	$2A3_h$	Maximum operating speed for 3rd displacement	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	11_h	$2A4_h$	3rd displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	12_h	$2A5_h$	Waiting after the 3rd displacement is completed timing	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (Multi-segment position function parameters)	13h	2A6h	Paragraph 4 mobile displacement	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	14h	2A8h	Maximum operating speed for segment 4 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	15h	2A9h	4th displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	16h	2AAh	Waiting time after completion of paragraph 4 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	17h	2ABh	Paragraph 5 mobile displacement	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	18h	2ADh	Maximum operating speed for segment 5 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	19h	2AEh	Paragraph 5 displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	1Ah	2AFh	Waiting time after completion of paragraph 5 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	1Bh	2B0h	Paragraph 6 mobile displacement	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011_h (Multi-segment position function parameters)	$1C_h$	$2B2_h$	Maximum operating speed for segment 6 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	$1D_h$	$2B3_h$	Paragraph 6 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1E_h$	$2B4_h$	Waiting time after completion of paragraph 6 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1F_h$	$2B5_h$	Paragraph 7 mobile displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	user unit	Suspension in effect
	20_h	$2B7_h$	Maximum operating speed for segment 7 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	21_h	$2B8_h$	Paragraph 7 displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	22_h	$2B9_h$	Waiting time after completion of paragraph 7 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	23_h	$2BA_h$	Paragraph 8 mobile displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	user unit	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (Multi-segment position function parameters)	24h	2BCh	Maximum operating speed for segment 8 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	25h	2BDh	Paragraph 8 displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	26h	2BEh	Waiting time after completion of paragraph 8 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	27h	2BFh	Paragraph 9 mobile displacements	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	28h	2C1h	Maximum operating speed for paragraph 9 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	29h	2C2h	Paragraph 9 displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Ah	2C3h	Waiting time after completion of paragraph 9 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Bh	2C4h	Paragraph 10 mobile displacement	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (Multi-segment position function parameters)	2Ch	2C6h	Maximum operating speed for paragraph 10 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	2Dh	2C7h	Paragraph 10 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Eh	2C8h	Waiting time after completion of paragraph 10 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Fh	2C9h	Paragraph 11 mobile displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	user unit	Suspension in effect
	30h	2CBh	Maximum operating speed for paragraph 11 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	31h	2CCh	Paragraph 11 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	32h	2CDh	Waiting time after completion of paragraph 11 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	33h	2CEh	Paragraph 12 mobile displacement	int32	-2147483648~2147483647	RW	NO	-1073741824	user unit	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2011h (Multi-segment position function parameters)	34h	2D0h	Segment 12 Displacement Maximum Run tempo	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	35h	2D1h	Paragraph 12 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	36h	2D2h	Waiting time after completion of paragraph 12 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	37h	2D3h	Paragraph 13 mobile displacement	int32	-2147483648~ 2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	38h	2D5h	Maximum operating speed for paragraph 13 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	39h	2D6h	Paragraph 13 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	3Ah	2D7h	Waiting time after completion of paragraph 13 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	3Bh	2D8h	Paragraph 14 mobile displacements	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	3Ch	2DAh	Maximum operating speed for paragraph 14 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	3Dh	2DBh	Paragraph 14 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	3Eh	2DCh	Waiting time after completion of paragraph 14 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	3Fh	2DDh	Paragraph 15 movement displacement	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	40h	2DFh	Maximum operating speed for paragraph 15 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	41h	2E0h	Paragraph 15 displacement plus deceleration time	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	42h	2E1h	Waiting time after completion of paragraph 15 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	43h	2E2h	Paragraph 16 mobile displacements	int32	-2147483648~2147483647	RW	NO	-107374 1824	user unit	Suspension in effect
	44h	2E4h	Maximum operating speed for paragraph 16 displacements	uint16	0~3000	RW	NO	1	rpm	Suspension in effect
	45h	2E5h	Paragraph 16 displacement acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	46h	2E6h	Waiting time after completion of paragraph 16 displacement	uint16	0~65535	RW	NO	0	ms	Suspension in effect
2012h (Multi-Speed Function Parameters)	01h	2E9h	Multi-speed command operation mode	uint16	0~2	RW	NO	0	1	Suspension in effect
		Specific description: see section 5.17.4 for details								
	02h	2EAh	Speed command end segment selection	uint16	0~16	RW	NO	1	1	Suspension in effect
	03h	2EBh	Number of multispeed cycles	uint16	0~65535	RW	NO	0	1	Suspension in effect
	0Ch	2F4h	1st speed command	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	0Dh	2F5h	Running time of the 1st instruction	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012_h (Multi-Speed Function Parameters)	$0E_h$	$2F6_h$	Acceleration and deceleration time for paragraph 1	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$0F_h$	$2F7_h$	Paragraph 2 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	10_h	$2F8_h$	Running time of the 2nd instruction	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	11_h	$2F9_h$	Paragraph 2 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	12_h	$2FA_h$	Paragraph 3 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	13_h	$2FB_h$	Running time of paragraph 3 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	14_h	$2FC_h$	Paragraph 3 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	15_h	$2FD_h$	Paragraph 4 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	16_h	$2FE_h$	Running time of paragraph 4 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	17_h	$2FF_h$	Paragraph 4 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	18_h	300_h	Paragraph 5 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012_h (Multi-Speed Function Parameters)	19_h	301_h	Running time of paragraph 5 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1A_h$	302_h	Paragraph 5 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1B_h$	303_h	Paragraph 6 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	$1C_h$	304_h	Running time of paragraph 6 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1D_h$	305_h	Paragraph 6 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$1E_h$	306_h	Paragraph 7 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	$1F_h$	307_h	Running time of paragraph 7 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	20_h	308_h	Paragraph 7 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	21_h	309_h	Paragraph 8 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	22_h	$30A_h$	Running time of paragraph 8 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	23_h	$30B_h$	Paragraph 8 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012h (Multi-Speed Function Parameters)	24h	30Ch	Paragraph 9 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	25h	30Dh	Running time of paragraph 9 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	26h	30Eh	Paragraph 9 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	27h	30Fh	Paragraph 10 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	28h	310h	Running time of paragraph 10 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	29h	311h	Paragraph 10 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Ah	312h	Paragraph 11 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	2Bh	313h	Running time of paragraph 11 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Ch	314h	Paragraph 11 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	2Dh	315h	Paragraph 12 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012_h (Multi-Speed Function Parameters)	$2E_h$	316_h	Running time of paragraph 12 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	$2F_h$	317_h	Paragraph 12 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	30_h	318_h	Paragraph 13 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	31_h	319_h	Running time of paragraph 13 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	32_h	$31A_h$	Paragraph 13 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	33_h	$31B_h$	Paragraph 14 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	34_h	$31C_h$	Running time of paragraph 14 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	35_h	$31D_h$	Paragraph 14 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	36_h	$31E_h$	Paragraph 15 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2012h (Multi-Speed Function Parameters)	37h	31Fh	Running time of paragraph 15 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	38h	320h	Paragraph 15 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	39h	321h	Paragraph 16 instructions	int16	0~3000	RW	NO	-6000	rpm	Suspension in effect
	3Ah	322h	Running time of paragraph 16 instructions	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	3Bh	323h	Paragraph 16 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
	3Bh	323h	Paragraph 16 acceleration and deceleration times	uint16	0~65535	RW	NO	0	ms	Suspension in effect
2017h (virtual input terminal parameters)	01h	326h	VDI1 terminal function selection	uint16	0~44	RW	NO	0	1	Suspension in effect
			Specific description: see section 6.1 for details							
	02h	327h	VDI1 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
			Specific description: see section 6.1 for details							
	03h	328h	VDI2 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2017h (virtual input terminal parameters)	04h	329h	VDI2 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	05h	32Ah	VDI3 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	06h	32Bh	VDI3 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	07h	32Ch	VDI4 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	08h	32Dh	VDI4 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	09h	32Eh	VDI5 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	0Ah	32Fh	VDI5 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	0Bh	330h	VDI6 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	0Ch	331h	VDI6 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	0Dh	332h	VDI7 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	0Eh	333h	VDI7 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	0Fh	334h	VDI8 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	10h	335h	VDI8 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
2017_h (virtual input terminal parameters)	11_h	336_h	VDI9 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	12_h	337_h	VDI9 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	13_h	338_h	VDI10 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	14_h	339_h	VDI10 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	15_h	$33A_h$	VDI11 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	16_h	$33B_h$	VDI11 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	17_h	$33C_h$	VDI12 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	18_h	$33D_h$	VDI12 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	19_h	$33E_h$	VDI13 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	$1A_h$	$33F_h$	VDI13 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	$1B_h$	340_h	VDI14 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	$1C_h$	341_h	VDI14 Terminal Logic Selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	$1D_h$	342_h	VDI15 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
	1E$_h$	343$_h$	VDI15 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
	1F$_h$	344$_h$	VDI16 terminal function selection	uint16	0~39	RW	NO	0	1	Suspension in effect
	20$_h$	345$_h$	VDI16 terminal logic selection	uint16	0~1	RW	NO	0	1	Suspension in effect
2031$_h$ (communication given relevant variables)	01$_h$	373$_h$	Communication Given VDI Virtual Levels	uint16	0~65535	RW	NO	0	1	Suspension in effect
			Specific description: 2031$_h$: 01$_h$ bit(n)=1 means VDI(n+1) is high; bit(n)=0 means VDI(n+1) is low							
	02$_h$	374$_h$	Communication given Do output state	uint16	0~65535	RW	NO	0	1	Suspension in effect
			2004$_h$ When the logic selection of is negative, 2031$_h$: 02$_h$ =1,Do outputs low level; 2031$_h$: 02$_h$ =0,Do outputs high level							

10.3 Sub-Protocol Definition Parameter 6000h Group Description

Table 10-7

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
603F$_h$	00$_h$	37F$_h$	Error code, see section 8.1 for details	uint16	0~65535	RO	TPDO	0	-	-
6040$_h$	00$_h$	380$_h$	control letter	uint16	0~65535	RW	RPDO	0	1	Suspension in effect
6041$_h$	00$_h$	381$_h$	status word	uint16	0~65535	RO	TPDO	0	1	-
6042$_h$	00$_h$	382$_h$	Target speed for VM mode	int16	-3000~3000	RW	RPDO	0	1rpm	Suspension in effect
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	- - 2147483648~2147483647	RO	TPDO	0	1rpm	-
6046$_h$	01$_h$	385$_h$	Minimum speed for VM mode	uint32	0~6000	RW	RPDO	10	1rpm	Suspension in effect
	02$_h$	387$_h$	Maximum speed for VM mode	uint32	0~6000	RW	RPDO	3000	1rpm	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6048h	01h	389h	Acceleration in VM mode = delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Suspension in effect
	02h	38Bh		uint16	1~65535	RW	RPDO	1	1s	Suspension in effect
6049h	01h	38Ch	Deceleration of VM mode = delta speed/delta time	uint32	0~4294967295	RW	RPDO	500	1rpm	Suspension in effect
	02h	38Eh		uint16	1~65535	RW	RPDO	1	1s	Suspension in effect
604Ah	01h	38Fh	Emergency stop speed in VM mode = delta speed/delta time	uint32	0~4294967295	RW	RPDO	800	1rpm	Suspension in effect
	02h	391h		uint16	1~65535	RW	RPDO	1	1s	Suspension in effect
604Ch	01h	394h	Scaling factor for VM mode	int32	-2147483648~2147483647	RW	RPDO	0	1	Suspension in effect
	02h	396h		int32	-2147483648~2147483647	RW	RPDO	0	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
605A$_h$	00$_h$	3BF$_h$	Quick stop mode selection, see section 3.1.4 for details	int16	0~2	RW	NO	2	1	Suspension in effect
605Bh	00	3BDh	Shutdown mode selection, see section 3.1.4 for details	int16	0~1	RW	NO	0	1	Suspension in effect
605Ch	00	3BEh	Disablement operation shutdown mode selection, see section 3.1.4 for details	int16	0~1	RW	NO	0	1	Suspension in effect
605D$_h$	00$_h$	3C0$_h$	Pause mode selection, see section 3.1.4 for details	int16	1~2	RW	NO	1	1	Suspension in effect
605E$_h$	00$_h$	3C1$_h$	Failure shutdown mode selection, see section 3.1.4 for details	int16	0~2	RW	NO	0	1	Suspension in effect
6060$_h$	00$_h$	3C2$_h$	Mode selection, see section 5.1 for details	int8	0~10	RW	RPDO	1	1	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6061h	00h	3C3h	Operating mode display, see section 5.2 for details	int8	0~10	RO	TPDO	1	1	-
6062h	00h	3C4h	Target position command value currently in effect within the drive	int32	-2147483648~2147483647	RO	TPDO	0	user unit	-
6063h	00h	3C6h	Feedback on the current absolute position of the motor	int32	-2147483648~2147483647	RO	TPDO	0	Encoder Unit	-
6064h	00h	3C8h	Feedback on the current absolute user position of the motor	int32	-2147483648~2147483647	RO	TPDO	0	user unit	-
			Specific description: Position feedback 6064h × Gear ratio (6091h) = Position feedback 6063h							
6065h	00h	3CAh	Positional deviation oversize threshold	uint32	0~4294967295	RW	RPDO	50	user unit	Suspension in effect
			Specific description: When the position deviation exceeds ±6065h and the time reaches 6066h , a tracking error is reported.							
6066h	00h	3CCh	Excessive position deviation timeout	uint16	0~65535	RW	RPDO	30000	1ms	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			Specific description: When the position deviation exceeds ±6065h and the time reaches 6066h , a tracking error is reported.							
6067h	00h	3CDh	Position reaches threshold	uint32	0~4294967295	RW	RPDO	10	user unit	Suspension in effect
			Specific description: Position deviation within ±6067h and time up to 6068h is considered as position arrival, bit10=1 of status word 6041h in position class mode							
6068h	00h	3CFh	position arrival time window	uint16	0~65535	RW	RPDO	5	1ms	Suspension in effect
			Specific description: the position is considered to be reached when the position deviation is within ±6067h and the time reaches 6068h . In position class mode, bit10=1 in status word 6041 h							
6069h	00h	3D0h	Feedback value of the speed sensor	int32	-2147483648~2147483647	RO	TPDO	0	User unit/s	-
606Bh	00h	3D3h	Speed command value validated within the controller	int32	-2147483648~2147483647	RO	RPDO	0	User unit/s	-
606Ch	00h	3D5h	Current actual speed feedback value	int32	-2147483648~2147483647	RO	TPDO	0	rpm	-
606Dh	00h	3D7h	Velocity reaches threshold	uint16	0~65535	RW	RPDO	500	rpm	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			Specific description: the difference between the target speed 60FF$_h$ (converted to motor speed rpm) and the actual motor speed is within ±606D, and when the time reaches 606E$_h$, it is considered that the speed is reached, and bit10=1 of the status word 6041 $_h$							
606E$_h$	00$_h$	3D8$_h$	Velocity Arrival Time Window	uint16	0~65535	RW	RPDO	5	ms	Suspension in effect
			Specific description: the difference between the target speed 60FF$_h$ (converted to motor speed rpm) and the actual motor speed is within ±606D, and when the time reaches 606E$_h$, it is considered that the speed is reached, and bit10=1 of the status word 6041 $_h$							
606F$_h$	00$_h$	3D9$_h$	Zero reaches threshold	uint16	0~65535	RW	RPDO	5	User unit/s	Suspension in effect
			Specific description: the current speed 6069$_h$ is within ±606F, and the time reaches 6070$_h$, it is considered that the zero speed arrives, the status word 6041$_h$ bit12=1							
6070$_h$	00$_h$	3DA$_h$	zero degree arrival time window	uint16	0~65535	RW	RPDO	5	ms	Suspension in effect
			Specific description: the current speed 6069$_h$ is within ±606F, and the time reaches 6070$_h$, it is considered that the zero speed arrives, the status word 6041$_h$ bit12=1							
6071$_h$	00$_h$	3DB$_h$	Target torque	int16	-1000~1000	RW	RPDO	0	0.10%	Suspension in effect
			Specific description: 100.0% corresponds to 1 times the rated torque of the motor.							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6072h	00h	3DCh	Maximum torque	uint16	0~2000	RW	RPDO	1000	0.10%	Suspension in effect
6073h	00h	3DDh	Maximum current	uint16	0~2000	RW	NO	1500	0.10%	Suspension in effect
6074h	00h	3DEh	Torque command, reflecting the torque command entered in the enable state.	int16	-32768~32767	RO	TPDO	0	0.10%	-
6075h	00h	3DFh	rated current	uint32	0~11700	RW	RPDO	11700	0.001A	Suspension in effect
6076h	00h	3E1h	Rated torque	uint32	0~1270	RW	RPDO	1270	0.01N.m	Suspension in effect
6077h	00h	3E3h	Torque feedback value	int16	-32768~32767	RO	TPDO	-	0.10%	-
6078h	00h	3E4h	Current feedback value (percentage corresponding to rated current)	int16	-32768~32767	RO	TPDO	-	0.10%	-
607Ah	00h	3E7h	target location	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
607C$_h$	00$_h$	3ED$_h$	origin offset value	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect
			Specific description: Set the physical position of the mechanical zero point deviation from the motor home position in the home return mode; effective when bit15=1 of status word 6041$_h$ when home return operation is completed							
607D$_h$	01$_h$	3EF$_h$	Software absolute position minimum limit	int32	-2147483648~2147483647	RW	RPDO	-65535	user unit	Suspension in effect
	02$_h$	3F1$_h$	Maximum software absolute position limit	int32	-2147483648~2147483647	RW	RPDO	65535	user unit	Suspension in effect
607E$_h$	00$_h$	3F3$_h$	Command polarity, see section 4.3 for details	uint8	0~255	RW	NO	0	1	Suspension in effect
607F$_h$	00$_h$	3F4$_h$	Maximum profile speed	uint32	0~1000000	RW	RPDO	500000	User unit/s	Suspension in effect
6080$_h$	00$_h$	3F6	Maximum motor speed	uint32	0~6000	RW	RPDO	4000	rpm	Suspension in effect
6081$_h$	00$_h$	3F8$_h$	Profile speed	uint32	0~1000000	RW	RPDO	500000	User unit/s	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
			Specific description: Set the uniform running speed of the segment displacement command in contour position mode, which is calculated as follows: motor speed (rpm) $=\frac{6081h\times 齿轮比\ 6091h}{编码器分辨率}\times 60$							
6082h	00h	3FAh	Contour End Velocity	uint32	0~1000000	RW	RPDO	0	User unit/s	Suspension in effect
6083h	00h	3FCh	contour acceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Suspension in effect
6084h	00h	3FEh	contour deceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s^2	Suspension in effect
6085h	00h	400h	Quick stop deceleration	uint32	0~4294967295	RW	RPDO	500000	User unit/s^2	Suspension in effect
6086h	00h	402h	Contour Slope Type	int16	0~3	RW	RPDO	3	1	Suspension in effect
			Specific description: Sets the type of curve for motor position commands or speed commands. 0- Linear, 3- S-curve							
6087h	00h	403h	Torque Ramp	uint32	0~4294967295	RW	RPDO	10	0.10%	Suspension in effect
			Specific description: Setting the torque command acceleration in profile torque mode, the meaning of which is: torque command increment per second							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
6088h	00h	405h	Torque Ramp Type	int16	0~2	RW	RPDO	0	1	Suspension in effect
			Specific descriptions: 0- slopes, 2- no slopes							
608Fh	01h	406h	Encoder increment	uint32	0~4294967295	RW	NO	10000	1	Suspension in effect
	02h	408h	Motor RPM	uint32	0~4294967295	RW	NO	1	1	Suspension in effect
6091h	01h	40Eh	gear ratio	uint32	0~4294967295	RW	NO	1	1	Suspension in effect
	02h	410h		uint32	0~4294967295	RW	NO	1	1	Suspension in effect
6098h	00h	416h	Origin regression method, see section 5.7.4 for details	int8	17~45	RW	RPDO	20	1	Suspension in effect
6099h	01h	417h	Find switching speed (high speed)	uint32	0~4294967295	RW	RPDO	10000	User unit/s	Suspension in effect
6099h	02h	419h	Finding home speed (low speed)	int32	-2147483648~2147483647	RW	RPDO	2730	User unit/s	Suspension in effect

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
609A$_h$	00$_h$	41B$_h$	(math.) origin regression acceleration	uint32	0~4294967295	RW	RPDO	409600	User unit/s	Suspension in effect
60A3$_h$	00$_h$	41D$_h$	Number of contours plus acceleration used, only 60A4$_h$:01$_h$ and 60A4$_h$:02 are used in this device.$_h$	Usint8	2	RW	RPDO	2	1	Suspension in effect
60A4$_h$	01$_h$	41E$_h$	Contour plus acceleration	uint32	0~4294967295	RW	NO	50000	User unit/s^2	Suspension in effect
	02$_h$	420$_h$	Contour Deceleration	uint32	0~4294967295	RW	NO	50000	User unit/s^2	Suspension in effect
60B0$_h$	00$_h$	426$_h$	position bias	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect
			Specific description: Set the position instruction bias amount in cycle synchronized position mode, after bias: target position = 607A$_h$ +60B0$_h$							
60B1$_h$	00$_h$	428$_h$	Speed Bias	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect
			Specific description: set the speed command bias amount in cycle synchronized speed mode, after bias: target speed = 60FF$_h$ +60B1$_h$							

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
60B2h	00h	42Ah	Torque Bias	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect
			Specific description: Set the torque command bias amount in cycle synchronized torque mode, after bias: target torque = 6071h +60B2h							
60C1h	01h	42Dh	Target position for interpolation	int32	-2147483648~2147483647	RW	RPDO	0	user unit	Suspension in effect
60C2h	01h	42Fh	The interpolation cycle time constant t	uint8	0~255	RW	NO	20	s	Suspension in effect
	02h	430h	The interpolation cycle time index n	Int8	-128~127	RW	NO	-1	1	Suspension in effect
60C5h	00h	43Bh	Maximum profile acceleration	uint32	0~4294967295	RW	RPDO	500000	User unit/s2	Suspension in effect
60C6h	00h	43Dh	Maximum contour deceleration	uint32	0~4294967295	RW	RPDO	500000	User unit/s^2	Suspension in effect
60F2h	00h	43Fh	Positioning Options	uint16	0~2	RW	RPDO	0	1	Suspension in effect
			Specific descriptions are as follows.							

<table>
<tr><th>Index</th><th>Sub-index</th><th>Name</th><th>Description</th><th>Data type</th><th>Data range</th><th>Access-ibility</th><th>Mapp-ability</th><th>Factory setting</th><th>Unit</th><th>Activate method</th></tr>
<tr><td></td><td></td><td></td><td colspan="9"><table>
<tr><th>Bit1</th><th>Bit0</th><th>descriptive</th></tr>
<tr><td>0</td><td>0</td><td>Relative position shift, relative to target position (607A_h)</td></tr>
<tr><td>0</td><td>1</td><td>Relative position shift, relative to the entered position command (60FC_h)</td></tr>
<tr><td>1</td><td>0</td><td>Relative position shift, relative to the entered 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 bias</td><td>int32</td><td>-2147483648~2147483647</td><td>RO</td><td>TPDO</td><td>0</td><td>user unit</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>-2147483648~2147483647</td><td>RO</td><td>TPDO</td><td>0</td><td>unit of command (computing)</td><td>-</td></tr>
<tr><td rowspan="2">60FC_h</td><td rowspan="2">00_h</td><td>446_h</td><td>Motor Position Command</td><td>int32</td><td>-2147483648~2147483647</td><td>RO</td><td>TPDO</td><td>0</td><td>user unit</td><td>-</td></tr>
<tr><td colspan="9">Specific description: Position command 60FC_h = Position command 6062_h × electronic gear ratio (6091)_h</td></tr>
<tr><td>60FD_h</td><td>00_h</td><td>44C_h</td><td>Input signal (DI signal) monitoring</td><td>uint32</td><td>0~4294967295</td><td>RO</td><td>TPDO</td><td>0</td><td>1</td><td>-</td></tr>
<tr><td rowspan="2">60FE_h</td><td>01_h</td><td>44E_h</td><td>Physical output</td><td>uint32</td><td>0~4294967295</td><td>RO</td><td>TPDO</td><td>0</td><td>1</td><td>-</td></tr>
<tr><td>02_h</td><td>450_h</td><td>Physical output enable</td><td>uint32</td><td>0~4294967295</td><td>RO</td><td>TPDO</td><td>0</td><td>1</td><td>-</td></tr>
</table>

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
60FF$_h$	00$_h$	448$_h$	target speed	int32	-2147483648~2147483647	RW	RPDO	0	User unit/s	Suspension in effect
			Specific description: Setting the user speed command in contour speed mode and cycle synchronized speed mode							

11 Version Information

Manual Version	Date	Revision History
A	2025/8/13	Establish
A01	2026/1/15	Modify description
A02	2026/6/10	Modify description of default

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

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

