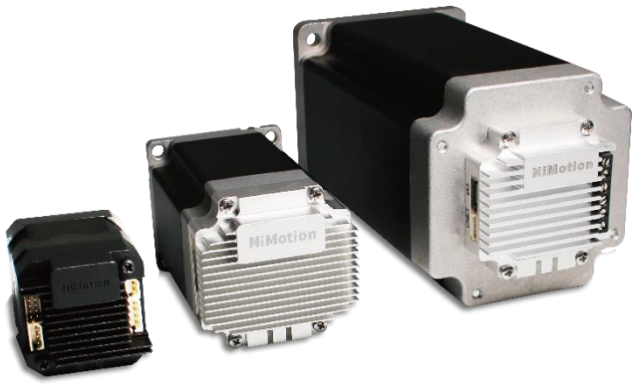
Integrated Low-voltage Stepper Motor

CANopen Communication User Manual

版 本 号：A01



Beijing NiMotion Control Technology Co., Ltd.

Contents

1 About the Manual

1.1 Introduction

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

1.2 The Meaning of Numerical Values

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

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

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

1.3 Bit Definition

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

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

$55_{hex}=85_{dec}$

2 Terms and Abbreviations

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

3 Communication Network

3.1 Overview of CANopen Protocol

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

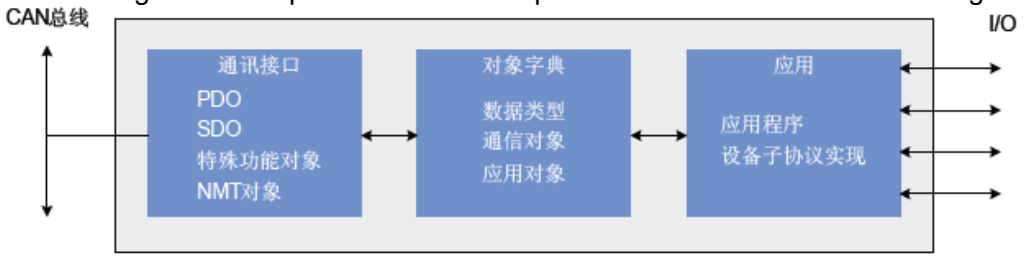


Figure 3-1

The communication objects are as follows:

Table 3-1

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

3.2 SDO Access

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

SDO Transport Framework

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

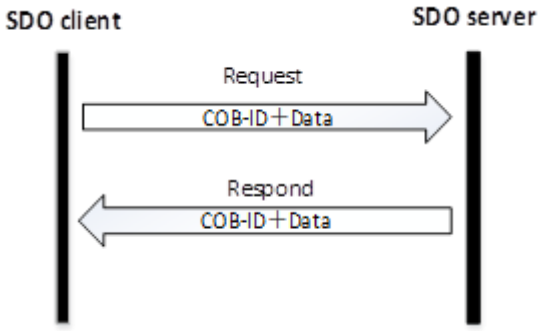


Figure 3-2 The SDO transmission model

3.2.1 SDO Transmission Message

SDO transmission message is composed of COB-ID and data segments. The data segment adopts little endian mode, i.e. the lower bit in the front, and the high bit in the back. All SDO

message data segments must be 8-bytes long.

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

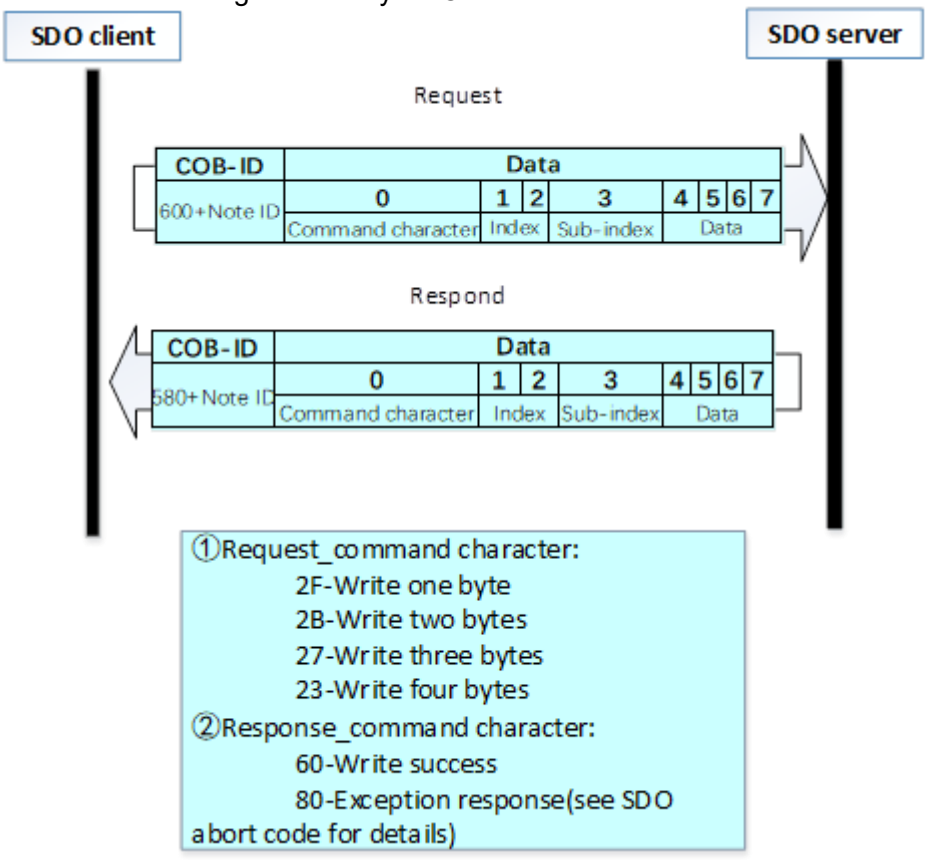


Figure 3-3

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

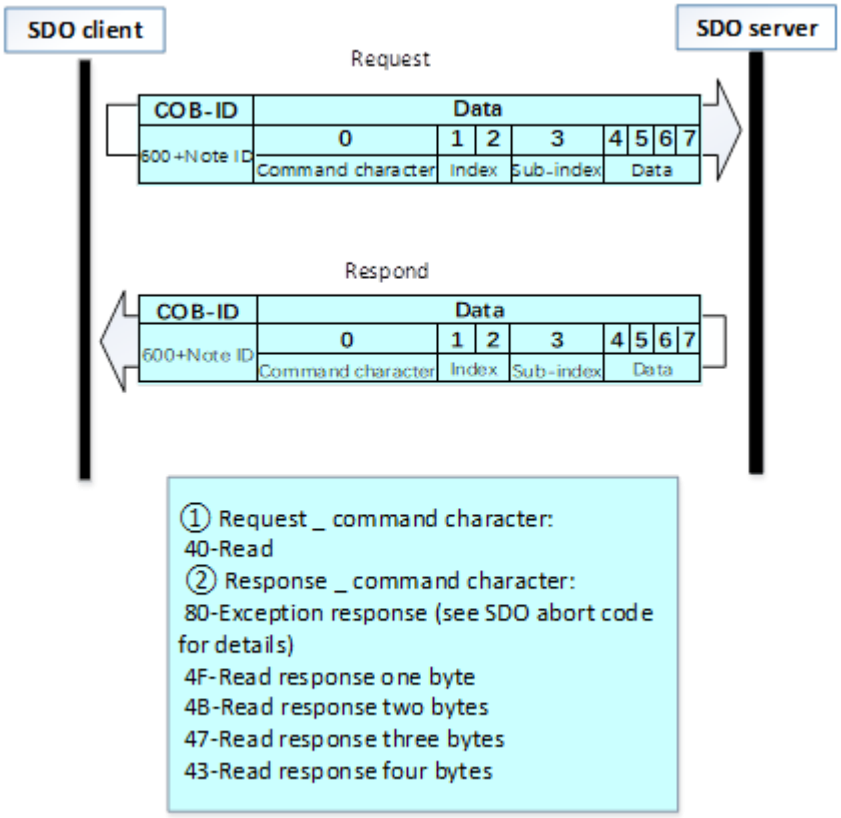


Figure 3-4

3.2.2 SDO Error Message

When an error occurs during reading or writing (for example, due to the object dictionary does

not exist), the controller will not confirm it, but send a error report . As shown in the figure below, the object in 6041_h is read-only, when the object is write, it returns an error code of 0x06010002 (the lower bits first, the higher bits second).

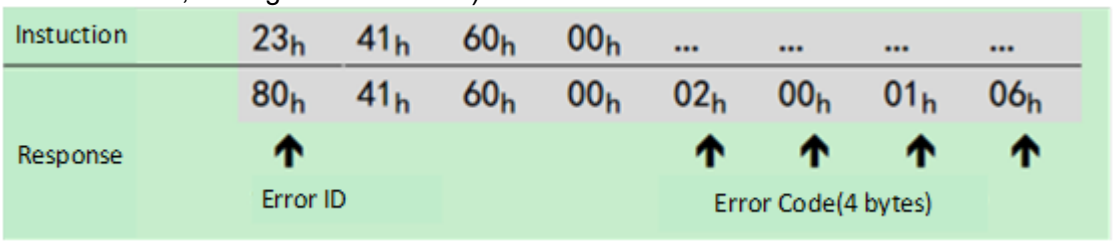


Figure 3-5 Examples of SDO error reports

SDO abnormal response aborted code (Hex):

Table 3-2

Interrupt Code	Description
05 03 00 00	Trigger bit does not change alternately
05 04 00 00	SDO protocol timeout
05 04 00 01	Illegal or unknown Client / Server, command word
05 04 00 02	Invalid block size (Block transfer mode only)
05 04 00 03	Invalid sequence number (Block transfer mode only)
05 04 00 04	CRC error (Block transfer mode only)
05 04 00 05	Out of memory
06 01 00 00	Object does not addressable
06 01 00 01	Attempt to read write-only objects
06 01 00 02	Attempt to write read-only object
06 02 00 00	Object does not exist in object dictionary
06 04 00 41	Object cannot be mapped to PDO
06 04 00 42	Number or length of objects exceeds PDO length
06 04 00 43	General parameters incompatible
06 04 00 47	General equipment internal incompatibility
06 06 00 00	Object access failed due to hardware error
06 07 00 10	Data type mismatch, service parameter length mismatch
06 07 00 12	Data type mismatch, service parameter length is too large
06 07 00 13	Data type mismatch, service parameter length is too short
06 09 00 11	Sub-index does not exist
06 09 00 30	Out of parameter value range
06 09 00 31	Write parameter value is too large (during writing)
06 09 00 32	Write parameter value is too small (during writing)
06 09 00 36	Max value less than min value
08 00 00 00	General error (e.g. bytes number wrong)
08 00 00 20	Data cannot be transferred or saved to the application program
08 00 00 21	Data cannot be transferred or saved to the application due to local control
08 00 00 22	Data cannot be transferred or saved to the application due to the current status of the device
08 00 00 23	The object dictionary dynamically generates an error or does not exist

3.3 PDO Message

Process data object (PDO) is used to transmit real-time data, which is the most important data transmission mode in CANopen. Through PDO, data can be transferred in the way of event-

control or periodic. Because PDO transmission does not need response, and PDO length can be less than 8-bytes, so the transmission is fast.

3.3.1 PDO Transmission Framework

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

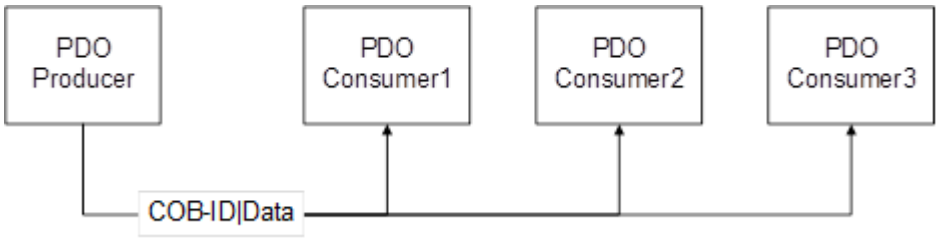


Figure 3-6 PDO Transmission model

3.3.2 PDO Object

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

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

3.3.3 PDO Communication Parameters

1）PDO Identifier COB-ID

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

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

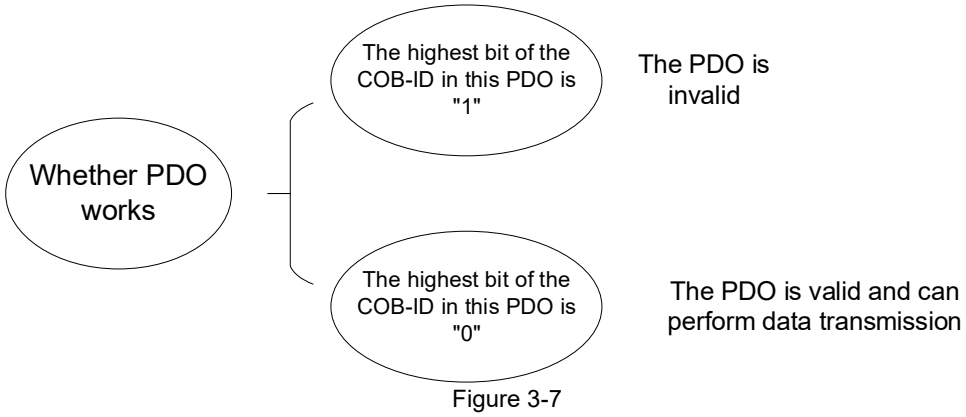


Figure 3-7

2）Transmission type of PDO

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

mode.

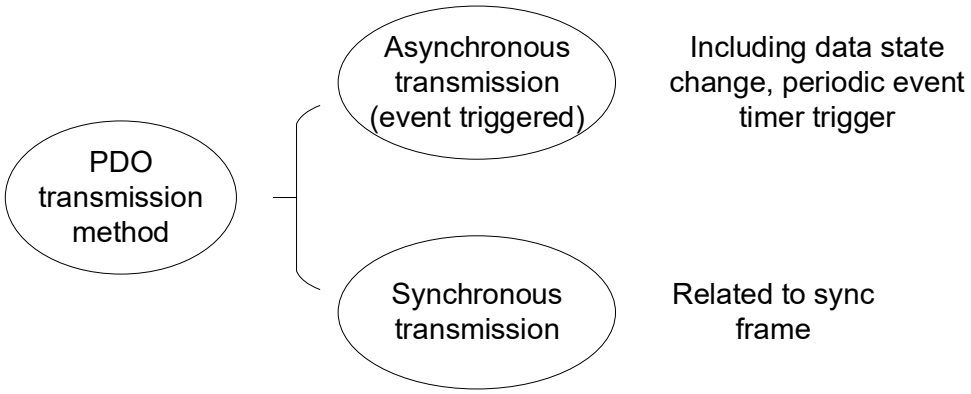


Figure 3-8 PDO transmission method

The transmission trigger type is as follows:

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

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

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

  When the transport type of TPDO is 254 or 255, the TPDO is sent when the mapping data changes or the event timer arrives.

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

3）**Prohibition Time**

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

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

4）**Event Timer**

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

3.3.4 PDO Mapping

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

Table 3-3

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

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

Table 3-4

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

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

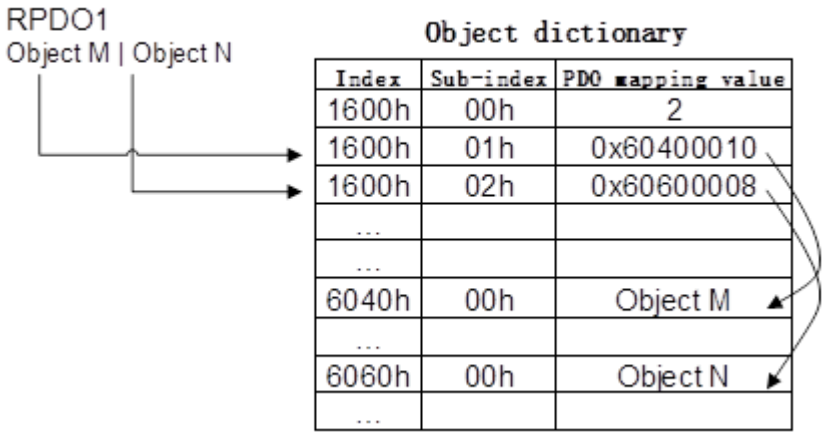


Figure 3-9 RPDO1 mapping relationship

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

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

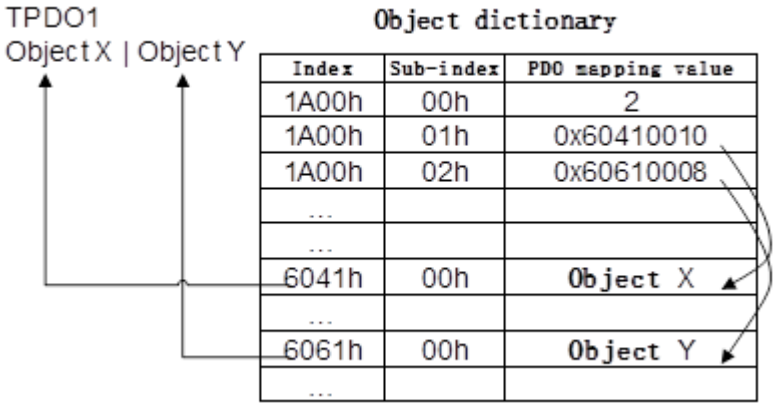


Figure 3-10

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

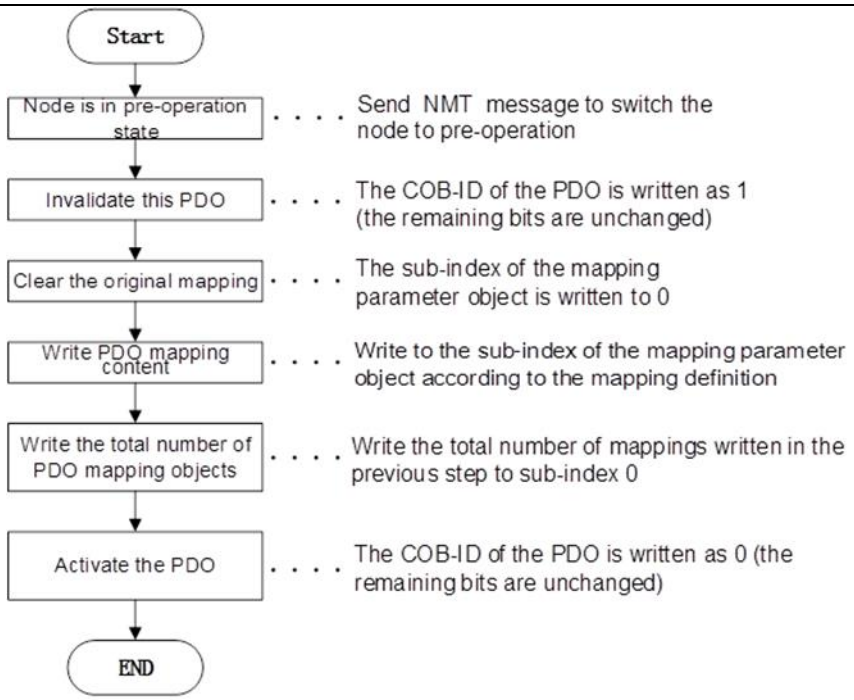


Figure 3-11

3.4 EMCY Message

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

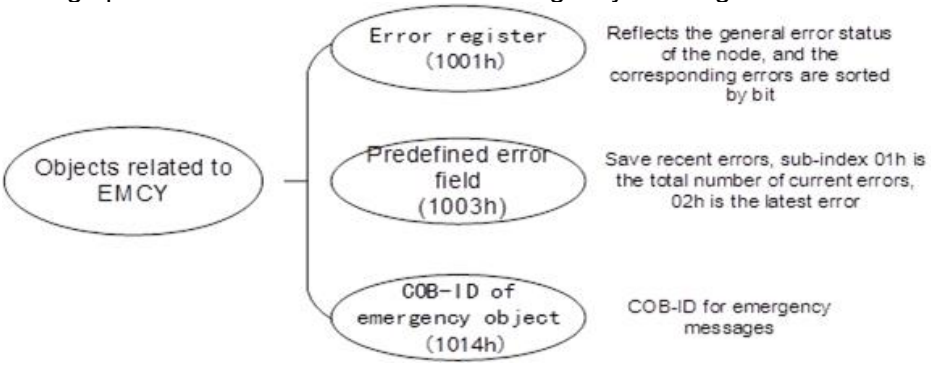


Figure 3-12

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

Table 3-5

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

Auxiliary byte displays the user specified error code. Please refer to "9.1 Fault management" for details of error code and auxiliary byte definition.

The error register object is 1001h, and if a particular error occurs, the corresponding position 1, any occurrence of the error will set the general error. The meaning of bits is as follows:

1001_h

Bit	7	6	5	4	3	2	1	0
Description	Manufacturer's designation	-	-	Communication error	Temperature	Voltage	Current	General error

3.5 Network Management (NMT Service)

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

3.5.1 NMT Services

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

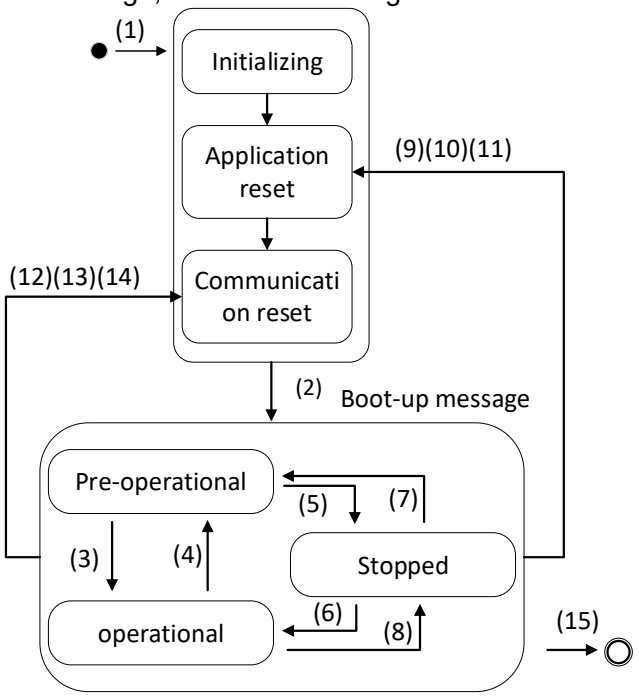


Figure 3-13

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

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

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

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

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

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

Table 3-6

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

After the device is powered on, it will automatically switches the Initializing state, including

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

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

Table 3-7

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

3.5.2 Error Control

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

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

Table 3-8

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

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

3.5.3 Start-up Agreement

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

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

Table 3-9

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

3.6 System Setting

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

3.6.1 Baud-Rate Setting

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

Table of setting value and baud-rate:

Table 3-10

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

3.6.2 Node ID

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

3.6.3 Connection of Communication Line

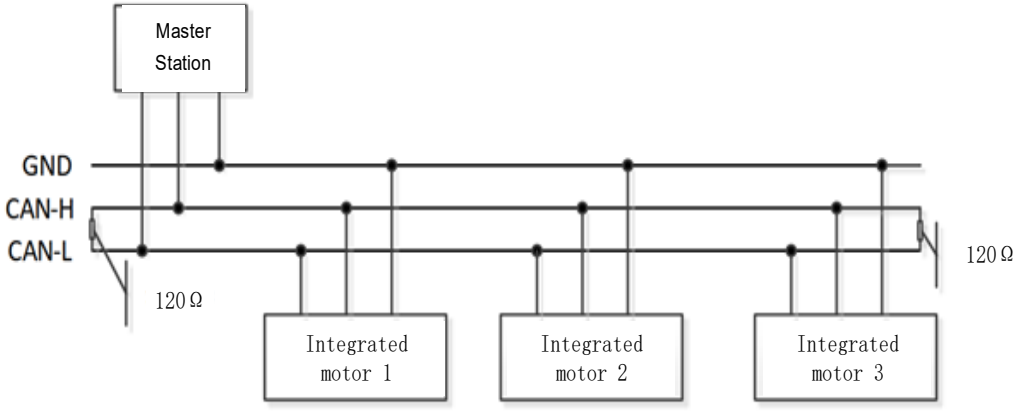


Figure 3-14

4 Control of Equipment

4.1 The CiA402 State Machine

The operating state of integrated low-voltage 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, and the actual state of the device can be read back from the status word object 6041_h.

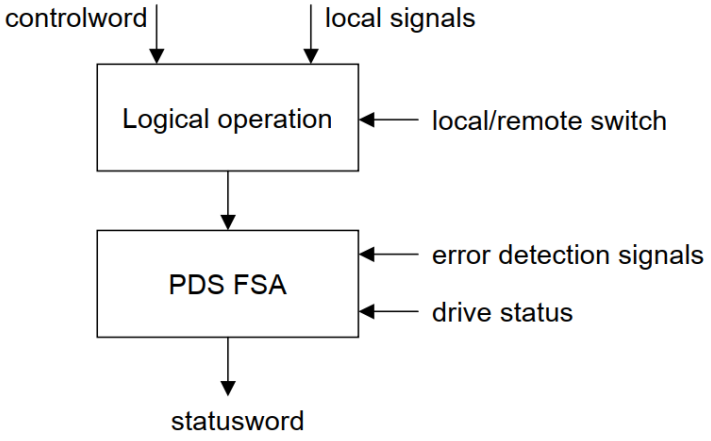


Figure 4-1

4.1.1 The CiA402 State Machine

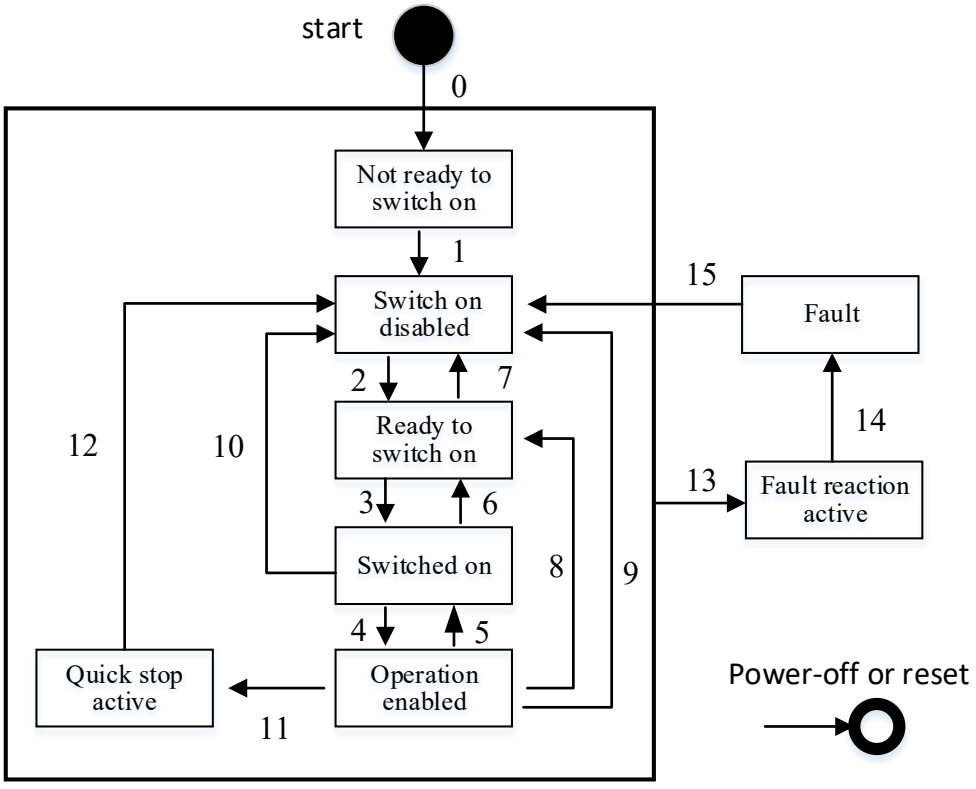


Figure 4-2

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

Table 4-1

Function	FSA states							
	Not ready to switch on	Switch on disabled	Ready to switch on	Switched on	Operation enabled	Quick stop active	Fault reaction active	Fault
Brake applied, if present	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

Low-level power applied	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High-level power applied	Yes/No	Yes/No	Yes/No	Yes	Yes	Yes	Yes	Yes/No
Drive function enabled	No	No	No	No	Yes	Yes	Yes	No
Configuration allowed	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

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

Table 4-2

Command	Bits of the *controlword*					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	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	X	X	0	X	7,9,10,12
Quick stop	0	X	0	1	X	7,10,11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4,16
Fault reset	↑	X	X	X	X	15
NOTE Automatic transition to Enable operation state after executing SWITCHED ON state functionality.						

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

The status word in each state are as follows:

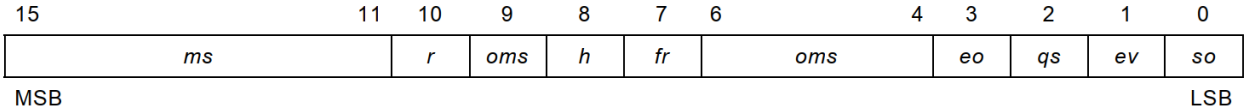
Table 4-3

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

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

4.1.2 Control Word

The bits of the control word are defined as follows:



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

Notes：

It is meaningless to assign each bit of the control word individually, It must be combined with other bits

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

4.1.3 Status Word

The bits of the status word are defined as follows:

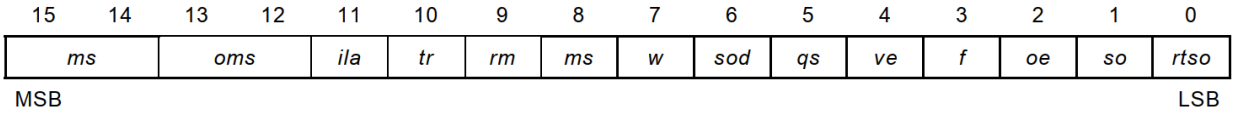


Table 4-4

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

Notes：

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

4.1.4 The Stop Mode

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

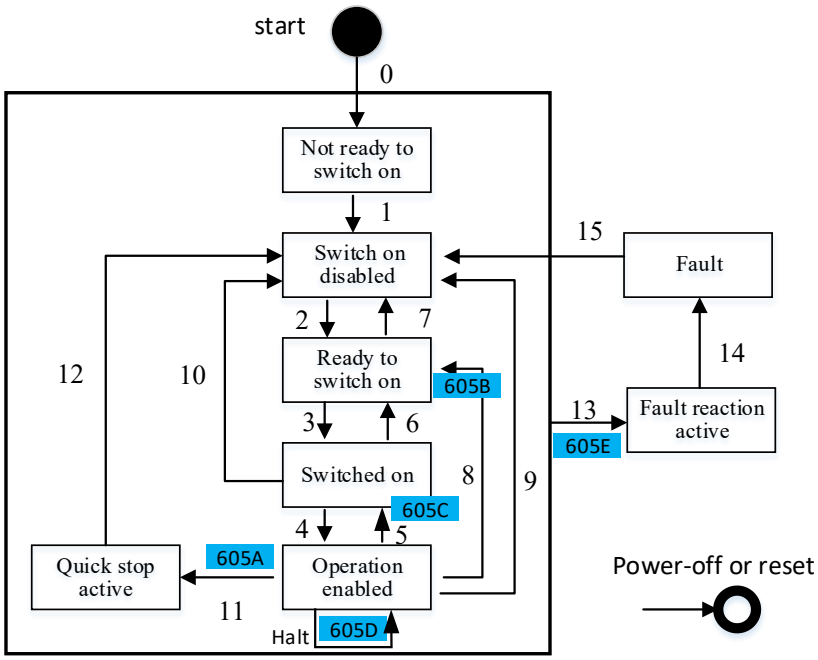


Figure 4-3

- **Fault**

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

Table 4-5

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

- **Quick Stop**

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

Table 4-6

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

- **Halt**

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

Table 4-7

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

- **Disable Operation**

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

Table 4-8

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

- **Shutdown**

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

Table 4-9

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

5 Unit Conversion

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

5.1 608Fh: Resolution of position encoder

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

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

The object description is specified in the table below:

Table 5-1

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

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

Table 5-2

Attribution	Description
Sub-index	01h
Description	Encoder increments
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	4000
Sub-index	02h
Description	Motor revolutions
Accessibility	rw
Mappability	No
Data type	Unsigned32
Default value	1

5.2 6091h:Transmission Ratio

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

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

The object description is specified in the table below:

Table 5-3

Attribution	Description
Index	6091h
Name	Gear ratio
Data structure	Array

Data type	Unsigned32

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

Table 5-4

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

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

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

  shaft position feedback=drive shaft position feedback × Gear ratio

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

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

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

- $$\text{motor shaft acceleration}=\frac{\text{acceleration of shaft}\times\text{gear ratio(6091h)}}{\text{encoder resolutions}}\times 60$$

5.3 $607E_h$:Polarity

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

Table 5-5

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

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

Table 5-6

Attribution	Description
Index	$607E_h$
Name	Polarity

Data structure	Variable
Accessibility	rw
Data type	Unsigned8
Default value	0

5.4 Example

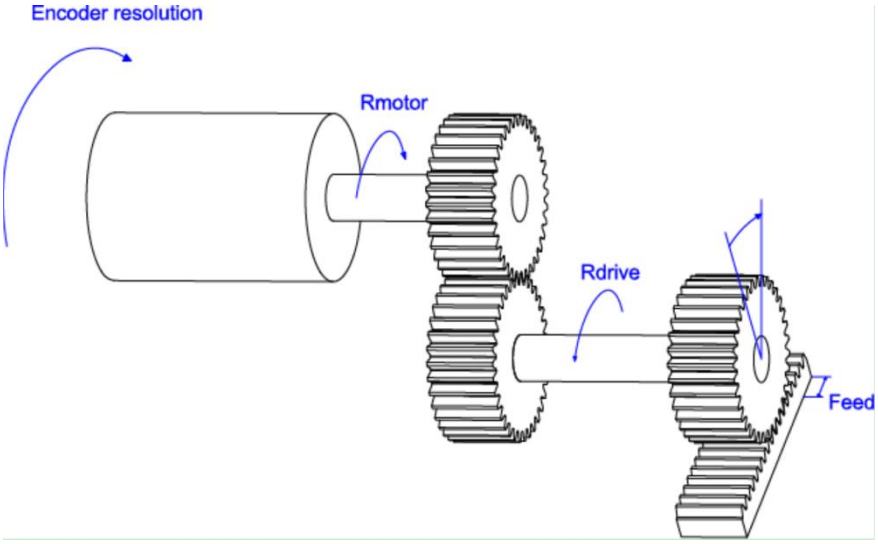


Figure 5-1

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

position encoder resolution $608F_h$:

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

gear ratio 6091_h:

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

6 Operation Mode

6.1 Overview of Operation Mode

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

Table 6-1

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

Sub index 01_h is to determine the mode:

Table 6-2

Sub-Index	01_h
Description	Control mode option
Accessibility	rw
PDO mapping	No
Data type	Unsigned8
Default value	0
Meaning of Value	0x00: CiA402 mode 0x01: Manufacturer position mode 0x02: Manufacturer velocity mode 0x03: Manufacturer torque mode 0x04: Manufacturer open loop mode

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

Table 6-3

Sub-Index	6060_h
Description	Mode Option
Accessibility	rw
PDO mapping	NO
Data type	Unsigned8
Default value	0x01
Sub-Index	-
Meaning of Value	0x00: Undefined 0x01: Profile position mode(PP) 0x02: Velocity mode (VM) 0x03: Profile velocity mode 0x04: Torque profile mode 0x05: Undefined 0x06: Homing mode (HM) 0x07: Interpolated position mode(IP) 0x08: Cyclic sync position mode(CSP) 0x09: Cyclic sync velocity mode (CSV) 0x0A: Cyclic sync Torque mode (CST)

6.2 Position Control Function

6.2.1 Structure Chart

Following error

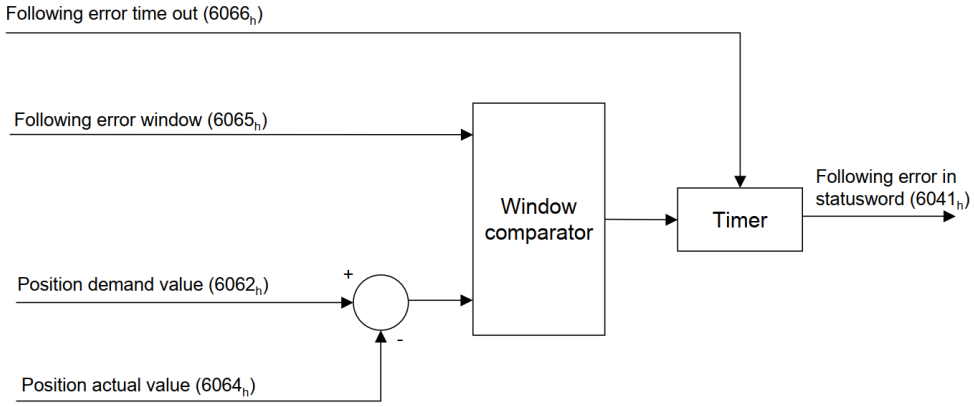


Figure 6-1

Position reached

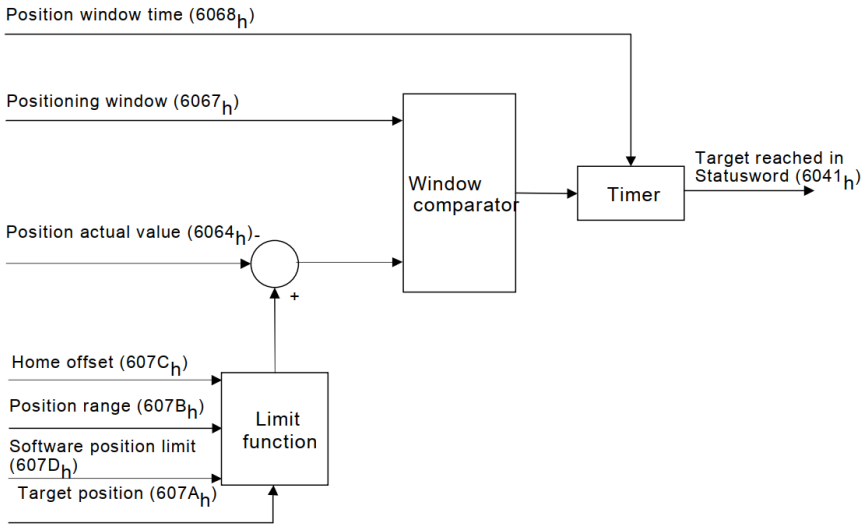


Figure 6-2

6.2.2 Objects related

Following error

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

Position Reached

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

6.2.3 Function Description

Target Following Error

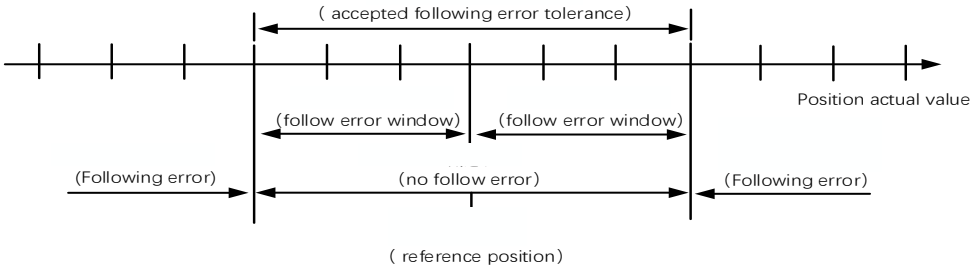


Figure 6-3

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

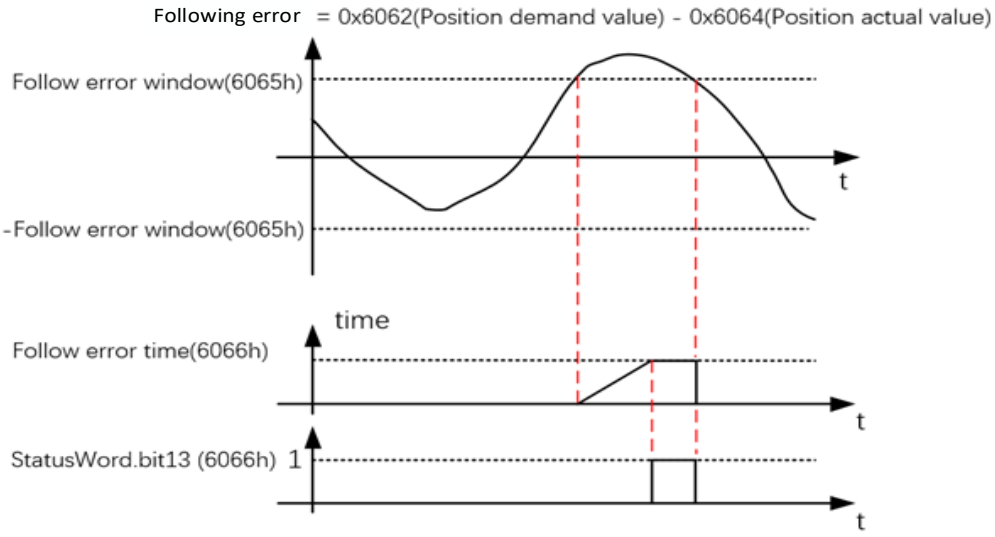


Figure 6-4

Target Reached

Figure of Error Acceptable:

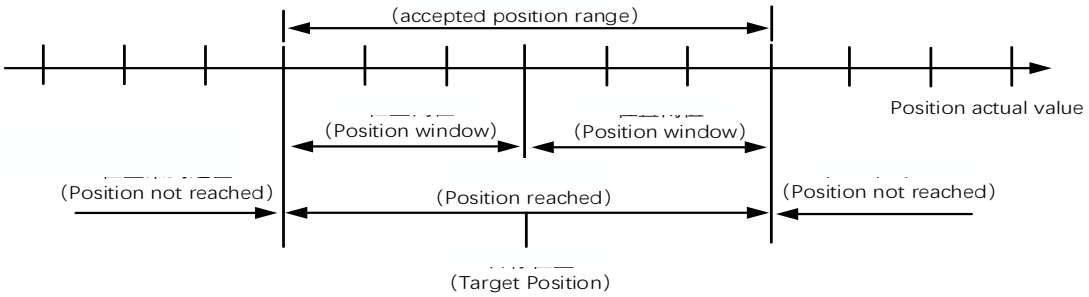


Figure 6-5

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

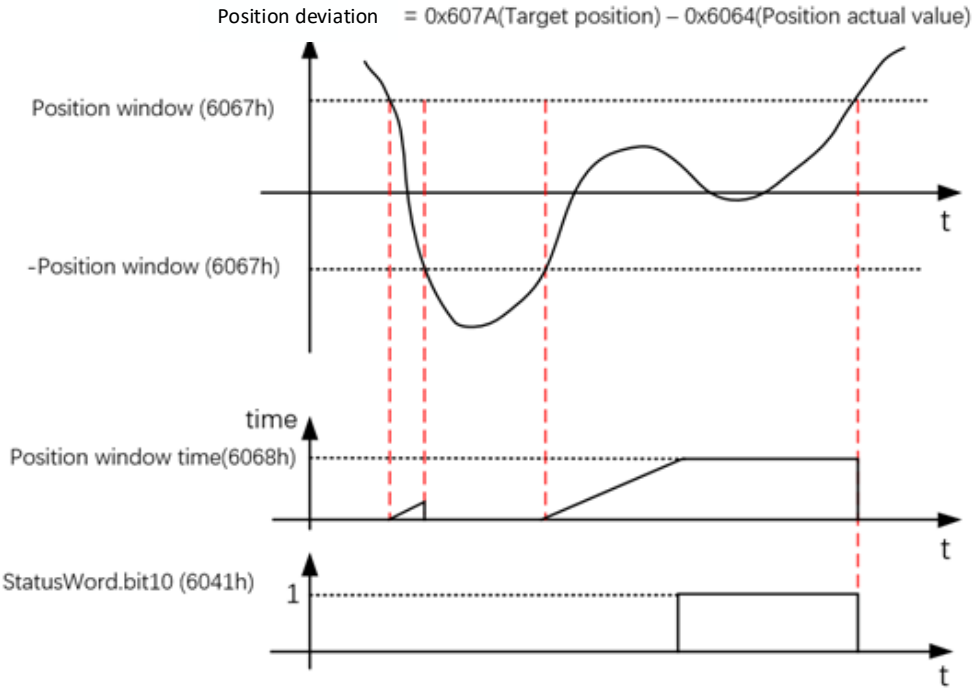


Figure 6-6

6.3 Profile Position Mode（PP）

The overall structure is shown in the figure below. The internal position curve is generated according to the target position, and the final position is generated after relevant conditions are restricted.

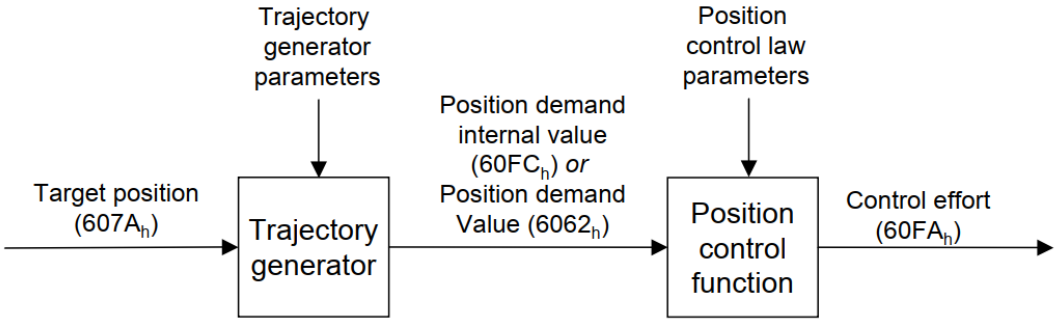


Figure 6-7

6.3.1 Structural Chart

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

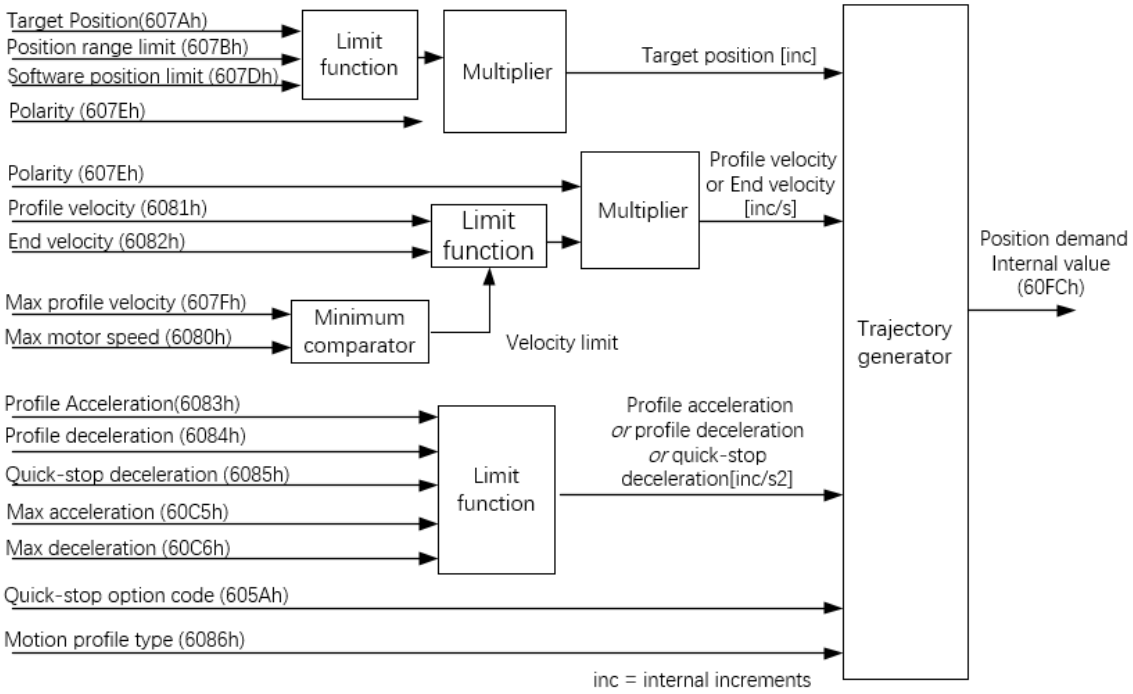


Figure 6-8

Note: In PP mode, the acceleration 6083_h is small, and when high speed accuracy is required, the configuration $608F_h$ needs to be adjusted (closed-loop configuration to 40000 and open-loop configuration to 16000).

6.3.2 Object Related

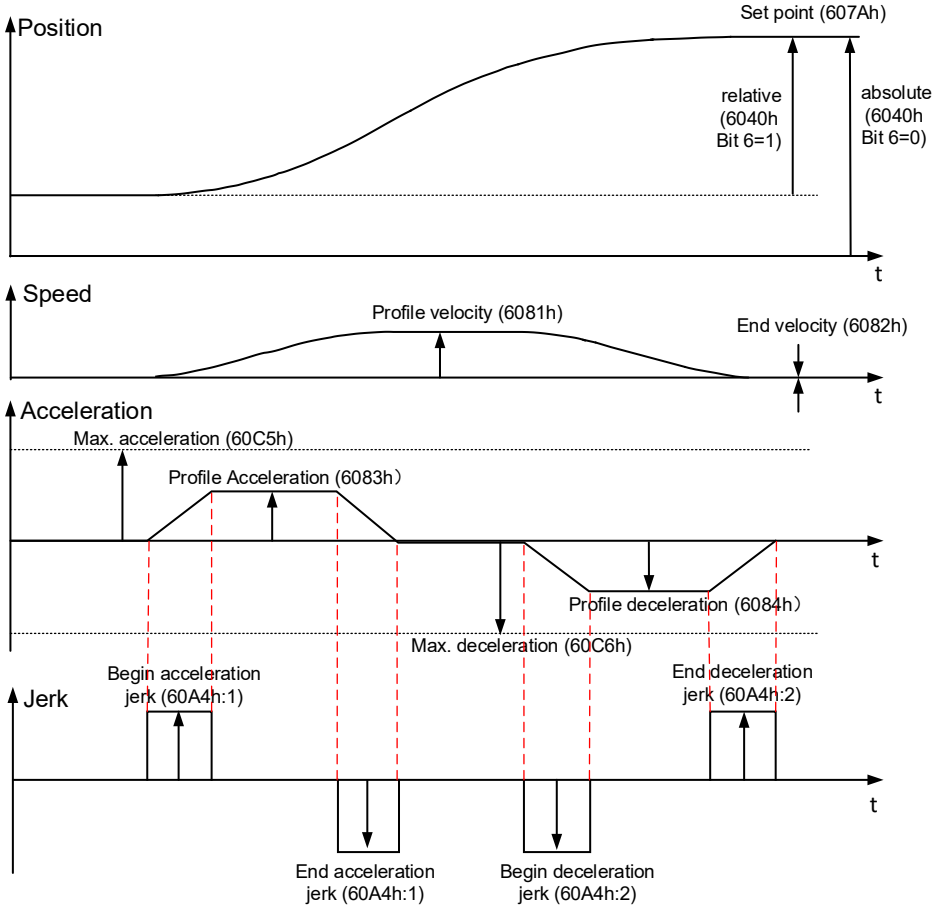
Pay attention to the following objects in this mode :

Table 6-4

Object Index	Description
$605D_h$ (Halt option code)	Halt option
$605A_h$ (Quick stop option code)	Quick stop option
6062_h (Position Demand Value)	Current position demand value inside the drive (user unit)
6063_h (Position Actual Enc Value)	Current position actual value feedback (encoder unit)
6064_h (Position Actual User Value)	Current actual position (user unit)
$607A_h$ (Target Position)	Target position (user unit)
$607B_h$ (Position Range Limit)	Position range limit (user unit)
$607D_h$ (Software Position Limit)	Target position limit (user unit)
$607E_h$ (Polarity)	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
$607F_h$ (Max Profile Velocity)	Maximum velocity (user unit /s), to limit the velocity
6080_h (Max motor speed)	Max motor speed(rpm)
6081_h (Profile Velocity)	Maximum velocity during uniform motion in this profile operation, (user unit / s), I.e. the velocity at the end of the acceleration slope during the positioning.
6082_h (End Velocity)	The velocity when reaching the target position(user unit/s), and the velocity at the end of the slope. Usually, this object is

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

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



6.3.3 Control Instruction and Status Information

Enable

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

Control Word

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

MSB LSB

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

Table 6-5

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

Table 6-8

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

Status Word

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

MSB LSB

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

Table 6-6

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

6.3.4 Function Description

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

1） Control Instruction Sequence- Immediate Update

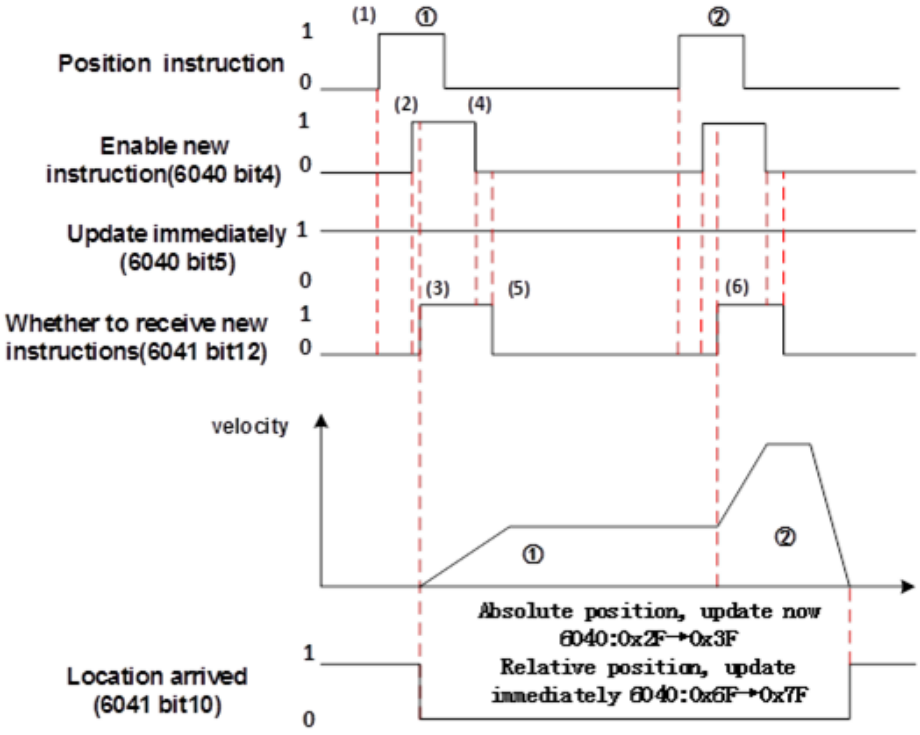


Figure 6-7

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

Table 6-7

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

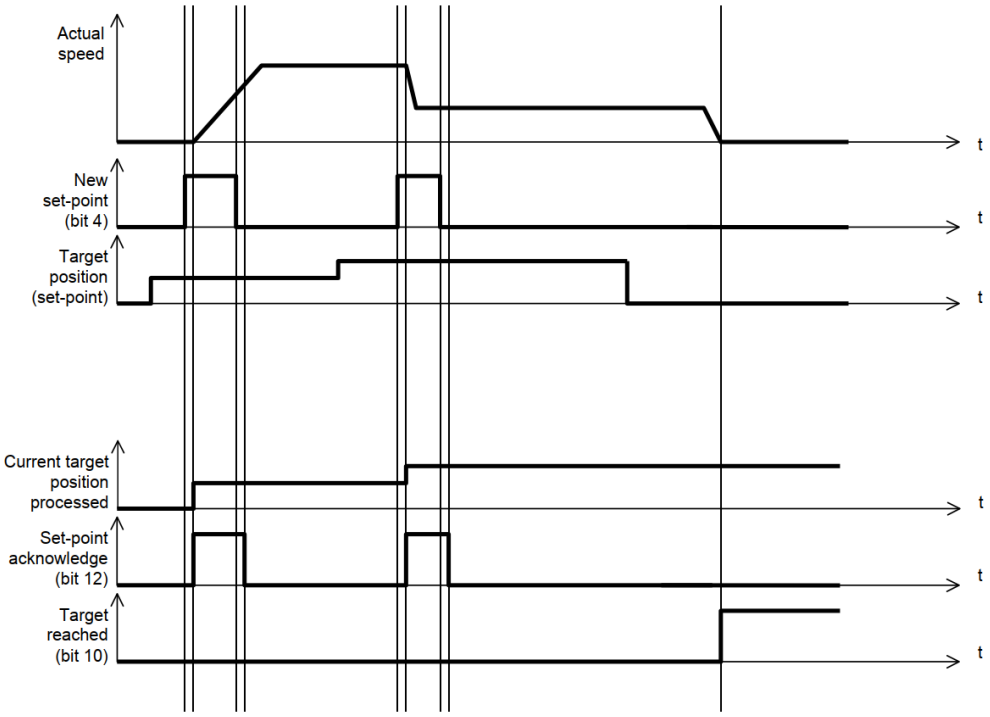


Figure 6-8

2） Control Instruction Sequence Non Immediate Update

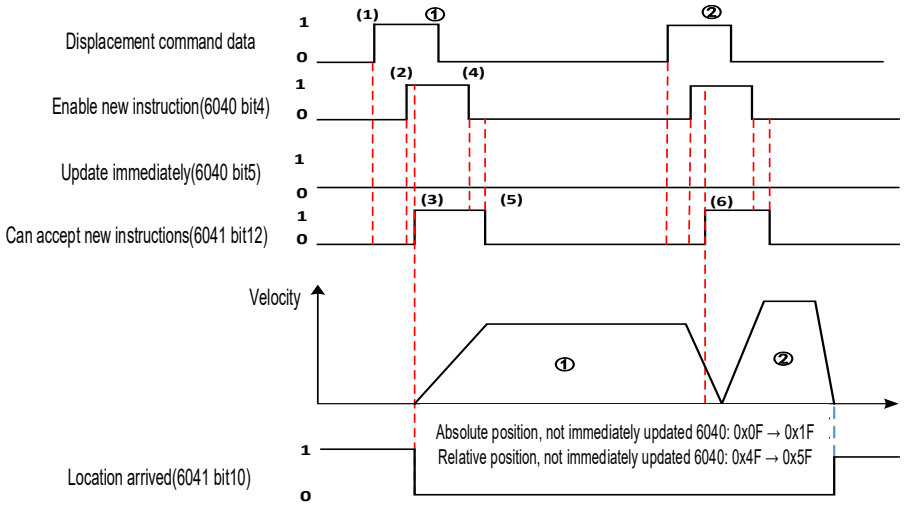


Figure 6-9

The notes (1) - (5) in the figure below are similar to the immediate-update mode. It can be seen from the comparison of position (6) in the figure, the difference is that in non immediate update mode, although the new target position ② has been accepted, it needs to reach position ① first, slow down and stop, and then move to new position ② . In non immediate-update mode, the motor has 5 buffers, i.e. it can have 5 target positions in the sequence at the same time and the motor will operate to these positions in turn. If the buffers is full, new position can't be accepted until the buffers is empty. These buffers are cleared when the motor is stopped.

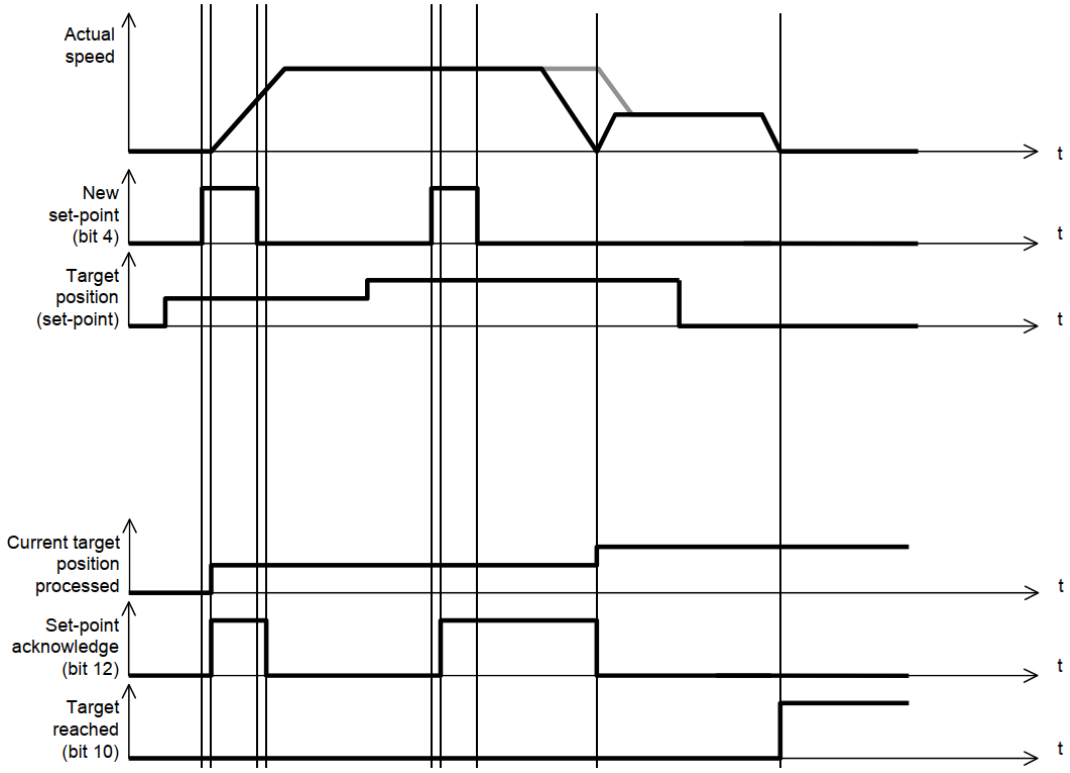


Figure 6-10

6.3.5 Impact Limited and Impact Unlimited Mode

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

- Impact Limited

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

- Impact Unlimited Mode

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

6.3.6 Configuration Examples

- Configuration mode

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

- Parameter configuration

  Write the target position 607Ah (user units).
  Write the current segment displacement command uniform running speed 6081h (user unit/s).
  Setting the acceleration 6083h (user units/s2) and deceleration 6084h (user units/s2) of the displacement.
  Send: 601 23 7A 60 00 10 27 00 00 (set the target position 607Ah to 10000)
  Send: 601 23 81 60 00 10 27 00 00 (set target speed 6081h to 10000)
  Send: 601 23 83 60 00 40 9C 00 00 (Set acceleration 6083h to 40,000)
  Send: 601 23 84 60 00 40 9C 00 00 (Set deceleration 6084_h to 40,000)

- Set the control word to enable the motor

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

- Trigger generator operation (after motor enable)

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

Table 6-8

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

Send: 601 2B 40 60 00 5F 00 00 00 (Set 6040h to 0x5F, relative position movement, not immediate update)
Send: 601 2B 40 60 00 1F 00 00 00 (Set 6040_h to 0x1F, absolute position movement, not updated immediately)
Send: 601 2B 40 60 00 6F 00 00 00 (set 6040h to 0x6F, relative position movement, immediate update)
Send: 601 2B 40 60 00 3F 00 00 00 (set 6040h to 0x3F, absolute position movement, update immediately)
The motor receives new position commands on the rising edge of bit 4 of control word 6040_h, so this bit needs to be cleared after each execution of a run.

- Monitoring parameters

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

6.4 Velocity Mode

6.4.1 Structural Chart

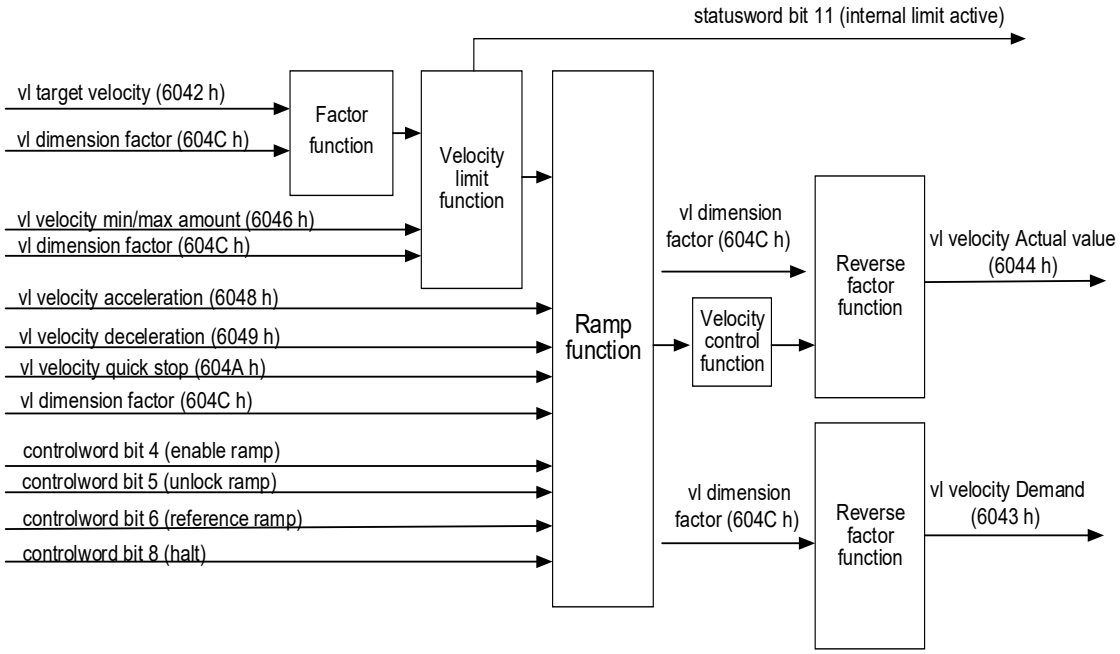


Figure 6-11

6.4.2 Objects Related

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

Table 6-9

Object Index	Description
6042_h (Vl target velocity)	Target velocity in VM mode,(default unit rpm, related to $604C_h$)
6043_h (Vl velocity demand)	Target velocity as the VM mode is activate,(default unit rpm, same as 6042_h)
6044_h (Vl velocity actual value)	Current feedback velocity actual value（rpm）
$604C_h$ (VL dimension factor)	Scaling factor for speed units in VM mode(default unit: rpm) Sub-index 1 contains the numerator (multiplier) used to calculate speed, and sub index 2 contains the denominator (divisor). Velocity unit: $\text{rpm} * \frac{\text{Sub - index 1}}{\text{Sub - index 2}}$ If any of the sub indexes 1 and 2 is 0, the speed unit is rpm.
6048_h (Velocity acceleration)	Vl velocity acceleration =Delta speed (6048_h:01_h) / Delta time (6048_h:02_h) Acceleration in VM mode. Sub-index 1 contains the speed change（unit: rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: Vl velocity acceleration =Delta speed (6048_h:01_h) / Delta time (6048_h:02_h)
6049_h (Velocity deceleration)	Deceleration in VM mode. Sub-index 1 contains the speed change（unit: rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: Vl velocity deceleration = Delta speed (6049_h:01_h) / Delta time (6049_h:02_h)
$604A_h$ (Quick stop deceleration)	Deceleration during Quick stop in VM mode. Sub-index 1 contains the speed change（unit: rpm) and sub-index 2 the corresponding time(unit: second). The acceleration is calculated as follows: Vl Quick stop deceleration = Delta speed ($604A_h$:01_h) / Delta time ($604A_h$:02_h)
6046_h (Velocity min max amount)	Speed limit in VM mode (default unit: rpm, related to $604C_h$). The details are as follows: 6046_h: 01_h set the minimum speed. If the target speed (6042_h) is lower than the minimum speed, its value will be set to the minimum speed of 6046_h: 01_h. 6046_h: 02_h sets the maximum speed. If the target speed (6042_h) is higher than the maximum speed, its value will be set to the maximum speed of 6046_h: 02_h.

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

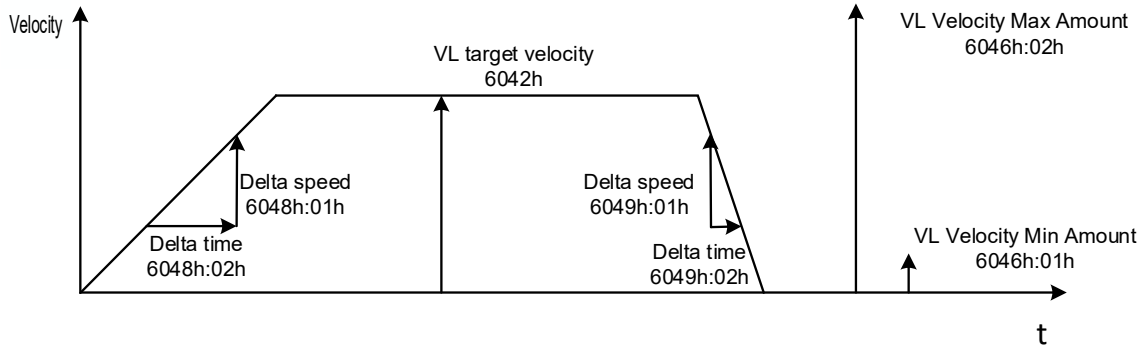


Figure 6-12

6.4.3 Control Instruction and Status Information

Enable

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

Control Word

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

MSB LSB

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

Table 6-10

Bit	Value	Description
2	0	When Quick stop is triggered, the motor will perform quick braking according to the Quick stop deceleration set in the object $604A_h$. The controller then switches to the "Switch on disabled" state
	1	No Operation
4	0	The output changes to the given immediately without acceleration and deceleration
	1	Adjust the velocity according to the acceleration and deceleration setting
5	0	No longer adjust velocity from the planner output, the velocity output shall be locked to the current value
	1	Adjust the speed according to the speed output of the planner
6	0	velocity curve planner input is 0
	1	velocity curve planner input is the given target velocity
8	0	No Operation
	1	Motor suspension

Status Word

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

MSB LSB

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

6.4.4 Function Description

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

6.4.5 Configuration Examples

- Configuration mode

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

- Parameter configuration

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

Send: 601 2B 48 60 02 01 00 00 00 (set 6048h:02h to 1)
Send: 601 23 49 60 01 F4 01 00 00 (Set 6049h:01h to 500)
Send: 601 2B 49 60 02 01 00 00 00 (Set 6049_h:02_h to 1)

- Write control word

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

- Set the target speed and run the motor.

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

- Parameter monitoring

  Current actual speed $606C_h$ (unit rpm).

6.5 Profile Velocity Mode (PV)

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

6.5.1 Structure Chart

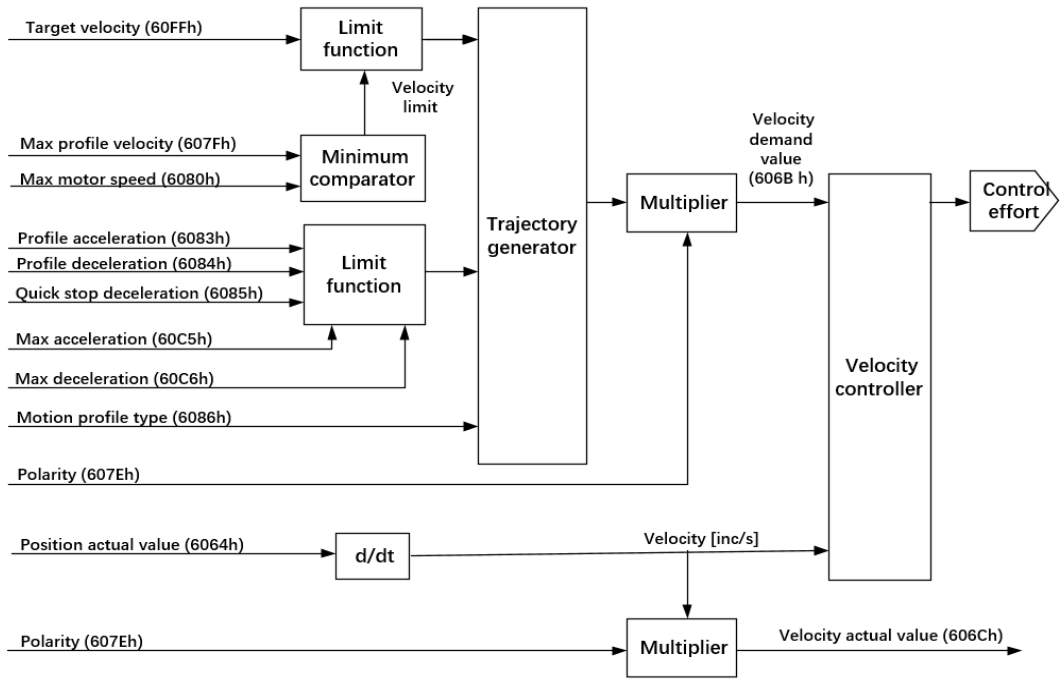


Figure 6-13

6.5.2 Objects Related

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

Table 6-11

Object index	Description
6063_h (Position Actual Enc Value)	Position Actual Enc Value(encoder unit)
6064_h（Position actual user value）	Position actual user value(user unit)
$606B_h$（Velocity demand value）	Velocity demand value
$606C_h$（Velocity actual value）	Current velocity actual value feedback(unit: rpm)

Object index	Description
$607E_h$（Polarity）	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
$607F_h$（Max profile velocity）	Maximum profile velocity during operation (user unit / s), which is used to limit the velocity
6080_h（Max motor speed）	Max motor speed(rpm)
6083_h（Profile acceleration）	The profile acceleration (user unit /s^2) during operation, which is limited by $60C5_h$.
6084_h（Profile deceleration）	The profile deceleration (user unit /s^2) during operation, which is limited by $60C6_h$.
6085_h（Quick stop deceleration）	Deceleration during "Quick stop" (user unit/s^2)
6086_h（Motion profile type）	Choose the type of curve, i.e. slope, If the value is "0", the impact (acceleration) will not be limited, i.e. ladder curve; If the value is "3", the value in $60A4_h$: 01_h-02_h will be used to limit the impact (acceleration), i.e. S-curve. This equipment only uses 01_h and 02_h indexes.
$60C5_h$（Max acceleration）	Max acceleration (user unit /s^2)
$60C6_h$（Max deceleration）	Max deceleration (user unit /s^2)
$60FF_h$（Target velocity）	Target velocity(user unit /s)

6.5.3 Control Instruction and Status Information

Enable

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

Control Word

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

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

Table 6-12

Bit	Value	Meaning
8	0	Motor keep operating
	1	Suspend

Status Word

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

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

Table 6-13

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

6.5.4 Function Description

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

6.5.5 Configuration Examples

- Configuration mode

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

- Configuration parameters.

  Write contour acceleration: 6083_h = 40000 (user units/s2).
  Write contour deceleration: 6084_h= 40000 (user units/s2).
  Send: 601 23 83 60 00 40 9C 00 00 (set acceleration 6083_h to 40,000)
  Send: 601 23 84 60 00 40 9C 00 00 (set deceleration 6084_h to 40000)

- Write control word.

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

- Set the target speed and run the motor.

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

- Parameter monitoring:

  Current actual speed $606C_h$ (unit rpm).

6.6 Homing Mode（HM）

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

6.6.1 Structure Chart

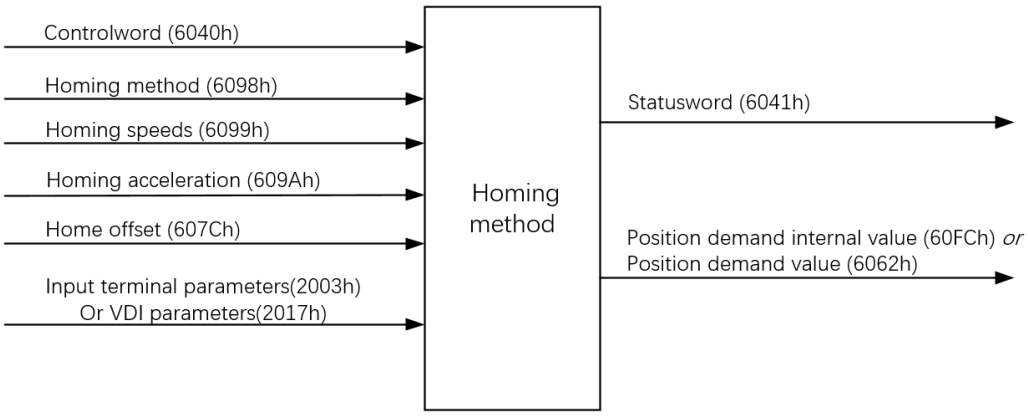


Figure 6-17

6.6.2 Objects Related

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

Table 6-17

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

Example of Velocity Meaning

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

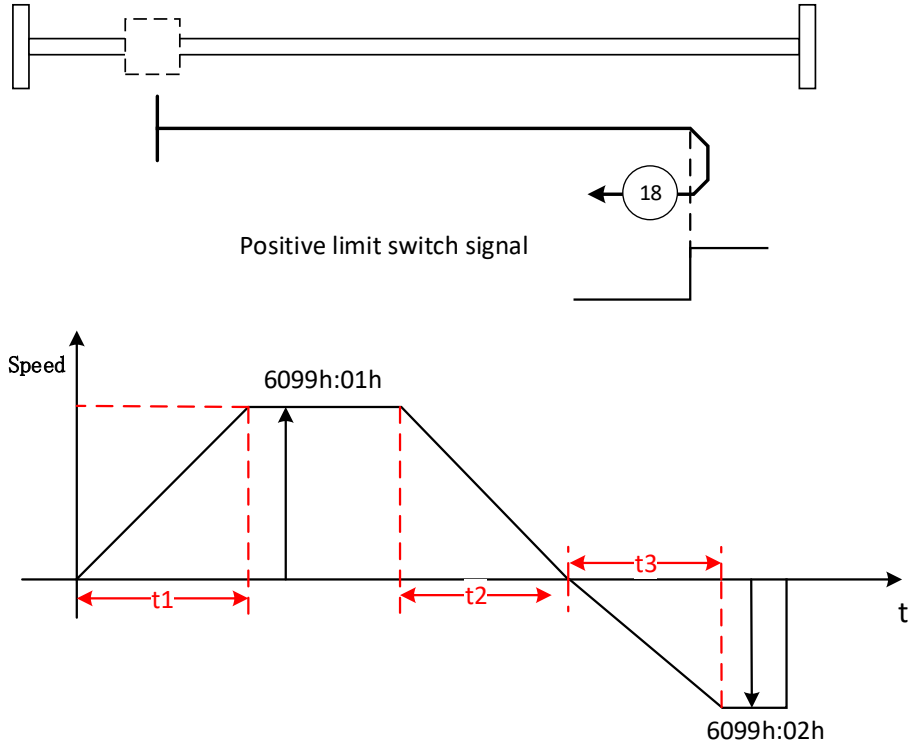


Figure 6-14

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

6.6.3 Control Instruction and Status Information

Enable

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

Control Word

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

MSB LSB

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

Table 6-18

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

Status Word

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

MSB LSB

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

Table 6-19

Bit	Value	Definition
10	0	Target position not reached
	1	Target position reached

12	0	Homing not completed
	1	Homing completed

6.6.4 Function Description

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

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

Table 6-20

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

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

The direction and switch status requiring homing are as follows:

Table 6-14

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

- Homing Method 17:

  Mechanical origin: negative limit switch

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

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

  The movement track is as follows:

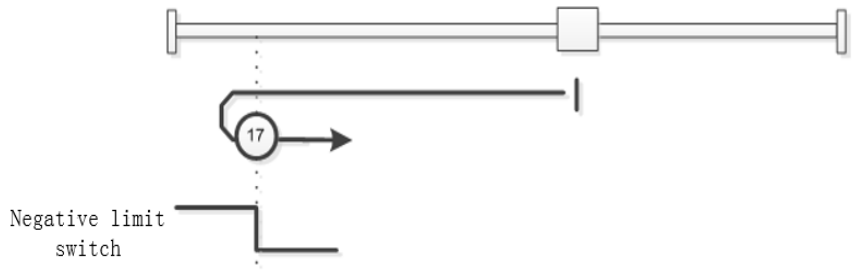


Figure 6-18

- Homing Method 18:

  Mechanical origin: positive limit switch

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

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

  The movement track is as follows:

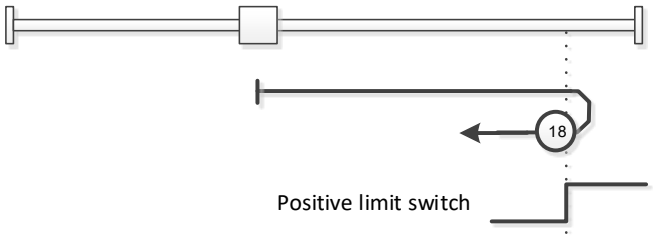


Figure 6-19

- Homing Method 19 and 20:

  Mechanical origin: origin switch

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

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

  The movement track is as follows:

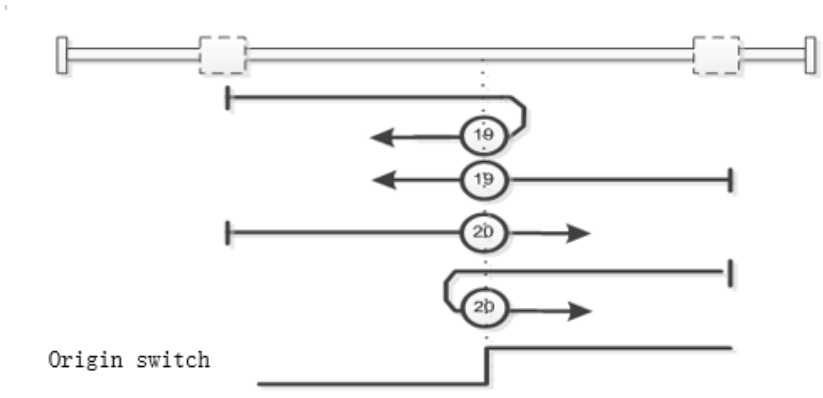


Figure 6-20

- Homing Method 21 and 22:

  Mechanical origin: origin switch

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

  pattern 22: When the origin switch is invalid, servo starts moving to home position at low speed. and halts until the origin switch is valid (edge-jump).When the origin switch is

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

The movement track is as follows:

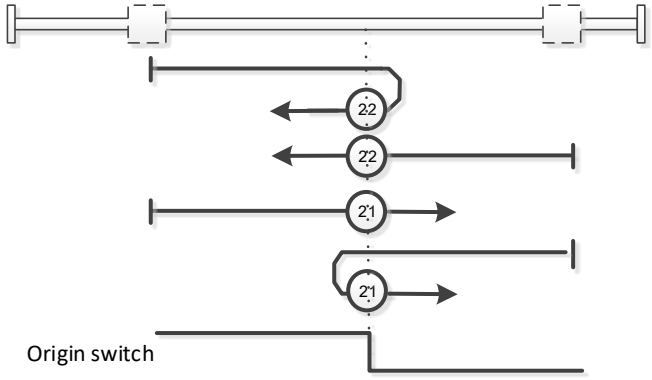


Figure 6-21

- Homing Method 23 ~ 26:

  Mechanical origin: origin switch

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

  Homing Method 23:

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

  Homing Method 24:

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

  Homing Method 25:

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

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

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

Homing Method 26:

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

The motion track is as follows:

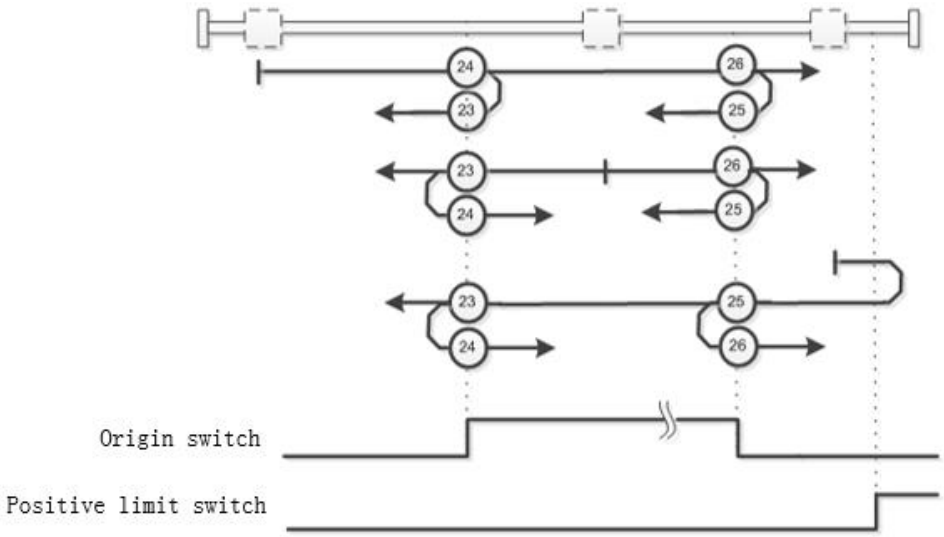


Figure 6-15

- Homing Method 27~30

Mechanical origin: origin switch

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

Homing Method 27:

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

Homing Method 28:

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

Homing Method 29:

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

Homing Method 30:

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

The motion track is as follows:

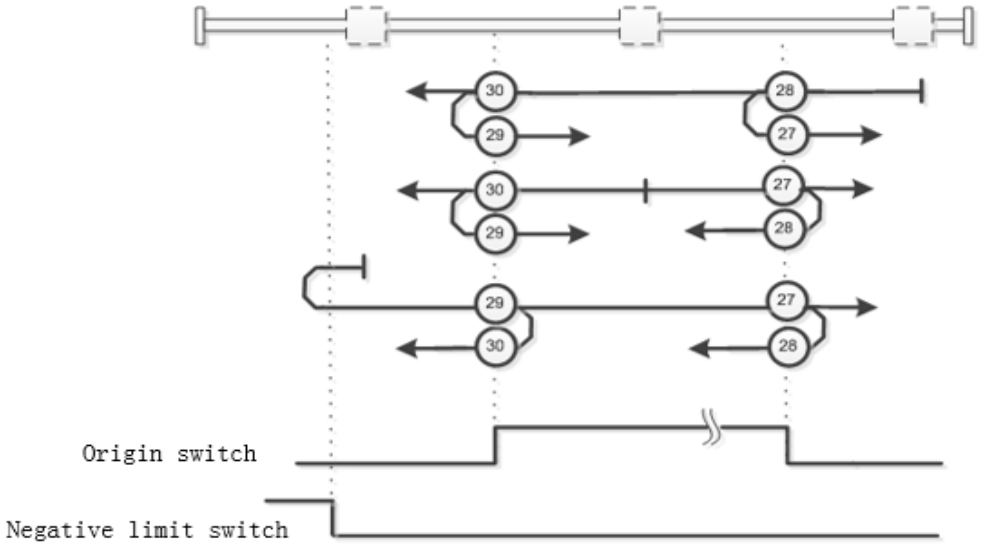


Figure 6-23

6.6.5 Configuration Example

- Setting mode

  Write 2002_h:01_h=0, run mode 6060_h=0x06 to operate in homing mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 04 00 00 00 (set to homing mode)
  Note: When switching modes, the motor should be in the deactivated state

- Setting terminal

  2003_h:01_h=15 (negative limit switch), 2003_h:02_h=0 (active low).
  Send: 601 2B 03 20 01 0F 00 00 00 (Set DI1 function as negative limit switch)
  Send: 601 2B 03 20 02 00 00 00 00 (set DI1 logic to active low)

- Write go home method

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

- Set find limit switch speed and find home signal speed

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

- Set go home acceleration

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

- Write control word

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

- Trigger generator operation (after motor enable)

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

6.7 Interpolated Position Mode（IP）

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

6.7.1 Structure Chart

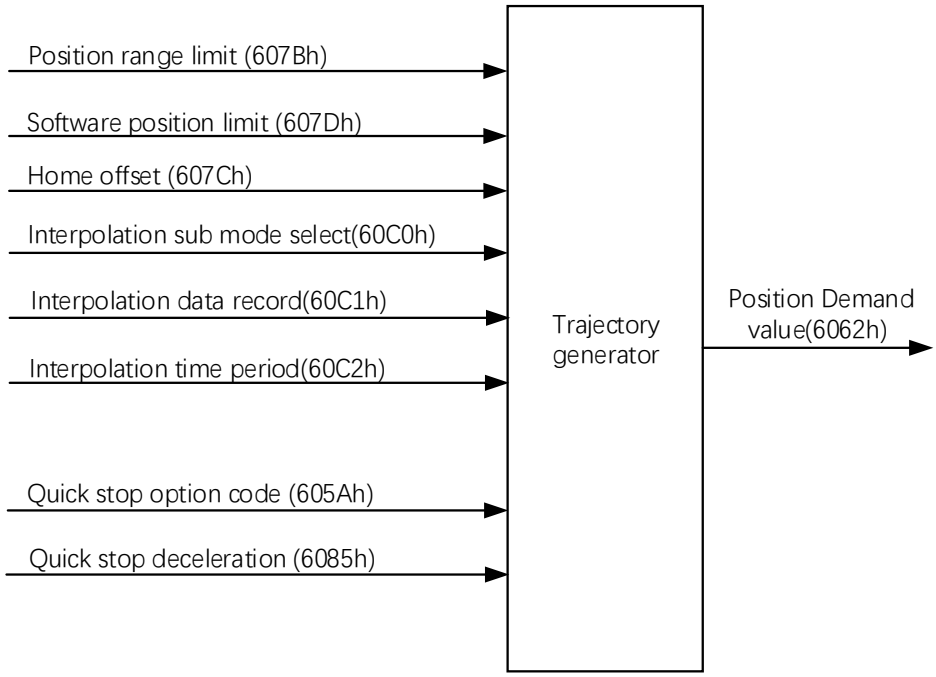


Figure 6-24

6.7.2 Objects Related

Table 6-22

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

6.7.3 Control Instruction and Status Information

Enable

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

Control Word

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

MSB LSB

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

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

MSB LSB

Table 6-23

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

Status Word

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

MSB LSB

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

Table 6-24

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

6.7.4 Function Description

Interpolation Diagram

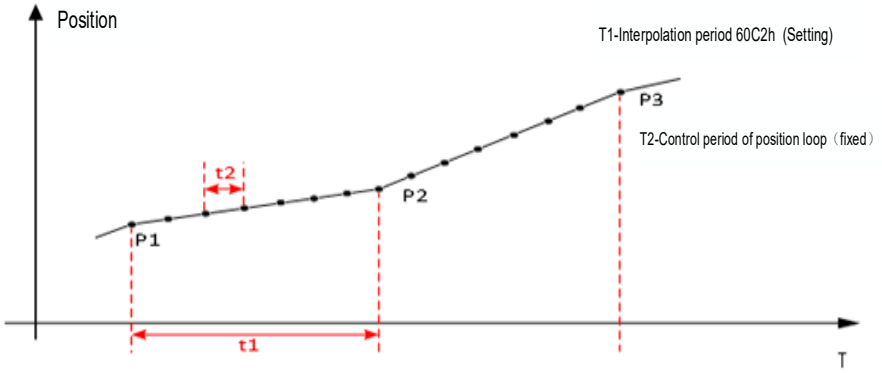


Figure 6-25

Notes:

t1: Interpolation period, which can be set in object dictionary $60C2_h$.
t2: Control period of position loop , determined by the servo drive internally.
P1 / P2 / P3: Absolute position. The absolute position instructions is sent by the object dictionary $60C1_h:01_h$. The interpolation mode only supports the absolute position instructions.

6.7.5 Configuration Examples

- Setting mode

  Write $2002_h:01_h$=0, run mode 6060h=0x07 to make it work in interpolation mode.
  Send: 601 2B 02 20 01 00 00 00 00 (set to CiA402 mode)
  Send: 601 2F 60 60 00 04 00 00 00 (set to interpolation mode)
  Note: When switching modes, the motor should be in the deactivated state

- Set interpolation time

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

- Write control word

  To enable the motor for the first time, you must first write 0x6, 0x7 and 0x1F to the control word 6040_h in sequence; to write 0x7 for subsequent disconnection; to write 0x1F for enabling.
  Send: 601 2B 40 60 00 06 00 00 00 (set 6040_h to 0x6 to make the motor ready)
  Send: 601 2B 40 60 00 07 00 00 00 (set 6040_h to 0x7 to disable motor)
  Send: 601 2B 40 60 00 01F 00 00 00 (Set 6040_h to 0x1F to enable motor)

- Trigger generator operation (after motor enable)

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

6.8 Cyclic Sync Position Mode（CSP）

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

6.8.1 Structure Chart

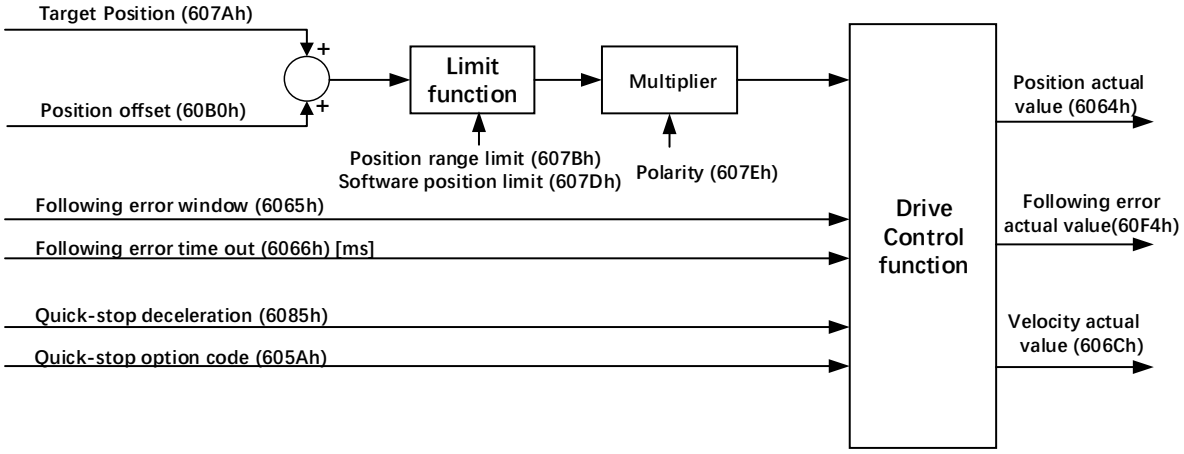


Figure 6-26

6.8.2 Objects Related

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

Table 6-25

Object Index	Description
$607A_h$ (Target position)	Target position(user unit)
$607B_h$ (Position Range Limit)	Position Range Limit(user unit)
$607D_h$ (Software Position Limit)	Target position limit (user unit)
$607E_h$ (Polarity)	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
6065_h (Following Error Window)	this object specifies the allowed tolerance from the rated position. If the actual position exceeds this tolerance for longer than the specified time (6066_h), tracking error will be reported, (user unit)
6066_h (Following Error Time Out):	Time range of following error , unit: millisecond. If the actual position exceeds the tolerance (6065_h) for longer than this time range, tracking error will be reported, Bit13 of the control word will be set to 1.
6085_h (Quick-Stop Deceleration)	Deceleration during "Quick stop" (user unit/s^2)
$605A_h$ (Quick-Stop Option Code)	Quick-Stop Option
$60B0_h$ (Position offset)	Position offset(user unit)
6064_h (Position Actual Value)	Current absolute position feedback from motor(user unit)
$606C_h$ (Velocity Actual Value)	Current velocity actual value feedback(unit: rpm)
$60F4_h$ (Following Error Actual Value)	Following Error Actual Value(user unit)

6.8.3 Control Instruction and Status Information

Enable

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

Control Word

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

Note The CSP mode only supports the absolute position instructions.

Status Word

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

MSB LSB

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

Table 6-26

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

6.8.4 Function Description

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

6.8.5 Configuration Example

- Configuration mode

  Set 2002_h:01_h=06060_h=0x02as velocity mode(VM)

- Set Control word

  6040_h = 0x060x070x7Fmotor operates;

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

6.9 Cyclic Sync Velocity Mode（CSV）

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

6.9.1 Structural Chart

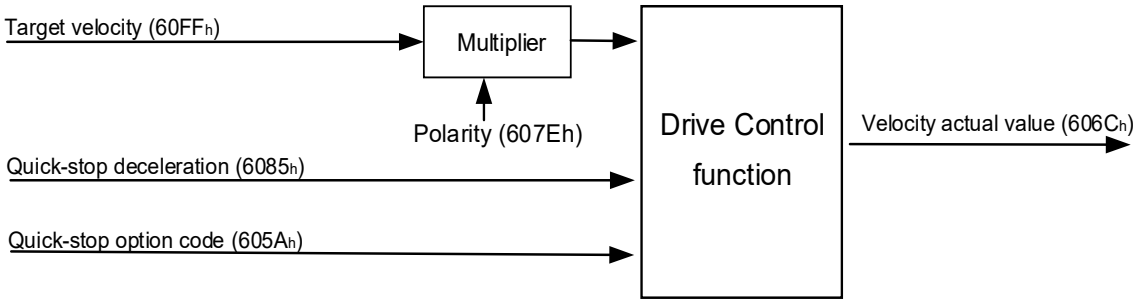


Figure 6-27

Object Related

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

Table 6-27

Object Index	Description
$605A_h$（Quick stop option code）	Quick stop option
$606C_h$（Velocity actual value）	Current velocity actual value feedback (unit: rpm)
$607E_h$（Polarity）	Rotation direction(polarity),See "5.1.4 $607E_h$: Polarity" for details
$60FF_h$（Target velocity）	Target velocity (user unit:/s)
6085_h（Quick-stop deceleration）	Deceleration during "quick stop" (user unit/s^2)

6.9.2 Control Instruction and Status Information

Enable

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

Control Word

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

Status Word

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

MSB LSB

In Cyclic Sync Velocity mode, the following bits in the object 6041_h (status word) have special function.

Table 6-28

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

6.9.3 Function Description

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

6.9.4 Configuration Examples

- Setting mode

  Write 2002_h:01_h=0, run mode 6060_h=0x09 to make it work in cyclic synchronous velocity mode.
  Note: When switching mode, the motor should be in the deactivated state

- Write control word

  6040_h=0x06→0x07→0x0F, the motor runs, the upper computer needs to write the speed object 60FFh in accordance with the synchronous cycle.

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

6.10 Nimotion Mode

6.10.1 Summary

The Nimotion mode is different from CIA402 mode, without state machine management and controlled by multi-functional terminal. Manufacturer modes include Manufacturer position mode, Manufacturer velocity mode.

6.11 Nimotion Position Mode

6.11.1 Structural Chart

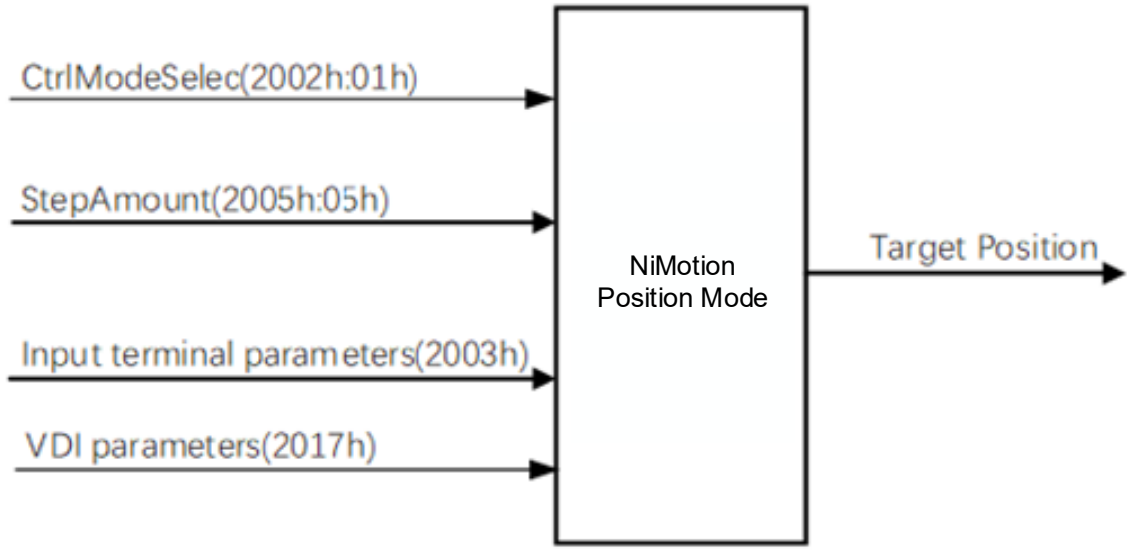


Figure 6-28

6.11.2 Objects Related

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

Table 6-29

Object Index	Description
2002_h:01_h (CtrlModeSelec)	Operation mode selection
2005_h:02_h (StepAmount)	Step amount of manufacturer position mode, , this parameter is the relative position amount (unit: closed loop: encoder unit inc, open loop: pulse number); Distinguish symbols, positive means forward turn, negative means reverse turn
2005_h:03_h (StepSpd)	Manufacturer position mode stepper motor speed, unit: rpm. The motor runs directly at this speed without acceleration or deceleration.
2017_h (VDI parameters)	Virtual terminal input function configuration (see "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Entity input terminal function configuration (see "7.1.1 Digital input" for details)

Note：1. Do not configure multiple input terminals for the same function at the same time. Otherwise, the sub-setting error will be reported.

2. The motor must be enabled before the step enable signal is effective.

6.11.3 Control Instruction and Status Information

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

Enable

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

① Set multi-function terminals (both physical terminals and virtual terminals are applicable. See 7.1 Digital input and output for specific configuration)

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

② Step some distance

Enable the motor by the terminal, and enable the step amounts through the terminal, then the motor will operates the distance (unit: Inc) preset in (2005_h: 05_h) for each stepping enabled; For each step enabled, the motor travels once.

6.11.4 Function Description

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

6.11.5 Configuration Examples

Parameter setting：

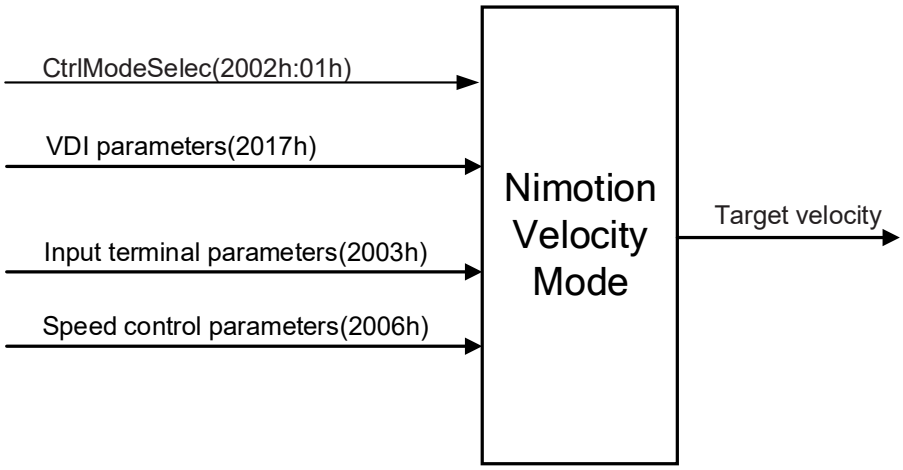
- Setting to Nimotion position mode: 2002_h:01_h=1.
- Setting the step amount: 2005_h:02_h = 50 (user units)
- Setting servo enable: 2003_h:03_h = 1, logic active low: 2003_h:04_h = 0.
- Set step amount enable: 2003_h:05_h = 20, logic is active on falling edge: 2003_h:06_h = 3.

Operation：

- Motor enable: DI1 input low.
- Step amount enable: DI2 input a falling edge, step a distance, each input a falling edge will step a distance.

6.12 Nimotion Velocity Mode

6.12.1 Structure Chart



6.12.2 Objects Related

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

Objects Index	Description
2002_h:01_h (CtrlModeSelec)	Operation mode selection
2006_h (Speed control parameters)	2006_h:01_h: Nimotion velocity mode main velocity reference source, 0-digital reference, 3-duty cycle reference (20Hz-20kHz, 1kHz recommended) 2006h:02h : Speed instruction digital given value，unit: rpm 2006h:03h : Velocity command acceleration ramp time constant, unit ms, representing the time it takes to accelerate from 0rpm to 1000rpm 2006h:04h : Velocity command deceleration ramp time constant, unit ms.
2017_h (VDI parameters)	Virtual terminal input function configuration (see "7.1.1 Digital input" for details)
2003_h (Input terminal parameters)	Function configuration of entity input terminals (see "7.1.1 Digital input")

6.12.3 Control Instruction and Status Information

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

Enable

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

① Selection of velocity source：

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

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

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

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

③Velocity, acceleration and maximum velocity setting

If the main velocity command source is selected as 0-digital reference, the velocity reference value needs to be set and written to the object 2006$_h$:02$_h$;

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

④Run

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

6.12.4 Function Description

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

6.12.5 Configuration Examples

Implementation of digital feeder speed regulation.：

① Parameter Setting：

- Setting to NiMotion velocity mode: 2002$_h$:01$_h$=2.
- Setting the main speed command source to 0 (digital give): 2006$_h$:01$_h$=0
- Setting the acceleration ramp time constant to 100ms: 2006$_h$:03$_h$=100 (unit ms).
  Setting the deceleration ramp time constant to 100ms: 2006$_h$:04$_h$=100 (unit ms).
- Setting the digital give to 60rpm: 2006$_h$:02$_h$=60(unit rpm).

- Set DI1 terminal.
  Select function as servo enable: 2003$_h$:03$_h$ = 1.
  Logic selection active low: 2003$_h$:04$_h$ = 0.

② Operation：

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

To achieve duty cycle input velocity regulation：

① Parameter setting.

- Setting the velocity mode to 2 (NiMotion speed): 2002$_h$:01$_h$ = 2.
- setting the main speed command source to 3 (duty cycle): 2006$_h$:01$_h$=3
- Setting the acceleration ramp time constant to 100ms: 2006$_h$:03$_h$=100 (unit ms).
  Setting the deceleration ramp time constant to 100ms: 2006$_h$:04$_h$=100 (unit ms).
- Setting the maximum speed to 3000rpm: 2000$_h$:0F$_h$=3000 (unit rpm).
  DI1 duty cycle input: 2003$_h$:03$_h$=44.
  DI2 duty cycle direction input: 2003$_h$:05$_h$=45.
  DI3 function number is set to servo enable.

Function selection for servo enable 2003_h:07_h = 1.
Logic select low active 2003_h:08_h = 0.

②Operation：

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

7 Special Function

7.1 Digital Inputs and Output

The integrated low-voltage stepper motor has input and output interfaces. The integrated digital inputs is used to connect auxiliary sensors of motion control, such as fretting limit switch, photoelectric switch, Hall sensor, etc.

Note：The terminals of different series of motors are different, see the operation manual of the corresponding motor for details.

7.1.1 Digital Input

Table 7-1

Object Index	Description
2003_h（Input terminal parameters）	Function configuration of entity input terminals
2017_h（VDI parameters）	Virtual terminal input function configuration
2021_h（DX parameters）	DX terminal function configuration
$200B_h:03_h$（MonitoredDiStates）	Actual state of voltage level corresponding to entity terminalsbit0-DI1~bit3-DI4

Note:

1、The actual number input in the corresponding motor manual shall prevail. The maximum number of digital entity input terminals is 5：Fixed input DI1 ~ 3；when DX1 and DX2 is configured as input. When $2021_h:01_h$ is 0，DX1 is configured as input；When $2021_h:02_h$ is 0，DX2 is configured as input。

2、Object 200Bh:03 Displays the actual input level status corresponding to the physical terminal.

3、The communication virtual input terminal is set in parameter group 2017_h.

(1) Physical terminal

Special function selection

The specific values are taken from the following table.

Table 7-1

Function Number	Function Definition	Optional terminal logic
0	No definition	-
1	Servo Enable	0、1
2	Alarm Reset	2、3、4
12	Pause（In Nimotion Mode）	0、1
14	Forward overtravel switch	0、1
15	Reverse overtravel switch	0、1
20	Step amount enable	2、3、4
31	Home switch	0、1
33	Set home position	2、3、4
38	Clear fault history	2、3、4
44	Duty cycle input (fixed DI1)	-
45	Duty cycle input direction (fixed DI2)	-

Note:

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

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

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

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

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

6 Object 200Bh:03h displays the actual input level status corresponding to the physical terminal：

$200B_h:03_h$

Bit	15~5	4	3	2	1	0
Description	-	DI5	DI4	DI3	DI2	DI1

Terminal Logic Selection

The multi-function terminal logic selection is set by the 2003h (Physical Input terminal) parameter group. The following table describes the values：

Table 7-3

Value	Logical definition
0	Active low
1	Active High
2	Effective rising edge
3	Effective falling edge
4	Both rising and falling edges are effective

(2) Virtual Terminal

Virtual VDI terminal special function definition

The specific values are taken from the following table.

Table 7-4

Function Number	Function Definition	Optional terminal logic
0	No definition	-
1	Servo Enable	0
2	Alarm reset	1
12	Pause（In Nimotion Mode）	0
20	Step amount enable	1
33	Set zero point	1
38	Clear fault history	1

Logic selection of virtual VDI terminal

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

Table 7-2

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

Operation:

Enable：The virtual terminal of the integrated stepper motor is valid when index $2017_h:01_h$ is 1, and the value of $2017_h:01_h$ is 0.

Configuration：Configured by the 2017_h subindex input terminal function number and logical selection.

Index $2017_h:02_h$ indicates the state corresponding to the virtual input terminal (bit0(VDI1)~ bit3(VDI4)), which is given by the communication mode and defined as follows:

$2017_h:02_h$

Bit	15~4	3	2	1	0
Description	-	VDI4	VDI3	VDI2	VDI1

Example:

Setting input terminal DI1 to positive limit switch is as follow

- Set DI1 as positive limit switch, 2003_h: 03_h = 14

- Logic selection is low-level-valid, 2003_h: 04_h = 0

7.1.2 Digital Output

The digital output of this device has two DX1 and DX2 that can be configured as digital output, and the parameter 2021_h needs to be configured:

If parameter 2021_h:01_h is set to 1, the DX1 port is configured as output. If parameter 2021_h:02_h is set to 1, the DX2 port is set to output.

Related Object Settings

Table 7-6

Object Index	Description
2004_h（Output terminal parameters）	Output terminal parameters
2021_h（DX parameters）	DX terminal configuration parameters can be configured Subindex: 2021_h:01_h (DX1InOutSelec) : DX1 function selection. If this parameter is set to 1, this parameter is set to the output terminal 2021_h:02_h (DX2InOutSelec) : specifies the DX2 function. If this parameter is set to 1, this parameter is used as the output terminal

Output Terminal Function Selection：

Table 7-7

Function number	Function definition
0	Configure DX as output
1	Alarm Output
2	Goal reached
3	Unenabled /enable status output（0-unenable；1-enable）
4	Whether the speed is zero（0-speed is 0；1-speed is not 0）
5	Locked-rotor output（in close-loop）（0-no locked；1-locked）

Note：Function When "0" (user control) is selected: The level of the output terminal is set by the corresponding bit of object 2004_h:01_h.

2004_h:01_h

Bit	15~2	1	0
Description	-	DO2	DO1

Output Terminal Logic Selection：

Table 7-8

Set value	Logical definition
0	Active low
1	Active high

Example

Example 1：Configure DX1 as user control and output high level:

- Configure DX1 as output，**2021_h:01_h** =1，
- Configure DX1 logic to be active at high level，**2004_h:02_h** =0

2004_h:03_h =1

- The communication setting DX1 output level,**2004_h:01_h** = 0x01， DX1 output high level.

Example 2：Configure DX2 as unenabled /enable status output：

- Configure DX2 as output，**2021_h:02_h** =1，
- Configure DX2 as Unenabled /enable status output, **2004_h:04_h** =3
  Configure DX2 logic to be active at high level：**2004_h:05_h** =1
- When motor is enabled，DX2 output high level；otherwise，DX2 output low level.

7.2 Parameter Saving and Restore

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

7.2.1 Parameter Saving

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

0x65766173 is written in 1010h:01h to save all parameters.

Example：

CAN-ID	Message	Note
0x600 + ID	23 10 10 01 73 61 76 65	Save all parameters

7.2.2 Parameter Restore

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

0x64616F6C is written in 1011_h:01_h to restore all parameters.

Example：

CAN-ID	Message	Note
0x600 + ID	23 11 10 01 6C 6F 61 64	Restore all parameters

7.3 Setting the Origin

Set the origin through the terminal：

Set the physical or virtual terminal function to set Origin (for details, see 7.1.1) so that the terminal can effectively set the origin.

The default function of the virtual terminal VDI1 is Set Origin (Valid rising edge) : The bit1 bit of object 2017h:02h changes from 0 to 1.

Example：

CAN-ID	Message	Note
0x600 + ID	2B 17 20 02 00 00 00 00	The bit1 of 2017_h:02_h is 0
0x600 + ID	2B 17 20 02 01 00 00 00	The bit1 of 2017_h:02_h is 1

7.4 Total Power-on Time

7.4.1 Reading

The total power-on time of the motor can be read by object 200Bh:06h, unit: 15min (min)

Example：

CAN-ID	Message	Note
0x600 + ID	40 0B 20 06 00 00 00 00	The total power-on time was read

7.5 Speed Filtering time

The filtering time of the speed display part can be set, and the parameter is in the object 2008_h:04_h. The larger the time, the more obvious the filtering effect.

7.6 Version

Hardware Version：Hardware Version is stored in 2002_h:05_h.
Software Version：Software Version is stored in 1018_h:03_h.

Example：

CAN-ID	Message	Note
0x600 + ID	40 02 20 05 00 00 00 00	Read the hardware version
0x600 + ID	40 18 10 03 00 00 00 00	Read the software version

8 Adjust

8.1 Summary

With the increase of the gain, the response speed of the system is accelerated, the adjustment accuracy of the system is improved, but the system is easy to overshoot, the stability of the system becomes worse, and even lead to the instability of the system. The gain value is too small, the adjustment accuracy is reduced, the response speed is slow, the adjustment time is longer, and the dynamic and static performance of the system is deteriorated.

Related objects：

Object Index	Description
2008_h:01_h (PosLoopGain)	Gain of position-loop, increasing it can accelerate positioning and improve the anti-disturbance capacity during the motor operating stably . But setting too large can cause system instability and oscillation.
2008_h:05_h (LockedSpdErrScale)	To determine the ratio of velocity error to gridlock, in units of 1% (as a percentage of target velocity)。
2008_h:06_h (LockedSpdThreshold)	Determine the speed threshold (unit: rpm) for blocking rotation. When the actual speed is lower than this threshold, and the speed error ratio exceeds 2008h:05h, and the duration exceeds the time set in 2008h:07h, it is determined that the motor is blocked.
2008_h:07_h (LockedTime)	Determine the time threshold for rotating blockage (unit: ms).
2008_h:09_h (SpdFdFwrGain)	Speed feedforward gain, speed feedforward can improve speed command response.
2008_h:$0A_h$ (SpdFdFwrFilterCoeff)	Velocity feedforward filter coefficient
2008_h:$0B_h$ (SpdFilterCoeff)	Velocity filter coefficient, range: 0-1000; The smaller the filter coefficient is, the larger and more stable the change delay of the speed instruction is, and the appropriate value should be set according to the actual situation.
2008_h:$0C_h$ (SpdLimitThreshold)	Speed change limiting threshold, unit: rpm/ms; The limiting threshold of acceleration to prevent the motor from stepping out of step needs to be adjusted according to the actual situation.
2008_h:$0D_h$ (CurrentFdFwrGain)	Current feedforward gain
2008_h:$0E_h$ (CurrentFilterCoeff)	Current filtering coefficient

8.2 Position Gain parameters

Object 2008_h:01_h is a position loop gain parameter. The gain parameters of the position loop are related to the motor response speed. In the cyclic synchronous position mode and the interpolation mode, the gain parameters are related to the synchronization period. As the period becomes larger, the gain parameters need to be appropriately smaller.

When the gain is too large, the motor will overshoot and jitter near the target position.

When the motor is loaded, the response speed of the motor is slow, and the gain needs to be reduced appropriately.

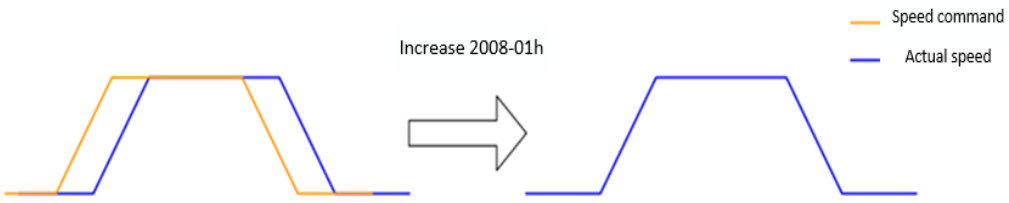


Figure 8-1

8.3 Parameters for Determining Block

The motor is blocked during operation: If the speed error continues to exceed the percentage

ratio (relative to the internal real-time target speed) set by object 2008h:05h, and the actual measured running speed is smaller than the speed threshold set by object 2008h:06h, and the duration exceeds the time threshold set by object 2008h:07h, it is determined that a gridlock occurs.

The three objects need to be adjusted to appropriate values based on the actual use of the system to better adapt to the actual system.

8.4 Feed Forward Parameter Adjustment

Speed Feedforward

Speed feedforward can improve the speed command response, the basic adjustment methods are as follows:

The speed feedforward filter coefficient 2008_h:$0A_h$ is set as a fixed value; Then, the speed feedforward gain 2008_h:09_h set value is gradually increased from 0 until a certain set value, the speed feedforward effect. When adjusting, 2008_h:$0A_h$ and 2008_h: 09_h should be adjusted repeatedly to find a good balance setting.

2008_h:09_h When set to 0, turn off speed feedforward.

Current Feedforward

The current feedforward filter coefficient is 2008_h:$0E_h$ and the current feedforward gain object is 2008_h:$0D_h$. When adjusting, you should repeatedly adjust 2008_h:$0E_h$ and 2008_h: $0D_h$ to find a good balanced setting.

2008_h:$0D_h$ When set to 0, turn off current feedforward.

8.5 Acceleration Limiting Parameter

The purpose of acceleration limiting is to prevent the given acceleration of the motor from exceeding the motor response speed, and the situation of out-of-step occurs.

The limiting parameter is 2008_h:$0C_h$, expressed in rpm/ms. The value should be adjusted according to the actual use situation to better adapt to the actual system.

9 Error Management

9.1 Alarm Codes

When the integrated servo has a warning or fault, the motor will actively alarm, and execute corresponding actions. The corresponding alarm code will be stored in object 1003_h: 02_h ~ 1003_h: 10_h, and the current fault number be stored in object 1003_h,the total fault numbers are stored in 1003_h:01_h.

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

The definition of the corresponding bits of the alarm code:

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

Note：

- Failure response code：

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

- Self-reset allowed：

00_h：No
01_h：Yes

Default alarm code definition

Alarm codes are adjustable parameters (individual cannot be adjusted), and users can configure them according to their actual needs (some can). Object 200Eh defines corresponding alarm codes. The specific contents are as follows：

Response Code	Self-reset Allowed	Error Code	Note
00_h	00_h	2300_h	Overcurrent（cannot be adjusted）
00_h	01_h	3110_h	Overvoltage
00_h	01_h	3120_h	Undervoltage
00_h	00_h	4310_h	High temperature alarm（cannot be adjusted）
00_h	01_h	8120_h	CAN passive error
00_h	01_h	8130_h	CAN mailbox overflow
00_h	01_h	8140_h	CAN bus disengagement
00_h	01_h	5530_h	EEPROM Fault
03_h	01_h	7121_h	Motor stalled（In close-loop）

00_h	01_h	5540_h	FLASH Error
03_h	01_h	6320_h	Parameter setting error
00_h	00_h	$FF00_h$	Overheat shutdown（cannot be adjusted）
00_h	00_h	$FF01_h$	Error command（cannot be adjusted）
00_h	00_h	$FF02_h$	Commands that cannot be executed（cannot be adjusted）
00_h	01_h	$FF03_h$	SPI communication failed（cannot be adjusted）
03_h	00_h	$FF04_h$	EEPROM self-test error（cannot be adjusted）
00_h	01_h	$FF07_h$	Self-test failure（cannot be adjusted）
00_h	01_h	$FF06_h$	Encoder self-test failure（In close-loop）
00_h	01_h	$FF08_h$	Encoder frame error（In close-loop）
00_h	01_h	$FF09_h$	Encoder check error（In close-loop）
00_h	01_h	$FF0A_h$	Encoder instruction error（In close-loop）
00_h	01_h	$FF0C_h$	Encoder field too high（In close-loop）
00_h	01_h	$FF0D_h$	Encoder field too low（In close-loop）
03_h	01_h	$FF0E_h$	Negative limit switch triggers alarm（No HM）
03_h	01_h	$FF0F_h$	Positive limit switch triggers alarm（No HM）
04_h	01_h	$FF10_h$	Software location limit（In PP、IP）
01_h	00_h	$FF11_h$	Wrong key

9.2 Fault Reset

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

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

9.3 Fault Detection Instructions

- Overcurrent detection

When the drive current is greater than the overload current set in 2001_h:07_h, the overcurrent fault occurs, and the drive axle is automatically cut off. Query the 1003_h error field.

- Over voltage and Under voltage error detection

When the voltage of servo power-supply is higher than the set voltage (2001_h:01_h), Over voltage error is triggered; when the voltage of servo power-supply is lower than the set voltage(2001_h:02_h),Under voltage error is triggered.

- Over temperature detection

Overheating occurs when the drive temperature is too high. By default, the fault response is unrecoverable and enters the fault state.

- CAN Communication error detection

When the controller CAN communication fails, the corresponding fault is reported.

Solution: Check whether the cable is correctly and firmly inserted; The wiring sequence is correct.

- EEPROM fault detection

An error is reported when EEPROM read/write faults occur.

- Motor stalled

The motor is blocked during operation: the speed error continues to exceed the ratio set by object 2008_h:05_h, and the actual measured speed is less than the speed set by object 2008_h: 06_h, and the duration exceeds the time set by 2008_h: 07_h.

In this case, the possible cause is that the running speed set under the load is too high, or the running current set is too small and the load is too large.

The solution:

Reduce operating speed; Increase the running current to the appropriate value; Adjust the position loop ratio P (2008_h:01_h) to the appropriate value; Adjust the lockup speed threshold (2008_h:06_h) or lockup time threshold (2008_h:07_h).

- FLASH fault detection

An error is reported when Flash write continuation fails.

- Encoder fault detection

The corresponding error is reported in case of abnormal magnetic encoder.

- Driver fault detection

This fault is reported when the driver chip communication error occurs.

- Terminal fault detection

Parameter Setting Fault detection. The fault is reported if the function of the multi-function terminal is not set in manufacturer mode, the same function number is set for different terminals (including virtual terminals), or the function number 14, 15, and 31 is not set in CIA402 origin regression mode.

Solution: Check the terminal configuration.

- Positive and negative limit switch fault detection

Without the origin regression mode of CIA402, if the terminal is configured with a positive or negative limit switch, the corresponding limit fault will occur when the positive or negative limit switch is encountered.。

- Software limit detection

The software limit is set in the contour position mode, interpolation mode and cyclic synchronization mode, and the position range value is set in the object $607B_h$ and $607D_h$ (the larger of the two values is the effective value). In the default fault response code configuration: When this position is reached, the motor stops and cannot continue to run in this direction, but it can run in reverse (in this case, the reverse target position only supports the absolute position).

10 Object Dictionary

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

Noun interpretation

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

Table 10-1

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

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

Table 10-2

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

Mappability: please refer to the following table for details

Table 10-3

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

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

Table 10-4

activate method	Description
Immediate activate	After parameter setting, the set value takes effect immediately

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

10.1 Communication Parameter Group 1000h Description

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
1000h	-	Device type	Device type	uint32	-	RO	NO	-	VAR	-
1001h	-	Error Register	Error Register	Uint8	-	RO	NO	0	VAR	-
1003h	-	Pre-defined error field	Pre-defined error field	-	-	RO	NO	-	ARRAY	-
	00h	NumberErrors	Number of alarms in the current device	uint8	-	RW	NO	-	VAR	-
	01h	ErrorField_1	Historical alarm cache	uint32	-	RO	NO	-	VAR	-
	02h	ErrorField_2		uint32	-	RO	NO	-	VAR	-
	03h	ErrorField_3		uint32	-	RO	NO	-	VAR	-
	04h	ErrorField_4		uint32	-	RO	NO	-	VAR	-
	05h	ErrorField_5		uint32	-	RO	NO	-	VAR	-
	06h	ErrorField_6		uint32	-	RO	NO	-	VAR	-
	07h	ErrorField_7		uint32	-	RO	NO	-	VAR	-
	08h	ErrorField_8		uint32	-	RO	NO	-	VAR	-
	09h	ErrorField_9		uint32	-	RO	NO	-	VAR	-
	0Ah	ErrorField_A		uint32	-	RO	NO	-	VAR	-
	0Bh	ErrorField_B		uint32	-	RO	NO	-	VAR	-
	0Ch	ErrorField_C		uint32	-	RO	NO	-	VAR	-
	0Dh	ErrorField_D		uint32	-	RO	NO	-	VAR	-
	0Eh	ErrorField_E		uint32	-	RO	NO	-	VAR	-
	0Fh	ErrorField_F		uint32	-	RO	NO	-	VAR	-
	10h	ErrorField_10		uint32	-	RO	NO	-	VAR	-
1008h	-	Manufacturer Device Name	Device name	String	-	RO	NO	-	-	-
1009h	-	Manufacturer Hardware Version	Hardware version	String	-	RO	NO	-	-	-
100Ah	-	Manufacturer Software Version	Software version	String	-	RO	NO	-	-	-
1010h	-	Store parameter	Save parameters	-	-	RO	NO	-	ARRAY	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
	00_h	Number of Entries	-	uint8	-	RO	No	4	VAR	-
	01_h	Save All Param	-	uint32	-	RW	NO	1	VAR	-
	04_h	Save factory all parameters	-	uint32	-	RW	NO	1	VAR	-
1011_h	-	Restore default parameter	-	-	-	RO	NO	-	ARRAY	-
	00_h	Number of Entries	-	uint8	-	RO	No	4	VAR	-
	01_h	Restore all default parameter	-	uint32	-	RW	NO	1	VAR -	-
	04_h	Restore factory all parameters	-	uint32	-	RW	NO	1	VAR	-
1017_h	-	Producer heartbeat time	-	uint16	-	RW	NO	0	ms	-
1018_h	-	Identity Object	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	R0	NO	-	VAR	-
	01_h	Vendor-ID	-	uint32	-	RO	NO	0x2D	-	-
	02_h	Product code	-	uint32	-	RO	NO	-	-	-
	03_h	APP Version	-	uint32	-	RO	NO	-	-	-
	04_h	Serial number	-	uint32	-	RO	NO	-	-	-
1400_h	-	Receive PDO 1 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	-	uint32	-	RW	NO	0x200+Node_ID	VAR	-
	02_h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
1401_h	-	Receive PDO 2 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	-	uint32	-	RW	NO	0x300+Node_ID	VAR	-
	02_h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
1402_h	-	Receive PDO 3 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	R0	NO	-	VAR	-
	01_h	COB-ID	-	uint32	-	RW	NO	0x400+Node_ID	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
1403h	-	Receive PDO 4 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	R0	NO	-	VAR	-
	01h	COB-ID	-	uint32	-	RW	NO	0x500+Node_ID	VAR	-
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
1600h	-	Drive Control	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	-	uint32	-	RW	NO	0x60400010	VAR	-
	02h	2st object to be mapped	-	uint32	-	RW	NO	0x60600008	VAR	-
1601h	-	Positioning Control	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	-	uint32	-	RW	NO	0x607A0020	VAR	-
	02h	2st object to be mapped	-	uint32	-	RW	NO	0x60C10120	VAR	-
1602h	-	Velocity Control	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	-	uint32	-	RW	NO	0x60420010	VAR	-
	02h	2st object to be mapped	-	uint32	-	RW	NO	0x60FF0020	VAR	-
1603h	-	Output Control	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01h	1st object to be mapped	-	uint32	-	RW	NO	0x20040110	VAR	-
1800h	-	Transmit PDO 1 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RO	NO	-	VAR	-
	01h	COB-ID	-	uint32	-	RW	NO	0x180+Node_ID	VAR	-
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
	03h	Inhibit Time	-	uint16	-	RW	NO	0x64	VAR	-
	05h	Event Timer	-	uint16	-	RW	NO	0x0	VAR	-
1801h	-	Transmit PDO 2 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RO	NO	-	VAR	-
	01h	COB-ID	-	uint32	-	RW	NO	0x280+Node_ID	VAR	-
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
	03h	Inhibit Time	-	uint16	-	RW	NO	0x64	VAR	-
	05h	Event Timer	-	uint16	-	RW	NO	0x0	VAR	-
1802h	-	Transmit PDO 3 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries		uint8	-	RO	NO	-	VAR	-
	01h	COB-ID	-	uint32	-	RW	NO	0x380+Node_ID	VAR	-
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
	03h	Inhibit Time	-	uint16	-	RW	NO	0x64	VAR	-
	05h	Event Timer	-	uint16	-	RW	NO	0x0	VAR	-
1803h	-	Transmit PDO 4 Communication Parameter	-	-	-	RO	NO	-	REC	-
	00h	Number of Entries	-	uint8	-	RO	NO	-	VAR	-
	01h	COB-ID	-	uint32	-	RW	NO	0x480+Node_ID	VAR	-
	02h	Transmission Type	-	uint8	-	RW	NO	0xFF	VAR	-
	03h	Inhibit Time	-	uint16	-	RW	NO	0x64	VAR	-

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
	05_h	Event Timer	-	uint16	-	RW	NO	0x0	VAR	-
$1A00_h$	-	Drive Status	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	-	uint32	-	RW	NO	0x60410010	VAR	-
	02_h	2st object to be mapped	-	uint32	-	RW	NO	0x60610008	VAR	-
$1A01_h$	-	Positioning Status	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	-	uint32	-	RW	NO	0x60630020	VAR	-
	02_h	2st object to be mapped	-	uint32	-	RW	NO	0x60640020	VAR	-
$1A02_h$	-	Velocity Status	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	-	uint32	-	RW	NO	0x606C0020	VAR	-
$1A03_h$	-	Input Status	-	-	-	RO	NO	-	REC	-
	00_h	Number of Entries	-	uint8	-	RW	NO	-	VAR	-
	01_h	1st object to be mapped	-	uint32	-	RW	NO	0x200B0310	VAR	-

10.2 Manufacturer Defined Parameter Group 2000h Description

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
2000h	01h	RatedVoltage	-	uint16	0-65535	RO	NO	24	1V	Shutdown effect
	02h	MaxMotorSpeed	-	uint16	0-65535	RO	NO	2000	1rpm	Shutdown effect
2001h	01h	BusOverVoltVal	-	uint16	10	RW	NO	30	1V	Shutdown effect
			0-9：Invalid，10V； 10: The corresponding alarm is more than 10V； 30：The corresponding alarm is more than 30V							
	02h	BusLowVoltVal	-	uint16	0-60	RW	NO	10	1V	Shutdown effect
			0：No alarm；1-9：Invalid；10: The corresponding alarm is more than 10V；							
	03h	RunCurrentVal	-	uint16	-	RW	NO	500	0.001A	Shutdown effect
	04h	Reserve1	-	uint16	-	RW	NO	-	-	
	05h	Reserve2	-	uint16	-	RW	NO	-	-	
	06h	HoldCurrentVal	-	uint16	-	RW	NO	500	0.001A	
	07h	OverCurrentVal	-	uint16	-	RW	NO	40（80）	0.1A	
	08h	StepMode	-	uint16	0-65535	RW	NO	3	-	
			00： full step 01： half step 02： 1/4 step 03： 1/8 step 04： 1/16 step							
	09h	LockCurrentVal	-	uint16		RW	NO	800	0.001A	Shutdown effect
2002h	01h	CtrlModeSelec	-	uint16	0-2	RW	NO	0	-	Shutdown effect
			0：CiA402 Mode 1：Nimotion Position Mode 2：Nimotion Velocity Mode							

Index	Sub-index	Name	Description	Data type	Data range	Access-ability	Mapp-ability	Factory setting	Unit	Activate method
	02h	CloseLoopEn	-	uint16	0-1	RW	NO	1	-	Shutdown effect
			0：Open-loop；1：Close-loop							
	03h	IAPVersion	-	uint32	-	RO	NO	-	-	Shutdown effect
	04h	-	-	uint32	-	-	-	-	-	-
	05h	HardwareVersion	-	uint32	-	RO	NO	-	-	Shutdown effect
2003h	01h	DI1FunSelec	-	uint16	-	RW	RPDO	0	-	Shutdown effect
	02h	DI1LogicSelec	-	uint16	0-4	RW	RPDO	0	-	Shutdown effect
			-							
	03h	DI2FunSelec	-	uint16	-	RW	RPDO	0	-	Shutdown effect
	04h	DI2LogicSelec	-	uint16	0-4	RW	RPDO	0	-	Shutdown effect
	05h	DI3FunSelec	-	uint16	-	RW	RPDO	0	-	
	06h	DI3LogicSelec	-	uint16	0-4	RW	RPDO	0	-	
	07h	DI4FunSelec	-	uint16	-	RW	RPDO	0	-	
	08h	DI4LogicSelec	-	uint16	0-4	RW	RPDO	0	-	
	09h	DI5FunSelec	-	uint16	-	RW	RPDO	0	-	
	0Ah	DI5LogicSelec	-	uint16	0-4	RW	RPDO	0	-	
	0Bh	IsDISetPullUp	Pull up the DI terminal （In STM57H）	uint16	0-7	RW	RPDO	0x07	-	
2004h	01h	DoStateCommSet	Communicate the given DO output state bit0(DO1)~bit1(DO2)	uint16	-	RW	RPDO	0	-	

	02h	DO1FunSelec	-	uint16	0-5	RW	RPDO	0	-	Shutdown effect
	03h	DO1LogicSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect
			-							
	04h	DO2FunSelec	-	uint16	0-5	RW	RPDO	0	-	Shutdown effect
	05h	DO2LogicSelec	-	uint16	0-1	RW	RPDO	1	-	
2005h	02h	StepAmount	-	int32	-	RW	RPDO	50	-	
	03h	StepSpd	-	uint32	0-100	RW	RPDO	100	rpm	
2006h	01h	SpeedRefSource	-	uint16	0-1	RW	RPDO	0	-	
			-							
	02h	KeypadSpeedRef	-	int16	-600~600	RW	RPDO	0	-	Immediate effect
	03h	SpeedRefAccelRampTime	-	uint16	0-65535	RW	RPDO	200	ms	Shutdown effect
	04h	SpeedRefDecelRampTime	-	uint16	0-65535	RW	RPDO	200	ms	Shutdown effect
2008h 控制系数	01h	PosLoopGain	Position loop gain	uint16	0-65535	RW	NO	200	0.00001	Immediate effect
	02h	Kpc	Current proportional coefficient	uint16	0-65535	RW	NO	500	0.01	
	03h	CtrlSpdFdBckFilterTime	Speed control filtering time	uint16	0-65535	RW	NO	10	ms	
	04h	DipSpdFdBckFilterTime	Speed display filtering time	uint16	0-65535	RW	NO	25	ms	
	05h	LockedSpdErrScale	Ratio threshold of error of gridlock velocity	uint16	0-100	RW	NO	40	1%	
	06h	LockedSpdThreshold	Gridlock speed threshold	uint16	0-100	RW	NO	6	rpm	
	07h	LockedTime	Gridlock time threshold	uint16	0-65535	RW	NO	2000	ms	
	08h	Reserve1	-	uint16	-	RW	NO	-	-	
	09h	SpdFdFwrGain	Speed feedforward gain	uint16	0-65535	RW	NO	0	0.01	
	0Ah	SpdFdFwrFilterCoeff	Velocity feedforward filter coefficient	uint16	0-1000	RW	NO	100	0.001	

	0Bh	SpdFilterCoeff	Velocity filtering coefficient	uint16	0-1000	RW	NO	500	0.001	
	0Ch	SpdLimitThreshold	Velocity change (acceleration) limiting threshold	uint16	0-1000	RW	NO	200	rpm/ms	Immediate effect
	0Dh	CurrentFdFwrGain	Current feedforward gain	uint16	0-65535	RW	NO	4000	0.1	Immediate effect
	0Eh	CurrentFilterCoeff	Current filtering coefficient	uint16	0-1000	RW	NO	1000	0.001	Immediate effect
200Bh 监控参数	01h	DriverState	-	uint16	-	RO	TPDO	-	-	-
			-							
	02h	ActualMotorSpeed	-	int16	-	RO	TPDO	-	rpm	-
	03h	MonitoredDiStates	-	uint16	-	RO	TPDO	-		-
	04h	MonitoredDoStates	-	uint16	-	RO	TPDO	-		-
	05h	MonitoredVDiStates	-	uint16	-	RO	TPDO	-		-
	06h	TotalRUNTime	Total running time	uint32	-	RO	TPDO	-	15min	-
	07h	PluseInDuty	Manufacturer speed mode pulse input duty cycle	uint16	-	RO	TPDO	-	%	-
	08h	BusVoltage	Sampled bus voltage	uint16	-	RO	TPDO	-	V	-
200Ch 通信配置参数	01h	CommunicationSelect	Communication mode selection	uint16	1-3	RW	NO	2		Shutdown effect
			1: EtherCat；2: Can；3: Serialport							
	02h	ServoShaftAddress	Node-ID	uint16	1-127	RW	NO	1	-	Restart to take effect
	03h	SerialPortBaudRate	BaudRate	uint16	0-8	RW	NO	0x08	-	

			0x00:10kbps； 0x01:20kbps； 0x02:50kbps； 0x03:100kbps； 0x04:125kbps； 0x05:250kbps； 0x06:500kbps； 0x07:800kbps； 0x08:1Mbps							
	04h	ModbusDataFormat	Modbus Data Format	uint16	0-3	RW	NO	0	-	Restart to take effect
			0x00:8 data bit, parity check, 1 stop bit 0x01:8 Data bit, parity check, 1 stop bit 0x02:8 Data bit, none Parity, and 1 stop bit 0x03:8 Data bit, none Parity, and 2 stop bit							
200Eh	01h	OverCurrent	-	uint32	-	RW	NO	0x2300	-	Immediate effect
	02h	OverVoltage	-	uint32	-	RW	NO	0x13110	-	
	03h	UnderVoltage	-	uint32	-	RW	NO	0x13120	-	
	04h	ThermalWarning	-	uint32	-	RW	NO	0x4310	-	
	05h	CAN_ErrPassiveSet	-	uint32	-	RW	NO	0x18120	-	
	06h	CAN_Overrun	-	uint32	-	RW	NO	0x18130	-	
	07h	CAN_Busoff	-	uint32	-	RW	NO	0x18140	-	
	08h	EERROM_Malfunction	-	uint32	-	RW	NO	0x15530	-	
	09h	EEPROMSelfCheckError	-	uint32	-	RW	NO	0x300FF04	-	
	0Ah	Motorlocked	-	uint32	-	RW	NO	0x03017121	-	
	0Bh	FlashInitError	-	uint32	-	RW	NO	0x15540	-	
	0Ch	DinSetError	-	uint32	-	RW	NO	0x3016320	-	
	0Dh	ThermalShutdown	-	uint32	-	RW	NO	0xFF00	-	
	0Eh	DriverChipWrongCMD	-	uint32	-	RW	NO	0xFF01	-	

	0F$_h$	DriverChipNotperfCMD	-	uint32	-	RW	NO	0xFF02	-	Immediate effect
	10$_h$	DriverChipSPIError	-	uint32	-	RW	NO	0xFF03	-	Immediate effect
	11$_h$	DriverChipSelfCheckError	-	uint32	-	RW	NO	0x1FF07	-	Immediate effect
	12$_h$	EncoderSelfCheckError	-	uint32	-	RW	NO	0x1FF06	-	Immediate effect
	13$_h$	EncoderFrameError	-	uint32	-	RW	NO	0x1FF08	-	Immediate effect
	14$_h$	EncoderCRCError	-	uint32	-	RW	NO	0x1FF09	-	Immediate effect
	15$_h$	EncoderCMDError	-	uint32	-	RW	NO	0x1FF0A	-	Immediate effect
	16$_h$	EncoderMagTooHighError	-	uint32	-	RW	NO	0x1FF0C	-	Immediate effect
	17$_h$	EncoderMagTooLowError	-	uint32	-	RW	NO	0x1FF0D	-	Immediate effect
	18$_h$	NegativeLimitSwitch	-	uint32	-	RW	NO	0x301FF0E	-	Immediate effect
	19$_h$	PositiveLimitSwitch	-	uint32	-	RW	NO	0x301FF0F	-	Immediate effect
	1A$_h$	PosRangLimit	-	uint32	-	RW	NO	0x401FF10	-	Immediate effect
2017$_h$	01$_h$	VDinEn	-	uint16	0-1	RW	RPDO	1	-	Immediate effect
			--							
	02$_h$	VDI_VirtualLevelCommSet	-	uint16	-	RW	RPDO	0	-	Immediate effect
			-							
	03$_h$	VDI1FunSelec	-	uint16	-	RW	RPDO	33	-	Shutdown effect
			-							
	04$_h$	VDI1LogicSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect

			-							
	05h	VDI2FunSelec	-	uint16	-	RW	RPDO	2	-	Shutdown effect
	06h	VDI2LogicSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect
	07h	VDI3FunSelec	-	uint16	-	RW	RPDO	38	-	Shutdown effect
	08h	VDI3LogicSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect
	09h	VDI4FunSelec	-	uint16	-	RW	RPDO	0	-	Shutdown effect
	0Ah	VDI4LogicSelec	-	uint16	0-1	RW	RPDO	0	-	Shutdown effect
	0Bh	VDI4FunSelec	-	uint16	-	RW	RPDO	0	-	Shutdown effect
	0Ch	VDI4LogicSelec	-	uint16	0-1	RW	RPDO	0	-	Shutdown effect
	0Dh	VDI6FunSelec	-	uint16	-	RW	RPDO	0	-	Shutdown effect
	0Eh	VDI6LogicSelec	-	uint16	0-1	RW	RPDO	0	-	Shutdown effect
2021h	01h	DX1InOutSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect
			-							
	02h	DX2InOutSelec	-	uint16	0-1	RW	RPDO	1	-	Shutdown effect
			-							

10.3 Sub-protocol Definition Parameter Group 6000h Description

Index	Sub-index	Name	Description	Data type	Data range	Access-ibility	Mapp-ability	Factory setting	Unit	Activate method
603Fh	00h	Error Code	See section 9.1 for details	uint16	0~65535	RO	TPDO	0	-	-
6040h	00h	Controlword	Control word	uint16	0~65535	RW	RPDO	0	1	Shutdown effect
6041h	00h	Statusword	Status word	uint16	0~65535	RO	TPDO	0	1	-
6042h	00h	VI target velocity	Target velocity in VM mode	int16	-3000~3000	RW	RPDO	0	1rpm	Shutdown effect
6043h	00h	VI velocity demand	Velocity demand in VM	int16	-3000~3000	R0	NO	0	1rpm	-
6046h	01h	VI velocitymin amount	VM mode speed min.	uint32	0~3000	RW	NO	0	1rpm	Shutdown effect
	02h	VI velocity max amount	VM mode speed maximum	uint32	0~3000	RW	NO	300	1rpm	Shutdown effect
6048h	01h	delta speed	Acceleration in VM=delta speed/delta time	uint32	0~300000	RW	NO	500	1rpm	Shutdown effect
	02h	delta time		uint16	1~65535	RW	NO	1	1s	Shutdown effect
6049h	01h	delta speed	Deceleration in VM=delta speed/delta time	uint32	0~300000	RW	NO	500	1rpm	Shutdown effect
	02h	delta time		uint16	1~65535	RW	NO	1	1s	Shutdown effect
604Ah	01h	delta speed	Braking acceleration in VM=delta speed/delta time	uint32	0~300000	RW	NO	1000	1rpm	Shutdown effect
	02h	delta time		uint16	1~65535	RW	NO	1	1s	Shutdown effect
604Ch	01h	vl dimension factor numerator	Scaling factor in VM	int32	-214748364~ 2147483647	RW	NO	0	1	Shutdown effect
	02h	vl dimension factor denominator		int32	-214748364~ +2147483647	RW	NO	0	1	Shutdown effect
		Sub index 1 contains the numerator (multiplier) used to calculate speed, and sub index 2 contains the denominator (divisor). Speed unit: rpm * sub index 1 / sub index 2. For example, if sub index 1 is set to 2 and sub index 2 is set to 1, then speed unit: 2rpm. If any of the sub indexes 1 and 2 is 0, the speed unit is rpm.								
605Ah	00h	Quick stop option code	See section 4.1.4 for details	int16	0~65535	RW	NO	2	1	Shutdown effect
605Bh	00h	Shutdown option code	See section 4.1.4 for details	int16	0~65535	RW	NO	0	1	Shutdown effect
605Ch	00h	Disable operation option code	See section 4.1.4 for details	int16	0~65535	RW	NO	0	1	Shutdown effect
605Dh	00h	Halt option code	See section 4.1.4 for details	int16	0~2	RW	NO	1	1	Shutdown effect
605Eh	00h	Fault reaction option code	See section 4.1.4 for details	int16	0~4	RO	NO	0	1	Shutdown effect
6060h	00h	Mode Operation	See section 6.2 for details	int8	0~10	RW	RPDO	1	1	Shutdown effect
6061h	00h	Modes of operation display	See section 6.2 for details	int8	0~10	RO	TPDO	1	1	-
6062h	00h	Position demand value	Position instruction value in effect inside the controller	int32	-2147483648~ +2147483647	RO	NO	0	User unit	-
6063h	00h	Position actual encoder value	-	int32	-2147483648~ +2147483647	RO	TPDO	0	Encoder unit	-
6064h	00h	Position actual user value	-	int32	-2147483648~ +2147483647	RO	TPDO	0	User unit	-
		Description: position feedback 6064h × gear ratio (6091h) = position feedback 6063h								

6065h	00h	Following error Window	-	uint32	0~4294967295	RW	NO	50	User unit	Shutdown effect

		Description: when the position deviation exceeds ± 6065h and the duration time reaches 6066h, Tracking error is triggered							
6066h	00h	Following error time out	-	uint16	0~65535	RW	NO	10000	1ms	Shutdown effect
		Description: when the position deviation exceeds ± 6065h and the duration time reaches 6066h, Tracking error is triggered							
6067h	00h	Position Window	-	uint32	0~4294967295	RW	NO	10	User unit	Shutdown effect
		Description: when the position deviation exceeds ± 6067h and the duration reaches 6068h, Target position is reached.Bit10 of status word 6041h=1 in position mode							
6068h	00h	Position Window time	-	uint16	0~65535	RW	NO	5	1ms	Shutdown effect
		Description: when the position deviation exceeds ± 6067h and the duration reaches 6068h, Target position is reached.Bit10 of status word 6041h=1 in position mode							
606Bh	00h	Velocity demand value	Velocity instruction value in effect inside the controller	int32	-2147483648~ +2147483647	RO	NO	0	User unit/s	-
606Ch	00h	Velocity actual value	Current velocity actual value feedback	int32	-2147483648~ +2147483647	RO	TPDO	0	rpm	-
606Dh	00h	Velocity Window	-	uint16	0~65535	RW	NO	100	rpm	Shutdown effect
		Description: when the difference between the target speed in 60FFh (converted into motor speed rpm) and the actual speed of the motor is within ± 606D, and the duration time reaches 606Eh, means that the speed is reached, and the bit10 in status word 6041h = 1							
606Eh	00h	Velocity Window time	-	uint16	0~65535	RW	NO	5	ms	Shutdown effect
		Description: when the difference between the target speed in 60FFh (converted into motor speed rpm) and the actual speed of the motor is within ± 606D, and the duration time reaches 606Eh, means that the speed is reached, and the bit10 in status word 6010h = 1							
606Fh	00h	Velocity threshold	Threshold of 0º arrival	uint16	0~65535	RW	NO	5	User unit/s	Shutdown effect
		Description: the current speed is under 6069h within ± 606Fh, and the duration time reaches 6070h,means that the speed is 0, and the bit12 in status word 6041h = 1							
6070h	00h	Velocity threshold time	Time window when 0º arrival	uint16	0~65535	RW	NO	5	ms	Shutdown effect
		Description: the current speed is under 6069h within ± 606F, and the duration time reaches 6070h,means that the speed is 0, and the bit12 in status word 6041h = 1							
607Ah	00h	Target position	-	int32	-2147483648~ +2147483647	RW	RPDO	0	User unit	Shutdown effect
607Bh	01h	MinPosRangLimt	-	int32	-2147483648~ +2147483647	RW	NO	-60000000	User unit	Shutdown effect
	02h	MaxPosRangLimt	-	int32	-2147483648~ +2147483647	RW	NO	60000000	User unit	Shutdown effect
607Ch	00h	Home offset	-	int32	-2^{31}~(2^{31}-1)	RW	NO	0	User unit	Shutdown effect
		Description: set the physical position where the mechanical home position deviates from the motor home position in the Homing mode; when the homing operation is completed, takes effect when bit15= 1 in status word 6041h							
607Dh	01h	MinPosLimt	-	int32	-2^{31}~(2^{31}-1)	RW	NO	-60000	Instruction unit	Shutdown effect
	02h	MaxPosLimt	-	int32	-2^{31}~(2^{31}-1)	RW	NO	60000	Instruction unit	Shutdown effect
607Eh	00h	Polarity	-	uint8	0~255	RW	NO	0	1	Shutdown effect
607Fh	00h	Max profile velocity	-	uint32	0~500000	RW	RPDO	40000 （Close-loop） 16000 （Open-loop）	User unit/s	Shutdown effect
6080h	00h	Max motor speed	-	uint32	0~500	RW	RPDO	600	rpm	Shutdown effect

6081h	00h	Profile velocity	-	uint32	0~500000	RW	NO	12000 (Close-loop) 4800 (Open-loop)	User unit/s	Shutdown effect
	Description: set the uniform speed in this position command section of the Profile position mode,									
6082h	00h	End velocity	-	uint32	0~500000	RW	NO	0	User unit/s	Shutdown effect
6083h	00h	Profile acceleration	-	uint32	0~4294967295	RW	NO	40000 (Close-loop) 16000 (Open-loop)	User unit/s^2	Shutdown effect
6084h	00h	Profile deceleration	-	uint32	0~4294967295	RW	NO	120000 (Close-loop) 48000 (Open-loop)	User unit/s^2	Shutdown effect
6085h	00h	Quick Stop deceleration	-	uint32	0~4294967295	RW	NO	400000 (Close-loop) 160000 (Open-loop)	User unit/s^2	Shutdown effect
6086h	00h	Motion profile type	-	int16	0~3	RW	NO	0	1	Shutdown effect
	Description: set the curve type of motor position command or speed command. 0 ~ linear, 3 ~ s curve									
608Fh	01h	Encoder increments	-	uint32	0~4294967295	RW	NO	4000(Close-loop) 1600 (Open-loop)	1	Shutdown effect
	02h	Motor revolutions	-	uint32	0~4294967295	RW	NO	1	1	Shutdown effect
	Description: used to set the turns of the motor corresponding to the encoder increment Resolution of position encoder= $\frac{\text{increment of encoder(608Fh:01h)}}{\text{motor turns (608Fh:02h)}}$									
6091h	01h	Motor revolutions	-	uint32	0~4294967295	RW	NO	1	1	Shutdown effect
	02h	Shaft revolutions		uint32	0~4294967295	RW	NO	1	1	Shutdown effect
	Description: it is used to establish the proportional relationship between the load axis displacement and the motor axis displacement specified by the user. See Section 5.2 for details									
6098h	00h	Homing method	See section 6.6.4 for details	int8	17~30	RW	NO	24	1	Shutdown effect
6099h	01h	Speed during search for switch	(high speed)	uint32	0~4294967295	RW	NO	12000	User unit/s	Shutdown effect
	02h	Speed during search for zero	(low speed)	int32	-2147483648~ +2147483647	RW	NO	4000	User unit/s	Shutdown effect
609Ah	00h	Homing acceleration	-	uint32	0~4294967295	RW	NO	80000	User unit/s	Shutdown effect

								（Open-loop）200000（Close-loop）		
60A3h	00h	Profile jerk use	Number of position jerk applications,This device only uses 60A4h:01h	uint8	2	RW	NO	2	1	Shutdown effect
60A4h	01h	Profile jerk 1	-	uint32	0~4294967295	RW	NO	15000	User unit/s³	Shutdown effect
	02h	Profile jerk 2	-	uint32	0~4294967295	RW	NO	30000	User unit/s³	Shutdown effect
60B0h	00h	Position offset	-	int32	-2147483648~+2147483647	RW	NO	0	User unit	Shutdown effect
		Description: set the servo velocity command offset in the Cyclic sync velocity mode, after the offset:								
60B1h	00h	Velocity offset	-	int32	-214748364~2147483647	RW	NO	0	User unit	Shutdown effect
		Description: set the servo velocity command offset in the Cyclic sync velocity mode, after the offset: Servo target velocity = 60FFh+60B1h								
60C1h	01h	Interpolation data record	-	int32	-2147483648~+2147483647	RW	RPDO	0	Instruction unit	Shutdown effect
60C2h	01h	Interpolation time period value	-	uint8	0~255	RW	NO	20	1	Shutdown effect
	02h	Interpolation time index	-	Int8	-128~+127	RW	NO	-3	10^ns	Shutdown effect
	Description: interpolation period = t×10^n seconds									
60C5h	00h	Max acceleration	-	uint32	0~4294967295	RW	NO	500000	User unit/s²	Shutdown effect
60C6h	00h	Max deceleration	-	uint32	0~4294967295	RW	NO	500000	User unit/s²	Shutdown effect
60F2h	00h	Positioning option code		uint16	0~2	RW	NO	0	1	Shutdown effect

<table>
<tr><td></td><td></td><td colspan="9"><table>
<tr><td>Bit1</td><td>Bit0</td><td>Description:</td></tr>
<tr><td>0</td><td>0</td><td>Relative position shift, relative to the target position（607A_h）</td></tr>
<tr><td>0</td><td>1</td><td>Relative position shift, relative to the position command（60FC_h）</td></tr>
<tr><td>1</td><td>0</td><td>Relative position shift, relative to the position command（6064_h）</td></tr>
</table></td></tr>
<tr><td>60F4_h</td><td>00_h</td><td>Following error actual value</td><td>-</td><td>int32</td><td>-2147483648~ +2147483647</td><td>RO</td><td>NO</td><td>0</td><td>User unit</td><td>-</td></tr>
<tr><td>60FC_h</td><td>00_h</td><td>Position demand internal value</td><td>-</td><td>int32</td><td>-2147483648~ +2147483647</td><td>RO</td><td>TPDO</td><td>0</td><td>User unit</td><td>-</td></tr>
<tr><td colspan="10">Description :Position command (60FC_h)= Position command(6062_h)x Electronic gear ratio(6091_h)</td></tr>
<tr><td>60FF_h</td><td>00_h</td><td>Target velocity</td><td>-</td><td>int32</td><td>-2147483648~ +2147483647</td><td>RW</td><td>RPDO</td><td>0</td><td>User unit/s</td><td>Shutdown effect</td></tr>
<tr><td colspan="10">Description: set the speed command in the Profile velocity mode and Cyclic sync velocity mode</td></tr>
</table>

11 Version

Manual version	Date	Revision History
A	2025/2/6	Establishing
A01	2026/1/15	Modify description

- 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

