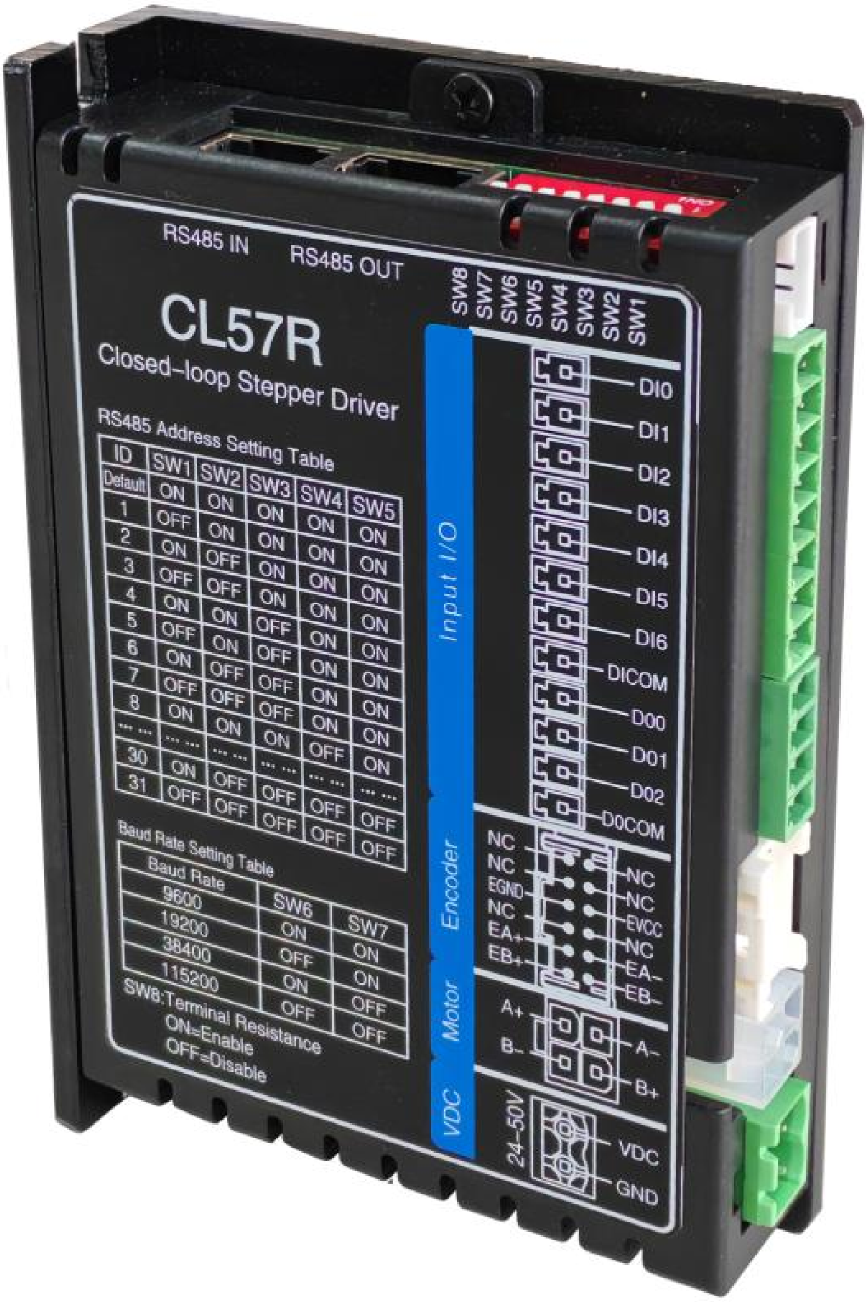
CL57R

RS485 Modbus digital nema23 hybrid servo motor



RS485 Address Setting Table

ID	SW1	SW2	SW3	SW4	SW5
Default	ON	ON	ON	ON	ON
1	OFF	ON	ON	ON	ON
2	ON	OFF	ON	ON	ON
3	OFF	OFF	ON	ON	ON
4	ON	ON	OFF	ON	ON
5	OFF	ON	OFF	ON	ON
6	ON	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON	ON
8	ON	ON	ON	OFF	ON
... ...	... ...	... ...	... ...	... ...	... ...
30	ON	OFF	OFF	OFF	OFF
31	OFF	OFF	OFF	OFF	OFF

Baud Rate Setting Table

Baud Rate	SW6	SW7
9600	ON	ON
19200	OFF	ON
38400	ON	OFF
115200	OFF	OFF

SW8:Terminal Resistance
ON=Enable
OFF=Disable

1. Product description

1.1product description

CL57R is a closed-loop stepper drive product based on the RS485 Modbus that supports the Modbus RTU protocol. This product integrates a single-axis controller function and uses the standard Modbus RTU bus communication protocol for control. It can mount a maximum of 32 axes and can realize multi-axis bus synchronous control. The driver has position control, speed control, 16-segment internal position and 16-segment Internal speed control, support for zero return, absolute/relative positioning, JOG and other functions, can be directly controlled using a touch screen or a controller with an RS485 interface.

1.2 Features

- Supports standard Modbus RTU protocol on RS485 bus ;
- No loss of step, accurate positioning ;
- The current size is intelligently adjusted according to the load to adapt to various mechanical load conditions.;
- Built-in acceleration, deceleration and smoothing filter control for smoother operation;
- User-defined segments;
- Supports position, speed, zero return, JOG, multi-stage position, multi-stage speed and other modes ;
- 7 input ports, 3 output port functions all programmable and configurable;
- Voltage range: DC+24V~48V;
- With overcurrent, overvoltage, position tolerance and other protections;

1.3 typical application

Suitable for various small and medium-sized automation equipment and instruments, such as industrial robots, textile machinery, special industrial sewing machines, wire stripping machines, marking machines, cutting machines, laser phototypesetting, plotters, CNC machine tools, engraving machines, automatic assembly equipment, etc. The application effect is particularly good in equipment where users expect low noise and high speed.

2. Electrical, mechanical and environmental specifications

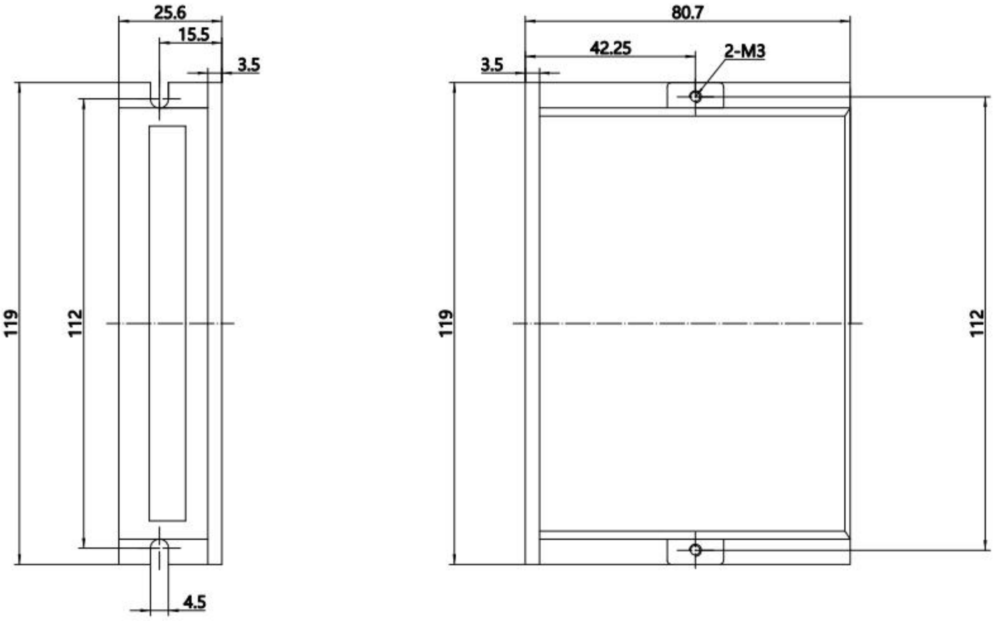
2.1Electrical indicators

Power supply	DC24~48V, recommended power supply DC36V
Output current	Peak 6.0A (current changes with load)
DI input current	10~50mA
DI input voltage	+24VDC
Communication type	RS485
Maximum communication rate	115200bps

2.2 Usage environment and parameters

cooling method	Natural cooling or external radiator	
Usage environment	Use occasions	Try to avoid dust, oil and corrosive gases
	temperature	0~50℃
	humidity	40~90%RH
	vibration	$5.9m/s^2$Max
storage temperature	-20℃~80℃	
weight		

2.3 Driver installation dimensions



3. Introduction to driver interface and wiring

3.1 Interface definition

(1) Power input port

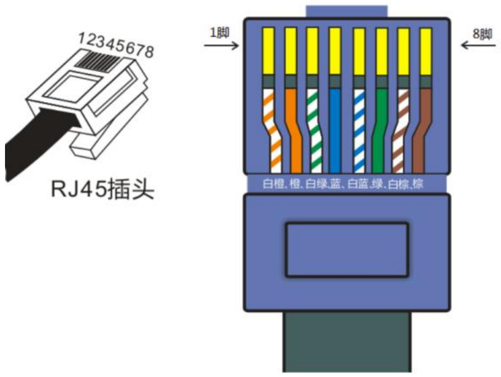
Terminal number	symbol	name	illustrate
1	+Vdc	DC power supply positive terminal	DC+24V~48V Recommended DC+36V power supply
2	GND	DC power ground	

(2) Motor port

pin	Symbol	Description
1	A+	A phase motor winding +
2	A-	A phase motor winding-
3	B+	B phase motor winding +
4	B-	B phase motor winding -

(3) RS485 communication port

引脚	Signal definition	Network cable color
1	RS485+	white orange
2	RS485-	orange
3	NC	White and green
4	NC	blue
5	GND	White and blue
6	GND	green
7	NC	white brown
8	NC	Brown



水晶头引脚顺序

(4) Encoder port

pin	definition	illustrate
1	SHIELD	Encoder shield
2	NC	RS485-
3	NC	
4	NC	
5	EVCC	Encoder power supply positive terminal
6	EGND	Encoder power supply negative terminal
7	NC	
8	NC	
9	EB+	
10	EB-	
11	EA+	
12	EA-	

(5) DI/DO port

Terminal number	Symbol definition	illustrate
1	DI0	Single-ended input port valid working voltage 24V
2	DI1	
3	DI2	
4	DI3	
5	DI4	
6	DI5	
7	DI6	
8	DICOM	Input port common port: Compatible with common anode and common cathode connection method
9	DO0	Single-ended output port
10	DO1	
11	DO2	
12	DOCOM	output port common port: Negative pole of power supply

(6) status indication

PWR：Power Indicator

When power is on, the green indicator light is always on.

ALM: Fault indicator.

The red light flashes once within 3 seconds: over current or phase-to-phase short circuit fault;

The red light flashes 2 times continuously within 3 seconds: over voltage fault;

The red light flashes 7 times continuously within 3 seconds: the position error exceeds the tolerance alarm.

4. DIP switch setting

CL57R uses a 5-digit DIP switch to set the driver station number and a 2-digit DIP switch to set the communication baud rate.

SW1~SW5: Driver station number setting. SW6~SW7: driver communication baud rate. The slave station number and communication baud rate need to be powered on again to take effect after being modified.

slave station number	SW1	SW2	SW3	SW4	SW5
default	on	on	on	on	on
1	off	on	on	on	on
2	on	off	on	on	on
3	off	off	on	on	on
4	on	on	off	on	on
5	off	on	off	on	on
6	on	off	off	on	on
7	off	off	off	on	on
8	on	on	on	off	on
9	off	on	on	off	on
10	on	off	on	off	on
11	off	off	on	off	on
12	on	on	off	off	on
13	off	on	off	off	on
14	on	off	off	off	on
15	off	off	off	off	on
16	on	on	on	on	off
17	off	on	on	on	off
18	on	off	on	on	off
19	off	off	on	on	off
20	on	on	off	on	off
21	off	on	off	on	off

22	on	off	off	on	off
23	off	off	off	on	off
24	on	on	on	off	off
25	off	on	on	off	off
26	on	off	on	off	off
27	off	off	on	off	off
28	on	on	off	off	off
29	off	on	off	off	off
30	on	off	off	off	off
31	off	off	off	off	off

Note: When setting the slave station number in the default file, you can define the slave station number by setting the custom driver slave station off number register (0x0020), ranging from 1 to 31.

Communication baud rate	SW6	SW7
9600	on	on
19200	off	on
38400	on	off
115200	off	off

Note: When the communication baud rate is set to 9600bps, the serial port data format is fixed to 8 data bits, no parity, and 1 stop bit. When set to the other three baud rates, the serial port data format is determined by the serial port data format register (0x0021).

SW8: RS485 terminal resistor. The driver at the end of the bus needs to set this DIP switch to ON, and the other drivers to OFF.

5. MODBUS communication protocol

5.1 MODBUS register address definition

Register address	name	illustrate	Scope (property)	default value
Monitoring parameters				
0x0000	Drive model	Drive model code	（RO）	5712
0x0001	Software version	Software version	（RO）	100
0x0002	Drive working mode	0: No operation in any mode 1: Location mode 2: Speed mode 3: Return to zero mode 4: JOG mode 5: Multi-segment position mode	（RO）	0

		6: Multi-speed mode		
0x0003	status word	Bit0: in place Bit1: Return to zero completed Bit2: Motor running Bit3: Alarm Bit4: Motor enable Bit5~Bit15: reserved	（RO）	
0x0004	error code	0: normal 1: Overcurrent 2: Overvoltage 4: Position is too poor	（RO）	
0x0005	Input terminal status flag	Bit0: X0 terminal input status Bit1: X1 terminal input status Bit2: X2 terminal input status Bit3: X3 terminal input status Bit4: X4 terminal input status Bit5: X5 terminal input status Bit6: X6 terminal input status 0: Input terminal disconnected 1: Input terminal closed	（RO）	
0x0006	Output terminal status flag	Bit0: Y0 terminal output status Bit1: Y1 terminal output status Bit2: Y2 terminal output status 0: The output end optocoupler is disconnected 1: The output end optocoupler is closed	（RO）	
0x0007	Current position is high	The upper 16 bits of the current position in absolute position	（RO）	

0x0008	Current position is low	The lower 16 bits of the current position in absolute position	（RO）	
0x0009	current speed	Current motor running speed unit: rpm	（RO）	
Input and output control				
0x0010	DI effective level	Bit0: Input terminal X0 control bit Bit1: Input terminal X1 control bit Bit2: Input terminal X2 control bit Bit3: Input terminal X3 control bit Bit4: Input terminal X4 control bit Bit5: Input terminal X5 control bit Bit6: Input terminal X6 control bit Bit7~Bit15: reserved 0: default 1: Level inversion	0~7 （RW/S）	0
0x0011	DI filter time	DI filter time, unit ms	0~1000	2
0x0012	DO effective level	Bit0: Output terminal Y0 control bit Bit1: Output terminal Y1 control bit Bit2: Output terminal Y0 control bit Bit3~Bit15: reserved 0: default 1: Level inversion	0~7 （RW/S）	0

0x0013	Input terminal X0 terminal function selection	0: no definition;
		1: Origin signal;
		2: Positive limit signal;
		3: Negative limit signal;
		4: Motor release signal:
		5: Alarm clear signal;
		6: Stop signal;
		7: Emergency stop signal;
			0~16（RW）	1

0x0014	Input terminal X1 terminal function selection		0~16（RW）	2
0x0015	Input terminal		0~16（RW）	3

	X2 terminal function selection	8: Forward JOG;
		9: Negative JOG;
		10: Return to zero trigger;
		11: Position path trigger;
		12: Speed path trigger;
		13: path address 0;
		14: path address 1;
		15: path address 2;
		16: path address 3;		

0x0016	Input terminal X3 terminal function selection		0~16（RW）	0
0x0017	Input terminal X4 terminal function selection		0~16（RW）	0
0x0018	Input terminal X5 terminal function selection		0~16（RW）	0
0x0019	Input terminal X6 terminal function selection		0~16（RW）	0

0x001A	Output terminal Y0 terminal function selection	0: no definition
		1: Alarm signal;
		2: Motor running;
		3: Return to zero completed;
		4: In place signal;
		5: Brake signal;
			0~5（RW）	1

0x001B	Output terminal Y1 terminal function selection		0~5（RW）	0
0x001C	Output terminal Y2 terminal function selection		0~5（RW）	0
Servo basic parameters				
0x0020	Custom drive slave number	1~31, customized slave address, SW1~SW5 of the DIP switch takes effect in the default gear. Others: invalid	1~31（RW/S）	1
0x0021	Serial port data format	0: 8 data bits, no parity, 1 stop bit; 1: 8 data bits, no parity, 2 stop bits; 2: 8 data bits, even parity, 1	0~3（RW/S）	0

		stop bit; 3: 8 data bits, odd parity, 1 stop bit; Note: After modification, you need to power on again to take effect.		
0x0022	Default orientation settings	0：CCW 1：CW	0~1 （RW/S）	0
0x0023	Segmentation settings	Number of subdivisions set	400~51200 （RW/S）	1000
0x0024	Whether the communication write function code value is updated to EEPROM synchronously	0: No update 1: Parameters with attribute RW/S are synchronized to EEPROM	0~1（RW/S）	0
0x0025	Over-travel parking method	0: Slow down and stop 1: Emergency stop	0~1 （RW/S）	0
Position speed motion control parameters				
0x0030	starting speed	Starting speed unit rpm	2~300	5
0x0031	acceleration time	Acceleration time unit ms	1~2000 （RW）	100
0x0032	deceleration time	deceleration time	1~2000 （RW）	100
0x0033	Maximum speed	Maximum speed unit rpm	-3000~3000 （RW）	60
0x0034	The total number of pulses at the target position is high	The total number of pulses at the target position is 16 bits higher	-2^{31}~$2^{31}-1$ pulse （RW）	0
0x0035	The total number of pulses at the target position is 16 bits higher	The total number of pulses at the target position is 16 bits lowert		5000
0x0036	motion control word	Bit0: Positioning control bit 0: Invalid, 1: Valid Bit1: Positioning mode bit 0: relative position, 1:	0~127 （RW）	0

		absolute position Bit2: Switching mode bit 0: Ignore new instructions when there is positioning operation 1: Interrupt the current positioning operation to execute new instructions Bit3: Speed control bit 0: Invalid 1: Valid Bit4: Zero return control bit 0: Invalid 1: Valid Bit5: Stop control bit 0: Invalid 1: Valid Bit6: Emergency stop control bit 0: Invalid 1: Valid		
0x0037	Auxiliary control	0: invalid 0x0001: Restore factory settings 0x0002: Save parameters 0x0004: Alarm cleared 0x0008: The current position can be cleared when the absolute position is used	0~8（RW）	0
0x0038	Motor enable/release	0: Release 1: enable	0~1（RW）	1
0x0039	Multi-segment position trigger mode	0: Ignore new instructions when positioning is running 1: Interrupt the current positioning operation to execute new instructions	0~1（RW）	0

0x003A	Multi-segment position absolute position/relative position mode settings	0: Relative position mode;
		1: Absolute position mode;
			0~1（RW）	0

Zero return and JOG control parameters				
0x0040	Zero return method	17: Negative limit approaching 18: Positive limit approach	0~35 （RW/S）	0

		24: Positive limit + origin approach 29: Negative limit + origin approach 35: The current position is the origin		
0x0041	Return to zero speed	Fast running speed close to zero point, unit rpm	5~3000 （RW/S）	120
0x0042	Return to zero crawling speed	Query the slow running speed of zero point, unit rpm	5~300 （RW/S）	60
0x0043	Zero return acceleration and deceleration time	Zero return acceleration and deceleration time, unit ms	1~2000 （RW/S）	200
0x0044	Origin compensation value H	Origin compensation value H, unit pulse	0~65535 （RW/S）	0
0x0045	Origin compensation value L	Origin compensation value L, unit pulse	0~65535 （RW/S）	0
0x0046	JOG running speed	Unit: r/min	0~3000 （RW）	60
0x0047	JOG operation acceleration time	Unit: ms	1~2000 （RW）	100
0x0048	JOG operation deceleration time	Unit: ms	1~2000 （RW）	100
Servo advanced parameters				
0x0050	Open and closed loop operating modes	0: Open loop, the full current during open loop operation is determined by the open loop holding current register (0x0055), with automatic half-current function 1: Closed loop	0~1 （RW/S）	1
0x0051	Encoder resolution	0: 1000 lines 1: 2500 lines	0~1	0
0x0052	Tracking error alarm threshold	Encoder resolution is in units of	400~65535 （RW/S）	4000

0x0053	Basic current ratio	Unit mA	0~60（RW/S）	30
0x0054	Closed loop operating current ratio	Unit mA	0~60（RW/S）	60
0x0055	Open loop operating current ratio	Unit mA	0~60（RW/S）	45
0x0056	Instruction filter time	Unit 50us	0~1024（RW/S）	60
0x0057	Current proportional gain	Factory default and cannot be modified	（RW/S）	1000
0x0058	Current integral gain	Factory default and cannot be modified	（RW/S）	200
0x0059	Position loop proportional gain	Factory default, no modification is recommended.	0~~1000（RW/S）	300
0x005A	speed proportional gain	Factory default, no modification is recommended.	0~1000（RW/S）	400
0x005B	Speed feed forward gain	Factory default, no modification is recommended.	0~300（RW/S）	80
Multiple locations				
0x0060~0x006F	High bit of the total number of pulses in the position path (Path 0~Road Jin 15)	The total number of position pulses is 16 bits higher	-2^{31}~$2^{31}-1$ pulse（RW）	0
0x0070~0x007F	The total number of pulses in the position path is low (path 0 ~ road strength 15)	The total number of position pulses is 16 bits lower		5000
0x0080~0x008F	Position path running speed (path 0 ~ road strength 15)	Unit: r/min	0~3000（RW）	120
0x0090~0x009F	Position path	Unit: ms	1~2000	100

	acceleration time (path 0 ~ road strength 15)		（RW）	
0x00A0~0x00AF	Position path deceleration time (path 0 ~ road strength 15)	Unit: ms	1~2000 （RW）	100
Multiple speeds				
0x00B0~0x00BF	Speed Road King running speed (path 0 ~ Road King 15)	Unit: r/min	-3000~3000 （RW）	120
0x00C0~0x00CF	Speed Road King acceleration time (Path 0 ~ Road King 15)	Unit: ms	1~2000 （RW）	100
0x00D0~0x00DF	Speed path deceleration time (path 0 ~ road strength 15)	Unit: ms	1~2000 （RW）	100

Note: Attributes RO - read-only, RW - readable and writable, RW/S - readable and writable, and are updated to the EEPROM memory synchronously when writing.

Note: Use function code 03H to read RO, RW and RW/S, and use function codes 06H and 10H to write RW and RW/S.

The 0x prefix indicates that it is followed by a hexadecimal number.

5.2 MODBUS common function code

MODBUS RTU communication data structure

STX (communication start)	The minimum time interval from the previous frame is 3.5 characters
ADR (slave address)	slave address
CMD (function code)	Function number: 1byte
DATA(n-1)	Data content
…….	
DATA(0)	
CRC (check code)	Check code: 2Byte
END (end of	The minimum time interval to the next frame is

communication)	3.5 characters

Each item in the communication data format box must be explained

STX: The minimum time interval from the previous frame is 3.5 characters.

ADR: slave address.

CMD (function code) and data characters: support **common function codes: 03H, 06H, 10H.**

- **◆Function code 03H (read holding register, readable and writable): read N words (16bit)**

For example: read the current position of the slave station number 01H. The current position is the register address 0x000A and 0x000B. Assume that the content of the 0x000A register is 0x0001 and the content of the 0x0009 register is 0x0C80.

command information

ADDR (slave address)	01H
CMD (function code)	03H
initial address	00H (high byte)
	0AH (low byte)
Number of registers	00H (high byte)
	02H (low byte)
CRC16 low byte	E4H
CRC16 high byte	09H

Instruction sequence: ″01 03 00 0A 00 02 E4 09″

response message

ADDR (slave address)	01H
CMD (function code)	03H
Number of bytes	04H
0x000A address register content	00H
	01H
0x000B address register content	0CH
	80H
CRC16 low byte	AFH
CRC16 high byte	53H

Response sequence: “01 03 04 00 01 0C 80 AF 53”

Exception response information

ADDR (slave address)	01H
CMD (function code) +0x80	83H
Exception information	01H,02H,03H or 04H
CRC16 low byte	
CRC16 high byte	

Meaning of exception information: 01H—Illegal function code, 02H—Illegal register address, 03H—Illegal data, 04H—CRC check error

◆Function code 06H (write a single holding register, readable and writable): write 1 word (16bit).

For example: write 100 (0x0064) into the maximum speed register with the slave station number 01H and the register address 0x0022.

command information

ADDR (slave address)	01H
CMD (function code)	06H
initial address	00H (high byte)
	22H (low byte)
Register value	00H（high byte）
	64H（low byte）
CRC16 low byte	28H
CRC16 high byte	2BH

response message

ADDR (slave address)	01H
CMD (function code)	06H
initial address	00H（high byte）
	22H（low byte）
Register value	00H（high byte）
	64H（low byte）
CRC16 low byte	28H
CRC16 high byte	2BH

Exception response information

ADDR（slave address）	01H
CMD（function code）+0x80	86H
Exception information	01H,02H,03H or 04H
CRC16 low byte	
CRC16 high byte	

Meaning of exception information: 01H—Illegal function code, 02H—Illegal register address, 03H—Illegal data, 04H—CRC check error

◆ Function code 10H (write multiple holding registers, readable and writable): write multiple holding registers (each register is 16 bits long).

For example: when writing to the drive target location with the slave station number 01H, write 0x0001 into the register with the address 0x0023. Write 0x0C80 to the register at address 0x0024

command information

ADDR (slave address)	01H
CMD (function code)	10H

initial address	00H
	23H
Number of registers	00H
	02H
Number of bytes	04H
0x0023 register value high byte	00H
0x0023 register value low byte	01H
0x0024 register value high byte	0CH
0x0024 register value low byte	80H
CRC16 low byte	D2H
CRC16 high byte	60H

Meaning of exception information: 01H—Illegal function code, 02H—Illegal register address, 03H—Illegal data, 04H—CRC check error

response message

ADDR（slave address）	01H
CMD（function code）	10H
initial address	00H
	23H
Number of registers	00H
	02H
CRC16 low byte	B0H
CRC16 high byte	02H

Exception response information

ADDR（slave address）	01H
CMD（function code）+0x80	90H
Exception information	01H,02H,03H or 04H
CRC16 low byte	
CRC16 high byte	

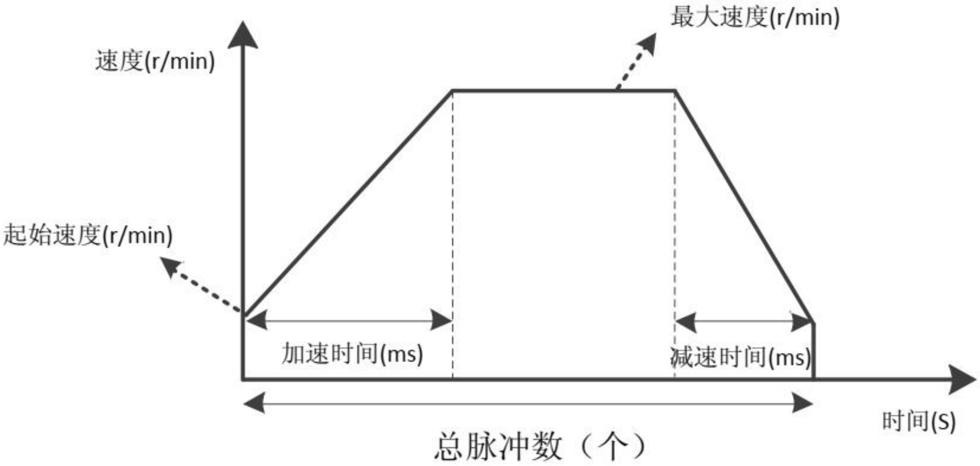
Meaning of exception information: 01H—Illegal function code, 02H—Illegal register address, 03H—Illegal data, 04H—CRC check error

6. Introduction to motion control functions

6.1 location mode

Position mode refers to the point-to-point motion control mode implemented by parameters such as starting speed (0x0030), acceleration time (0x0031), deceleration time (0x0032), maximum speed (0x0033), and total number of pulses at the target position (0x0034, 0x0035). , implemented using a trapezoidal acceleration and deceleration curve. When the target position pulse number is a positive number, it is defined as forward rotation, and when it is a negative number, it is defined as reverse rotation. Positioning operation is triggered via the control word (0x0036).

The acceleration and deceleration curve is divided into two situations. One is to accelerate to the maximum speed. The entire movement process is divided into three parts: acceleration, uniform speed and
deceleration.



Position mode acceleration and deceleration curve

The second is that the number of pulses is too small and it cannot accelerate to the maximum speed. The entire motion process is divided into two parts: acceleration and deceleration, and there is no uniform speed part.

Setting steps for location mode:

（1）Drive enable operation

After the driver is powered on or reset, if the driver is in a disabled state. Write 1 to the motor enable/release register (0x0038) to enable the motor and power on.

（2）Set location operating parameters

Set parameters such as starting speed (0x0030), acceleration time (0x0031), deceleration time (0x0032), maximum speed (0x0033), total number of pulses at the target position (0x0034, 0x0035).

（3）**start/ stop**

After the above position operation parameters are set, write 0x0001 to the control word register (0x0036) to start position mode operation, write 0x0010 to decelerate and stop, and write 0x0020 for emergency stop.

Position mode programming example (assuming the slave station number is 1, set it with DIP switches SW1~SW4, and use SW5~SW6 to select the communication baud rate)

------Motor power on-----

01 06 00 38 00 01 C9 C7 “The motor is enabled and powered”

-----Set position control parameters-----

01 06 00 30 00 1E 09 CD “Set the starting speed to 30rpm”

01 06 00 31 00 C8 D9 93 “Set the acceleration time to 200ms”

01 06 00 32 00 C8 29 93 “Set the deceleration time to 200ms”

01 06 00 33 00 78 79 E7 “Set maximum speed to 120rpm”

01 10 00 34 00 02 04 00 00 0C 80 F5 E8 “Set the total number of pulses to 3200”

It is also possible to write position control parameters in one go 01 10 00 30 00 06 0C 00 1E 00 C8 00 C8 00 78 00 00 0C 80 A4 48

-----Start position control-----

01 06 00 36 00 01 A8 04

-----Deceleration to stop-----

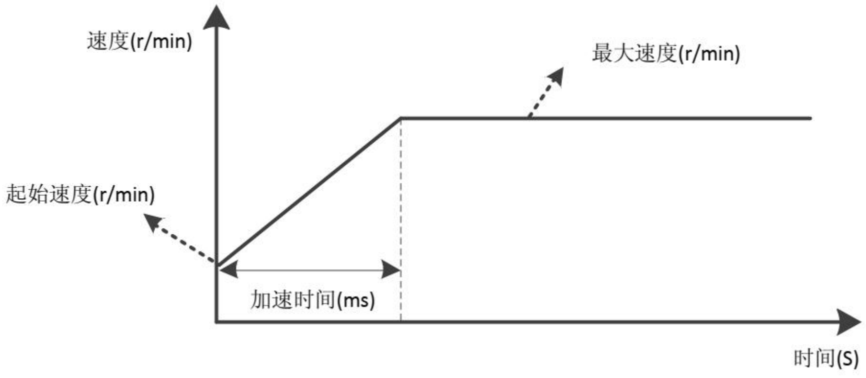
01 06 00 36 00 20 68 1C

-----emergency stop-----

01 06 00 36 00 40 68 34

6.2speed mode

Speed mode refers to the speed control mode determined by acceleration time, deceleration time and maximum speed parameters. After the motor accelerates to the maximum speed according to these three parameters, it runs at a constant speed at the maximum speed. During the movement, you can choose deceleration stop or emergency stop through the motion control word. When the maximum speed is positive, the motor is defined as forward rotation; when the maximum speed is negative, the motor is defined as reverse rotation.



Speed mode acceleration curve

Speed mode setting steps：

（1）Drive enable operation

After the driver is powered on or reset, if the driver is in a disabled state. Write 1 to the motor enable/release register (0x0038) to enable the motor and power on.

（2）Set speed operating parameters

Set parameters such as starting speed (0x0030), acceleration time (0x0031), deceleration time (0x0032), maximum speed (0x0033), etc.

（3）start /stop

After setting the above speed operation parameters, write 0x0001 to the control word register (0x0036) to start, write 0x0010 to decelerate and stop, and write x00020 for emergency stop.

Speed mode programming example (assuming the slave station number is 1)

-----Power on the motor-----

01 06 00 38 00 01 C9 C7 “The motor is enabled and powered”

-----Set speed control parameters-----

Drive enable operation01 06 00 31 00 C8 D9 93 “Set the acceleration time to200ms”

01 06 00 32 00 C8 29 93 “Set the deceleration time to200ms”

01 06 00 33 00 78 79 E7 “Set the maximum speed to120rpm”

It is also possible to write speed control parameters in one go01 10 00 30 00 04 08 00 1E 00 C8 00 C8 00 78 68 37

-----Start speed control-----

01 06 00 36 00 08 68 02

-----Deceleration to stop-----

01 06 00 36 00 20 68 1C

-----emergency stop-----

01 06 00 36 00 40 68 34

6.3Return to zero mode

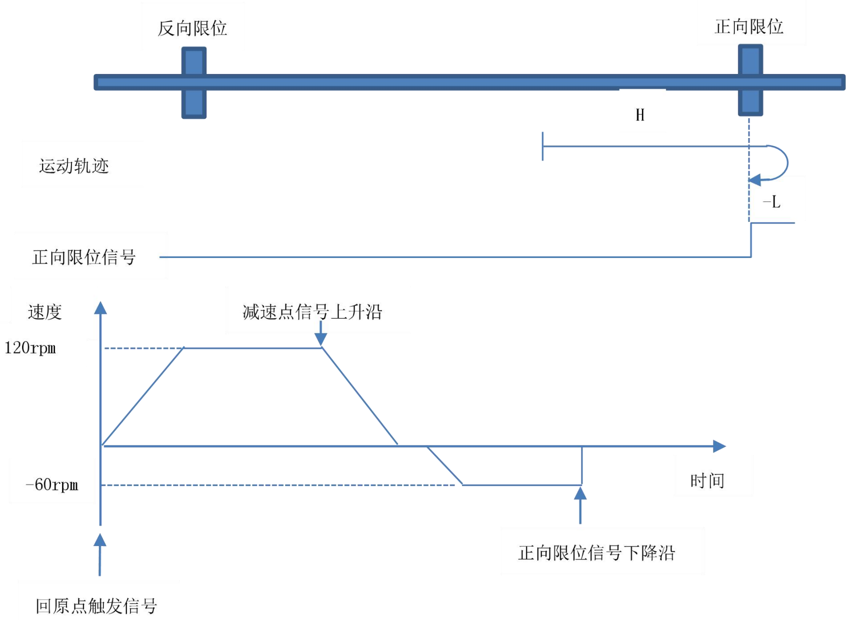
Currently, it supports 5 zero return modes, and the limit signal or origin signal needs to be used during the zero return process.

1） Zero return mode selection register (0x0040) = 18, find the origin in the positive direction (positive limit mode).

◆Origin: forward limit falling edge

◆Deceleration point: forward limit switch

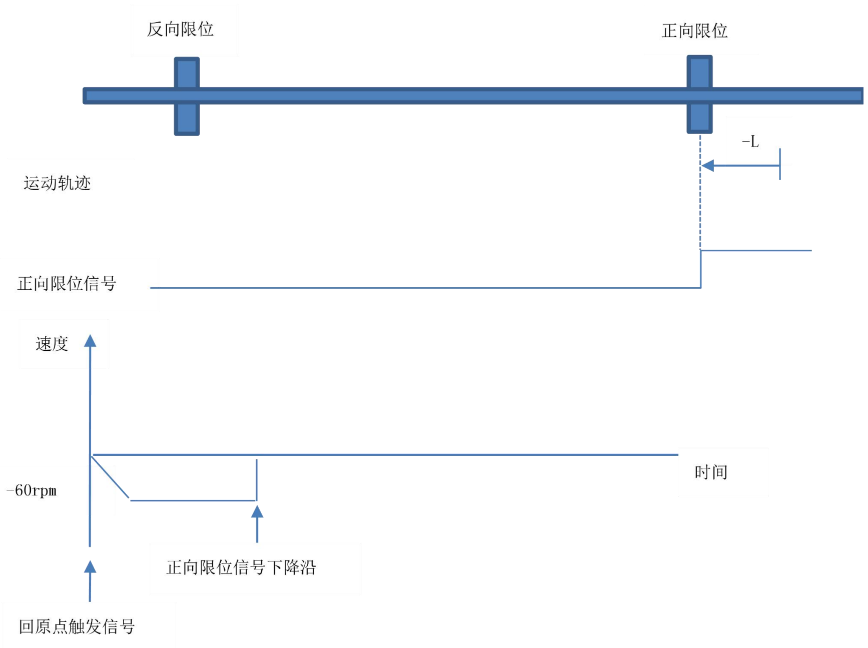
（1）The deceleration signal is invalid when starting to return to the origin.



0x0040=18 and the deceleration point signal is invalid

When starting to return to zero, P-OT=0, return to zero in the forward direction at high speed, after encountering the rising edge of P-OT, decelerate → reverse direction → run at low speed in the reverse direction, and stop after encountering the falling edge of P-OT.

（2）The deceleration point signal is valid when starting to return to zero.



0x0040=18 and the deceleration point signal is valid

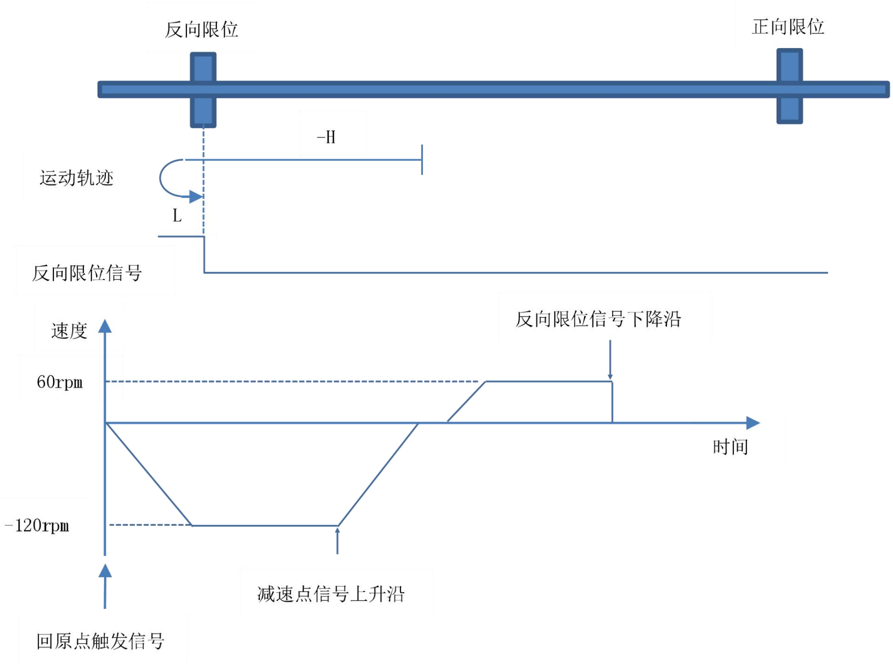
When starting the zero return, P-OT=1, it will directly return to zero in reverse direction at low speed, and stop when encountering the falling edge of P-OT.

2） Zero return mode selection register (0x0040) = 17, search for the origin in the opposite direction (negative limit mode).

◆Origin: Reverse limit falling edge

◆Deceleration point: reverse limit switch

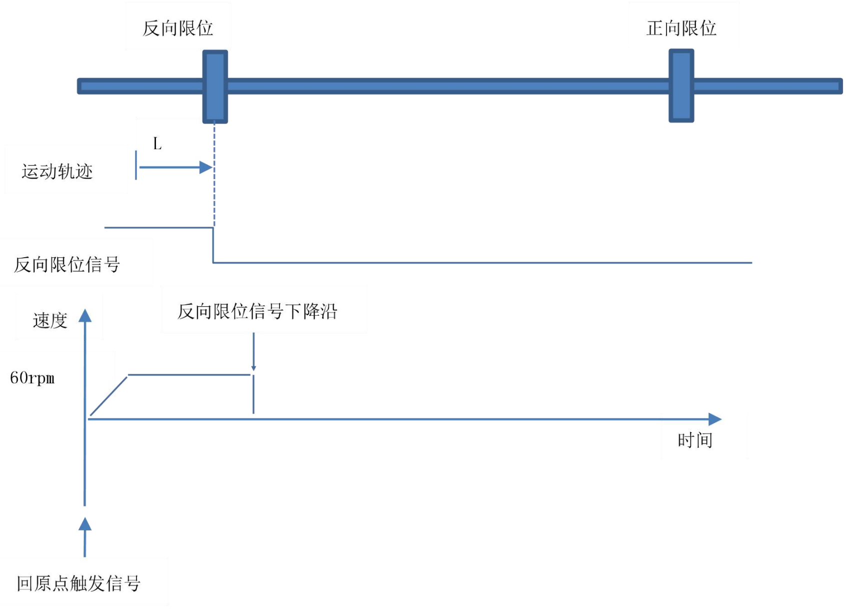
（1） The deceleration signal is invalid when starting to return to zero.



0x0040=17 and the deceleration point signal is invalid

When starting to return to zero, N-OT=0, start returning to zero at high speed in the reverse direction. After encountering the rising edge of N-OT, it will decelerate → reverse → run in forward low speed, and stop after encountering the falling edge of N-OT.

（2）The deceleration signal is valid when starting to return to zero.



0x0040=17 and the deceleration point signal is valid

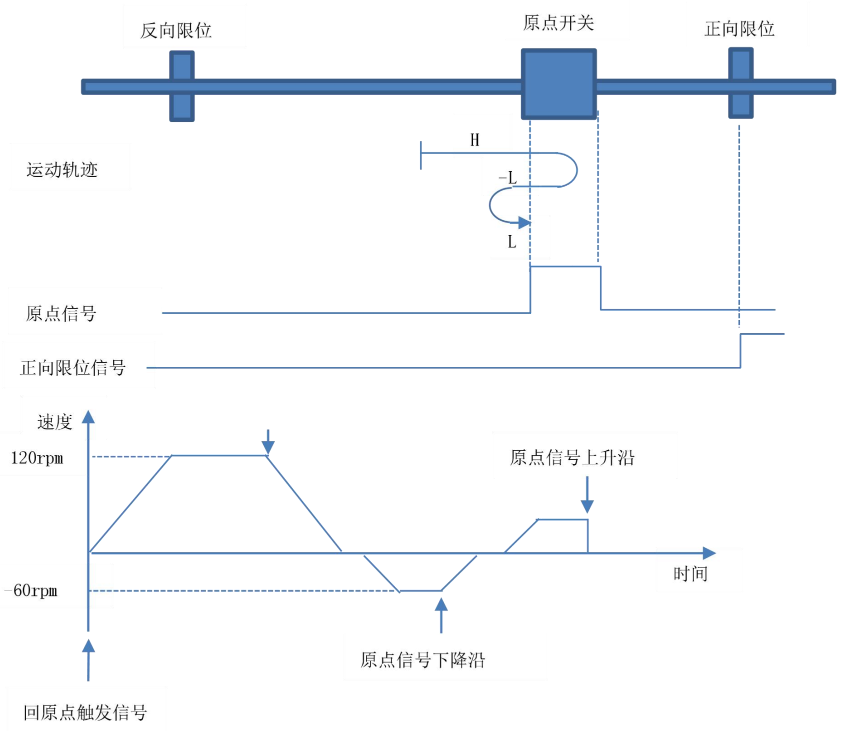
When starting the zero return, N-OT=1, it starts returning to zero directly in the forward direction at low speed, and stops when encountering the falling edge of N-OT.

3）Zero return mode selection register (0x0040) = 24, find the origin in the positive direction (positive limit + origin mode), and can be used with the positive limit switch.

◆Origin: Origin switch rising edge

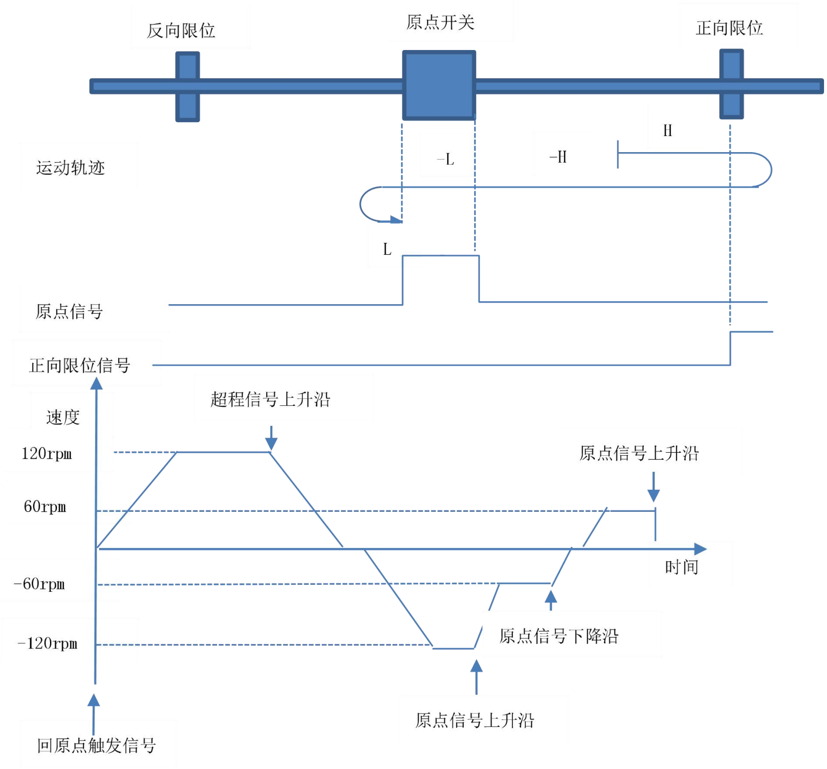
◆Deceleration point: origin switch

（1）The deceleration signal is invalid when returning to zero, and the forward limit switch is not encountered.



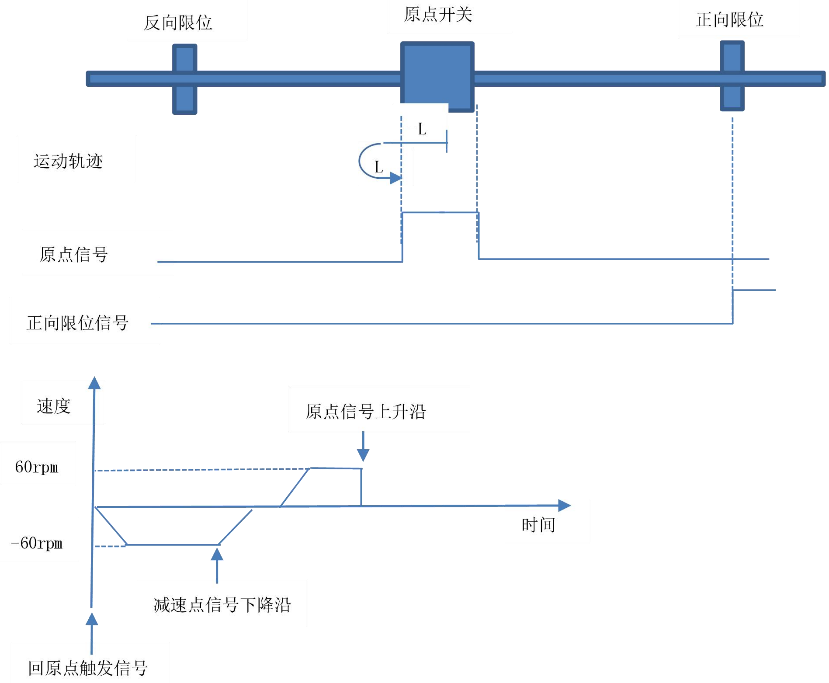
0x0040=24 and the deceleration point is invalid, the forward limit switch is not encountered When starting to return to zero, HW=0, and start returning to the origin at high speed in the forward direction. If the forward limit switch is not encountered, after encountering the rising edge of HW, it will decelerate → reverse → run at low speed in the reverse direction. When encountering the falling edge of HW, reverse operation will occur. In the forward direction, it runs at low speed and stops when it encounters the rising edge of HW.

（2）When returning to zero, the deceleration signal is invalid and the forward limit switch is encountered.



0x0040=24 and the deceleration point is invalid and the positive limit switch is encountered When starting to return to zero, HW=0, start returning to the origin at forward speed. If it encounters the limit switch, it will automatically reverse direction and run at high speed in the reverse direction. After encountering the rising edge of HW, it will decelerate and run at low speed. When encountering the falling edge of HW, it will reverse. To → Forward at low speed, stop after encountering the rising edge of HW

（3）The deceleration signal is valid when returning to zero



0x0040=24 and the deceleration point is valid

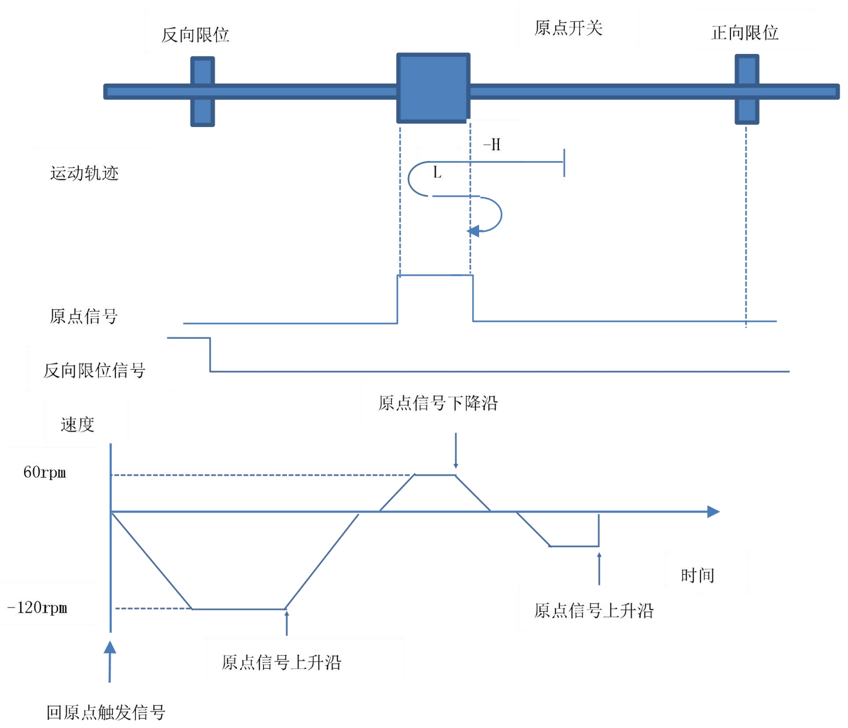
When HW=1 when starting the zero return, it will directly start to return to zero in the reverse direction at low speed. When it encounters the falling edge of HW → reverse → forward at low speed, it will stop when it encounters the rising edge of HW.

4)Zero return mode selection register (0x0040) = 29, search for the origin in the opposite direction (negative limit + origin mode), and can be used with the negative limit switch.

◆Origin: Origin switch rising edge

◆Deceleration point: origin switch

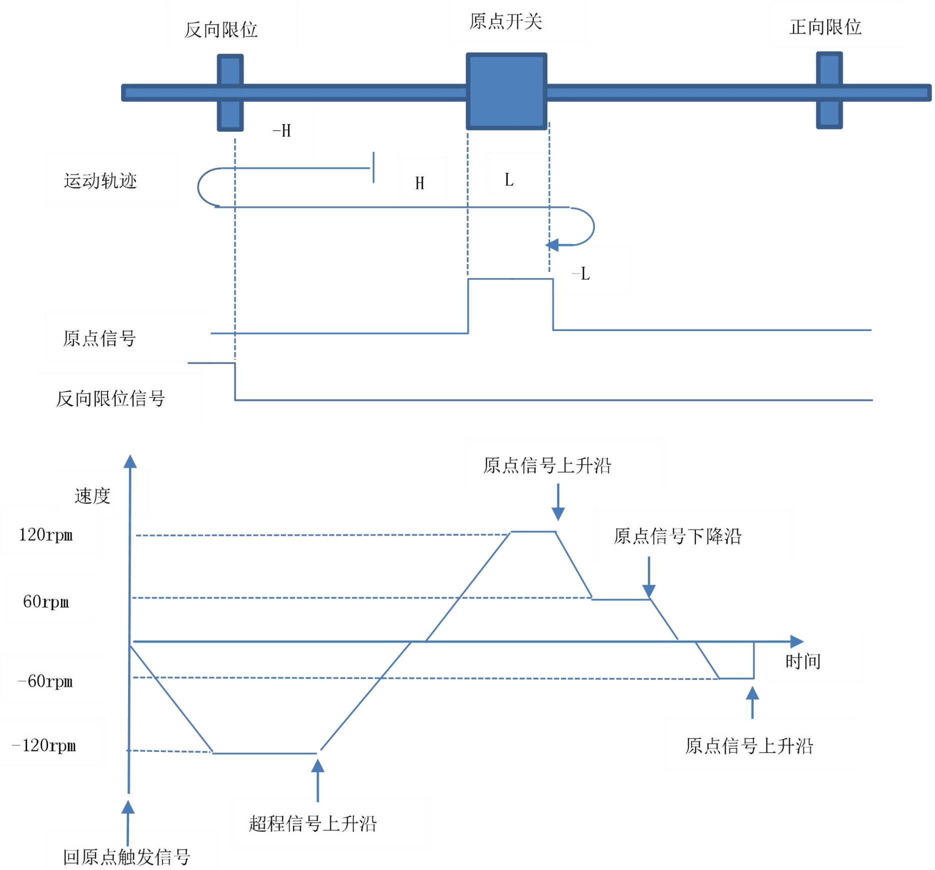
（1）The deceleration signal is invalid when returning to zero, and the reverse limit switch is not encountered.



0x0040=29 and the deceleration point is invalid, the forward limit switch is not encountered

When starting to return to zero, HW=0, start returning to zero at high speed in the reverse direction. If the reverse limit switch is not encountered, and the rising edge of HW is encountered, deceleration → reverse → forward low speed operation, and the falling edge of HW is encountered, reverse , runs in reverse at low speed, and stops when encountering the rising edge of HW.

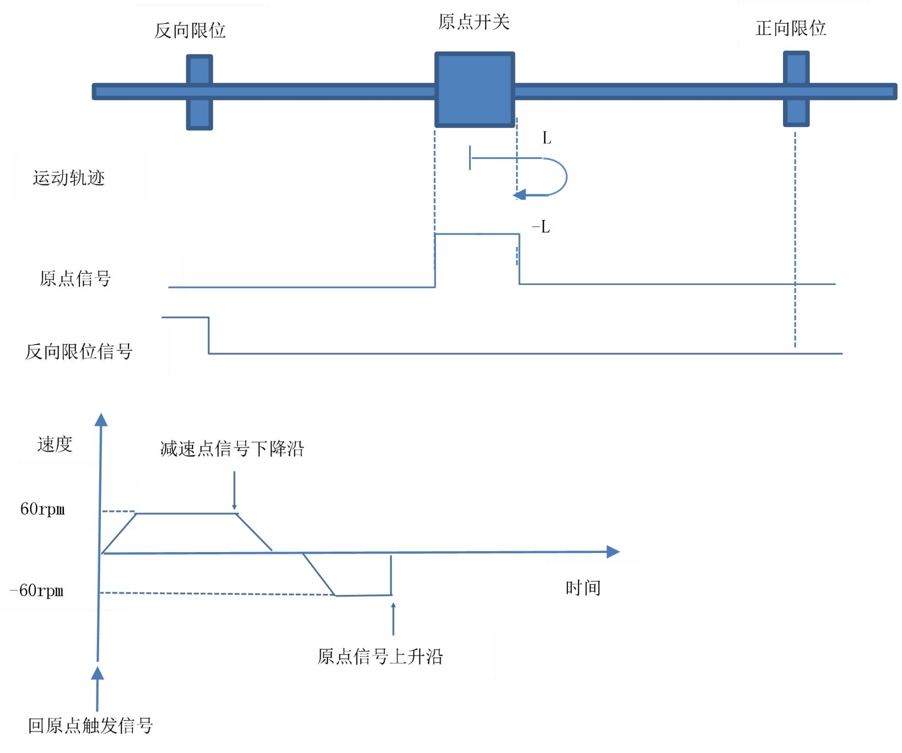
（2）The deceleration signal is invalid when returning to zero, and the reverse limit switch is encountered.



0x0040=29 and the deceleration point is invalid, encountering the reverse limit switch

When starting to return to zero, HW=0, start returning to zero at high speed in the reverse direction. If it encounters the limit switch, it will automatically reverse direction, run at high speed in the forward direction, decelerate when encountering the rising edge of HW, and run at low speed in the forward direction, and run at low speed in the forward direction when encountering the falling edge of HW. →Reverse→Reverse low speed, stop after encountering the rising edge of HW

（3）The deceleration signal is valid when returning to zero



0x0040=29 and the deceleration point is valid

When HW=1 when starting the zero return, it will directly start returning to zero in the forward direction at low speed. When it encounters the falling edge of HW → reverse direction → reverse low speed, it will stop when it encounters the rising edge of HW.

Zero return setting steps ：

（1）Drive enable operation

After the driver is powered on or reset, if the driver is in a disabled state. Write 1 to the motor enable/release register (0x0038) to enable the motor and power on.

（2）Set zero return mode

Set the return-to-origin mode and write 18 to the driver’ s working mode register (0x0040)

（3）Set zero return operation parameters

Set the zero return approach speed (0x0041), query the zero return speed (0x0042), zero return acceleration and deceleration time (0x0043) and other parameters.

（4）Start returning to zero

Write 0x0010 to the control word (0x0036) to start, write 0x0020 to decelerate and stop, and write x0040 for emergency stop.

Zero return programming example (assuming the slave station number is 1)

------Motor power on-----

01 06 00 38 00 01 C9 C7 “The motor is enabled and powered”

-----Set zero return operation parameters------

01 06 00 40 00 12 08 13 “Set the zero return mode selection register to 18”

01 06 00 41 00 78 D9 FC “Set the zero return approach speed to 120rpm”

01 06 00 42 00 1E A9 D6 “Set the speed query zero point speed to 30rpm”

01 06 00 43 00 64 79 F5 “Set the speed query zero point speed to 30rpm”

You can also write the return-to-origin parameters at once

```
01 10 00 40 00 04 08 00 12 00 78 00 1E 00 64 53 75
```

-----Start zero return-----

```
01 06 00 36 00 10 68 08
```

-----Deceleration to stop-----

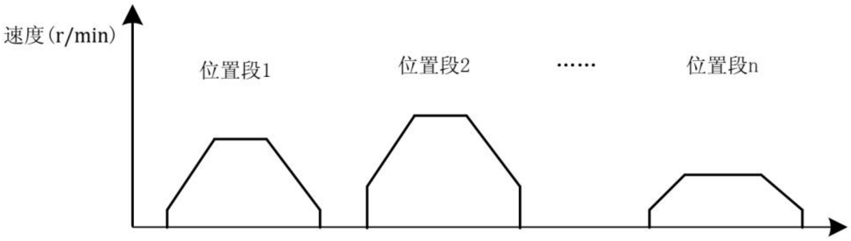
```
01 06 00 36 00 20 68 1C
```

-----emergency stop-----

```
01 06 00 36 00 40 68 34
```

6.4 Multi-segment position mode

The multi-segment position mode function is a working method that combines multiple position segments in a certain order, triggers movement through DI signals, and completes a series of position segment actions. This function can also be regarded as a multi-segment combination of the position mode described in 6.1 Position Mode. The difference is that the user can store the description parameters of several position segments such as acceleration and deceleration time, total number of pulses, etc. in EEPROM in advance. These positions need to be enabled. Simply provide a trigger signal to complete the work. Its working process is described as shown in the figure below.



多段位置工作模式

Multi-segment positions need to be selected and triggered through the DI port to run. An example is as follows:

Register address	Settings	illustrate
0x0015	11	DI2 configured to trigger on location path
0x0016	13	DI3 is configured as path selection switch 0
0x0017	14	DI4 is configured as path selection switch 1
0x0018	15	DI5 is configured as path selection switch 2
0x0019	16	DI6 is configured as path selection switch 3

After configuring the DI port according to the above table, select the position segment through DI3~DI6, and then use DI2 to trigger (rising edge) the position segment operation. A corresponding table describing the position segment parameters is as follows:

parameter name	illustrate
High bit of total pulse number (0x0060~0x006F)	Pulse number high bit
Low bit of total pulse number (0x0070~0x007F)	Low pulse number
Running speed (0x0080~0x008F)	Position segment running speed
Acceleration time (0x0090~0x009F)	Position segment running acceleration time
Deceleration time	Position segment operation deceleration

(0x00A0~0x00AF)	time

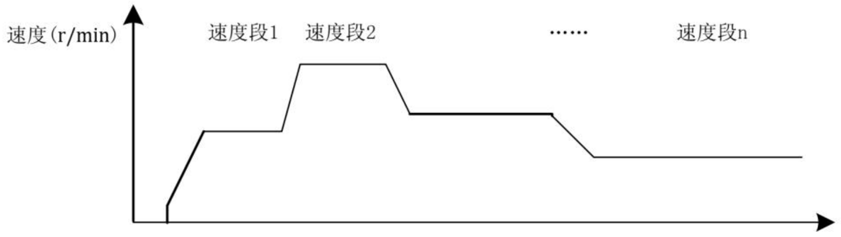
Multi-segment position mode settings

Register address	definition	default value	illustrate
0x0039	Multi-segment position trigger mode	0	0: Ignore new instructions when positioning is running 1: Interrupt the current positioning operation to execute new instructions

0x003A	Multi-segment position absolute position/relative position mode setting	0	0: Relative position mode;
			1: Absolute position mode;

6.5 Multi-speed mode

The multi-segment speed mode function is a working method that stores multiple speed segments in advance, triggers movement through DI signals, and completes a series of different speed actions. Its working process is described as shown in the figure below.



多段速度工作模式

Multi-stage speed needs to be selected and triggered through the DI port to run. An example is shown below.

Register address	Settings	illustrate
0x0015	12	DI2 configured as velocity path trigger
0x0016	13	DI3 is configured as path selection switch 0
0x0017	14	DI4 is configured as path selection switch 1
0x0018	15	DI5 is configured as path selection

		switch 2
0x0019	16	DI6 is configured as path selection switch 3

After configuring DI according to the above table, select the speed segment through DI3~DI6, and then DI2 triggers (on operation, disconnection stop) speed segment operation,

parameter name	illustrate
Running speed (0x00B0~0x00BF)	Speed section running speed
Acceleration time (0x00C0~0x00CF)	Speed segment running acceleration time
Deceleration time (0x00D0~0x00DF)	Speed segment operation deceleration time