MKS SERVO 42D/57D Closed-Loop Stepper Motor

Modbus RTU User Manual V1.0.9

Note: This manual corresponds to firmware version V 1.0.9 .

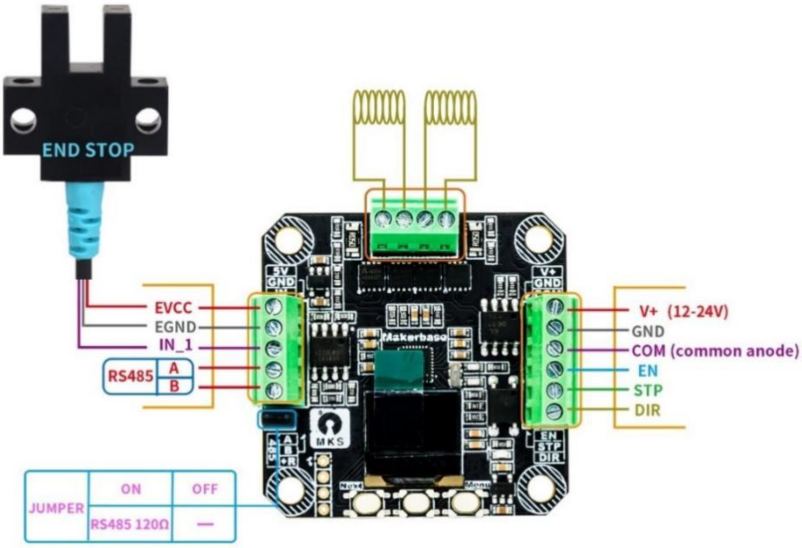
MKS SERVO42D/57D_RS485 Version Notes			
manual	content	Firmware	date
V1.0.9	1. The RTU instruction manual is now presented in a separate section, with a redesigned layout, categorized by function, and includes additional examples.	V1.0.9	Nov-2025
	2. Increase the heart rate protection time, see 3.2.14.		
	3. Added coordinate zeroing function, see 5.2 , 5.7 .		

Catalog

Part 1. Port and wiring instructions

1.1 Interface Description

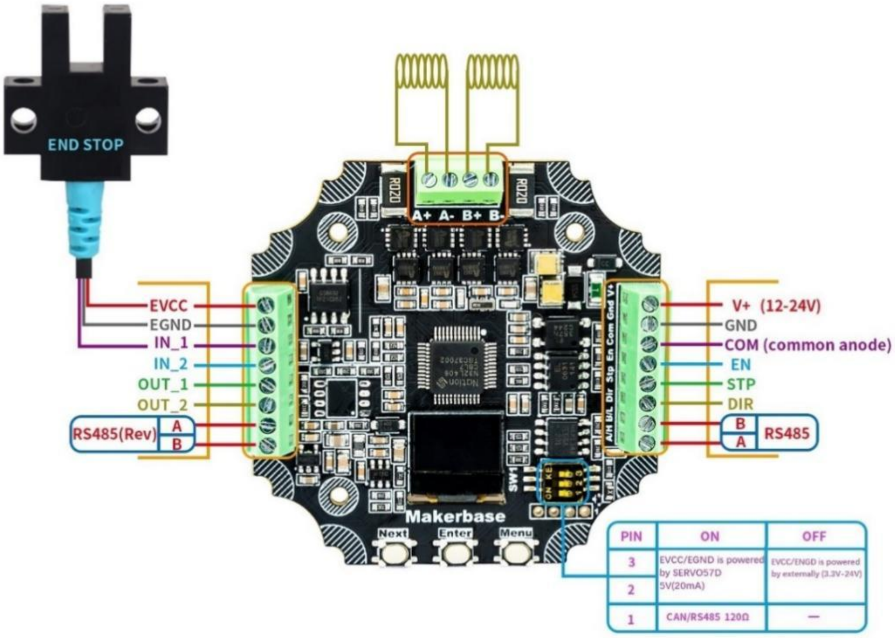
① S ERVO42D RS485 Interface Description



JUMPER	ON	OFF
	RS485 120Ω	—

Note: EVCC/ENGD is powered by a 5.0V/20mA power supply from the SERVO42D driver board .

② SERVO57D_RS485 Interface Description



PIN	ON	OFF
3	EVCC/EGND is powered by SERVO57D 5V(20mA)	EVCC/ENGD is powered by externally (3.3V-24V)
2		
1	CAN/RS485 120Ω	—

1.2 IO Port Description

Default port	Function	57D	28/35/42D
IN_1	home or left-limit	√	√
IN_2	right-limit	√	X
OUT_1	stall indication: 0-protected; 1-unprotected	√	X
OUT_2	Pulse frequency division output	√	X

Note: After the limit remapping function is enabled by the 9E instruction, the IN_1 and IN_2 input signals are invalid.

The port is defined as follows after remapping:

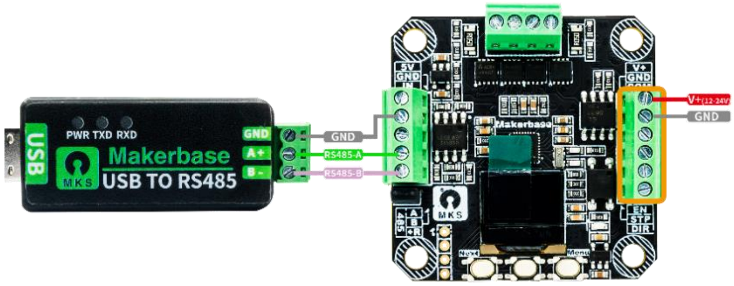
Remapped port	Function	57D	28/35/42D
En	home or left-limit	√	√
Dir	right-limit	√	√
Com	High level required	√	√

1.3 RS485 Wiring Method

Note: To reduce bus interference, the host computer and the motor should share a common ground, and RS485 A and B signals should be transmitted using shielded twisted-pair cable.

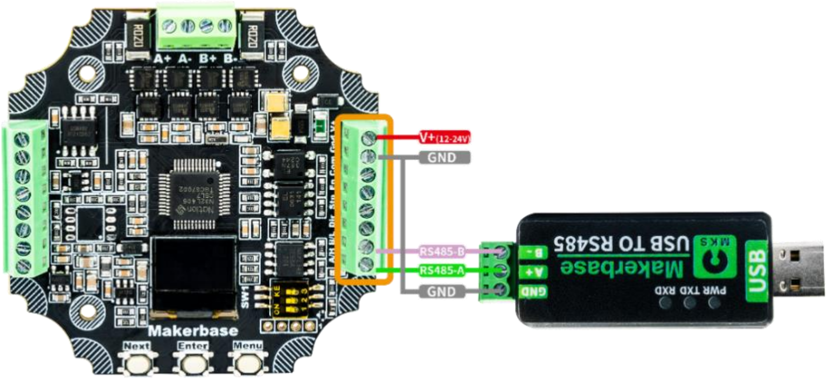
1. RS485 standalone communication wiring

① MKS SERVO42D RS485 standalone wiring



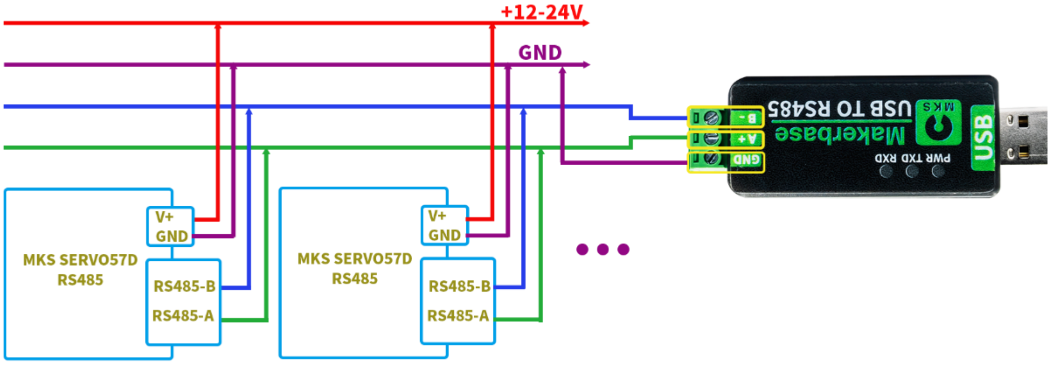
Note: A 120Ω terminating resistor is not required for stand-alone communication (i.e., do not connect a shorting cap).

② MKS SERVO57D RS485 standalone wiring



Note: A 120Ω terminating resistor is not required for standalone communication.

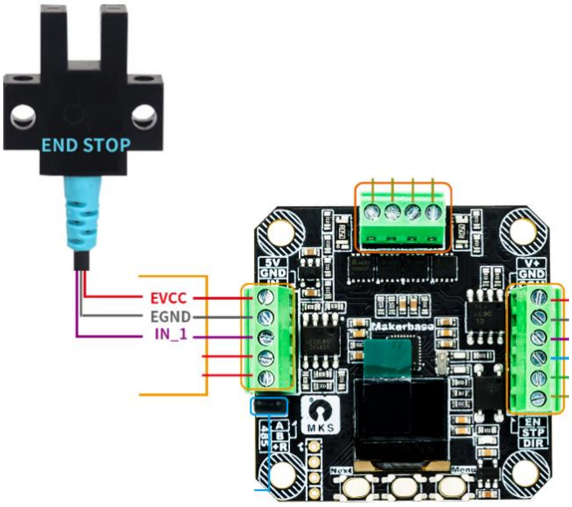
2. RS485 multi-machine communication wiring



1.4 External switch wiring method

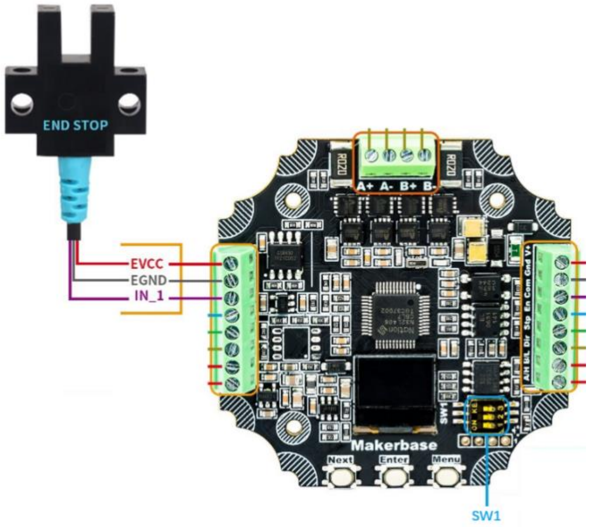
Note: NPN type limit switches are used by default.

① **MKS SERVO42D limit switch wiring**



Note: EVCC/ENGD is powered by the SERVO42D driver board at 5.0V/20mA.

② **MKS SERVO57D limit switch wiring**



SW1		
PIN	ON	OFF
3	EVCC/EGND are powered by SERVO57D at 5V (20mA).	EVCC/EGND is powered by an external source (3.3V-24V).
2		
1	RS485 120Ω terminating resistor	NULL

The mechanical switch only needs to be connected to two signal lines: “EGND” and “IN_1”. Pin 2 of the DIP switch must be in the ON state.

The DIP switches are in the OFF state by default. If the limit switches require internal power supply, the operation should be performed according to the table above.

Part 2. Communication Protocol Description

2.1 RS485 bus communication

The closed-loop motor incorporates the standard Modbus RTU communication protocol, supporting the reading and writing of single and multiple parameters by the Modbus RTU master station. Once a controller with Modbus protocol is successfully connected to the closed-loop motor, the controller can directly perform parameter settings and read operations on the motor. In bus control mode, the controller can modify operating command parameters such as position and speed to control motor operation.

Please refer to the relevant documents for the Modbus-RTU/RS485 protocol and standards.

Note: The RTU register addresses of the MKS D series closed-loop stepper motors are not continuous and do not support continuous data reading and writing. For standard Modbus RTU communication protocol, please select the MKS ES series closed-loop stepper motor.

2.2 Communication parameters

The default baud rate is 38400, with 8 data bits, none parity bits, and 1 stop bit.

2.3 Slave address

The default address is 01, and the value can be set from 00 to 255.

Note: 00 is the broadcast address. When using the broadcast address, the slave device will not respond.

2.4 Function code

The Modbus RTU protocol has a variety of bus commands. Closed-loop motors support the three most commonly used function code commands (04H/06H/10H). These three function codes can meet all the controller's control needs for the closed-loop motor.

The Modbus register has a 16-bit data length. When using Modbus commands, pay attention to the data type of the access parameters .

The parameter data type is UINT16. INT16 should be read using function code 04H and written using function code 06H.

For parameters of data type UINT32 or INT32, or for reading or writing multiple parameters consecutively, use function codes 03H for reading and 10H for writing.

Note: If the configuration fails,

Function code 06H, response frame register data is 0xFFFF

Function code 10H, response frame register count is 0

Part 3. General Instructions

3.1 Read parameter command

3.1.1 Reading the value of a multi-turn encoder

Note: Encoder single-turn value range is 0~0x4000

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	30H	00H	03H	B0H	04H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	Carry-over value (number of rotations)	Encoder value	CRC16	
			carry	value	high byte	low byte
01H	04H	06H	int32_t	uint16_t		

carry: the carry vaule of the encoder.

value: the current vaule of the encoder. (range 0~0x3FFF)

When value is greater than 0x3FFF, carry +=1.

When Value is less than 0, carry -=1.

For example:

If the current carry|value is 0x3FF0, After one turn CCW, the carry|value (+0x4000) is 0x13FF0.

If the current carry|value is 0x3FF0, After one turn CW, the carry|value (-0x4000) is 0xFFFFFFFF3FF0.

Note: The encoder value is updated regardless of whether the motor is enabled or not.

3.1.2 Read the cumulative multi-turn encoder value

Note: Encoder single-turn value range is 0~0x4000

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	31H	00H	03H	E1H	C4H

Uplink message (host computer ← driver board)					
Slave address	Function code	byte count	Encoder value	CRC16	
				high byte	low byte
01H	04H	06H	Value (int48_t)		

After one turn clockwise, the value += 0x4000;

After one turn CCW, the value -= 0x4000;

For example:

If the current value is 0x3FF0, After one turn CCW, the value(+0x4000) is 0x7FF0.

If the current value is 0x3FF0, After one turn CW, the value(-0x4000) is 0xFFFFFFFFFFF0.

3.1.3 Read the real-time speed of the motor

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	32H	00H	01H	90H	05H

Uplink message (host computer ← driver board)					
Slave address	Function code	byte count	Real-time rotation speed	CRC16	
				high byte	low byte
01H	04H	02H	speed (int16 _t)		

Note: The unit of rotational speed is RPM . The rotational speed is greater than 0 when rotating forward and less than 0 when rotating in reverse.

3.1.4 Read the number of received pulses

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	33H	00H	02H	81H	C4H

Uplink message (host computer ← driver board)					
Slave address	Function code	byte count	pulse count	CRC16	
				high byte	low byte
01H	04H	04H	pulses(uint32 _t)		

3.1.5 Read position angle error

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	39H	00H	02H	A1H	C6H

Uplink message (host computer ← driver board)					
Slave address	Function code	byte count	error	CRC16	
				high byte	low byte
01H	04H	04H	error (int 32_t)		

The difference between the controlled position angle and the real-time angle position of the motor is expressed in units of 0~51200, which represents 0~360° . For example, when the error is 1° , the value is 51200/360° = 142.22, and so on.

3.1.6 Read motor enable status

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	3AH	00H	01H	11H	C7H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	reserve	Enable state	CRC16	
					high byte	low byte
01H	04H	02H	00H	enable(uint8_t)		

enable = 1	Motor is enabled.
enable = 0	Motor is not enabled.

3.1.7 Read the motor return to zero status

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	3BH	00H	01H	40H	07H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	Single-cycle to zeroin	returns to zero	CRC16	
					high byte	low byte
01H	04H	02H	status 1	status 2		

status x = 0	It is returning to zero.
status x = 1	returned to zero successfully.
status x = 2	Not returned to zero or failed to return to zero.

Note: status1 indicates single-turn zero return state;
Status2 indicates non-single-turn zero return state;

3.1.8 Read version information

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	40H	00H	02H	70H	1FH

Uplink message (host computer ← driver board)					
Slave address	Function code	byte count	Version information	CRC16	
				high byte	low byte
01H	04H	04H	version(uint32_t)		

3.2 Configuration parameter commands

3.2.1 Calibrate the motor

The corresponding "Cal" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	80H	00H	01H	49H	E2H

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	80H	00H	01H	49H	E2H

Before calibrating the encoder, please ensure that the motor is not under load! It is recommended to calibrate it before installing it into the machine.

Note: After calibration, the driver board will automatically reset and restart.

3.2.2 Set the work mode

Corresponding to the "Mode" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	82H	00H	mode		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	82H	00H	mode		

mode = 00 CR_OPEN (Pulse interface open-loop mode)
mode = 01 CR_CLOSE (Pulse interface closed-loop mode)
mode = 02 CR_vFOC (Pulse Interface FOC Mode)
mode = 03 SR_OPEN (Serial interface open-loop mode)
mode = 04 SR_CLOSE (Serial interface closed-loop mode)
mode = 05 SR_vFOC (Serial interface FOC mode)

3.2.3 Set the working current

The corresponding "Ma" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	83H	Current			

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	83H	Current			

SERVO42D/28D/35D maximum operating current 3000mA

SERVO57D maximum operating current of 5200mA.

Note: If the highest bit of Current is 1, only the current value will be changed, and it will not be saved. This is suitable for changing the current value during operation.

3.2.4 Set holding current percentage

The corresponding "HoldMa" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	9BH	00H	holdMa		

holdMa = 00 10%

holdMa = 01 20%

…

holdMa = 08 90%

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	9BH	00H	holdMa		

3.2.5 Set subdivisions

The corresponding "MStep" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	84H	00H	micstep		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	84H	00H	micstep		

3.2.6 Set the active level of the En pin

The corresponding "En" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	85H	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	85H	00H	enable		

enable = 00 corresponds to low-level enable (L).

enable = 01 corresponds to a high-level enable (H) signal.

enable = 02 corresponds to always being enabled (Hold).

3.2.7 Set the motor rotation direction

The corresponding "Dir" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	86H	00H	dir		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	86H	00H	dir		

dir = 00 corresponds to clockwise rotation

dir = 01 corresponds to counter-clockwise rotation

Note: This is only valid for pulse interfaces; the direction of the serial interface is determined by the command.

3.2.8 Set automatic screen-off function

The corresponding "AutoSDD" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	87H	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	87H	00H	enable		

enable = 00 disables the automatic screen-off function.

enable = 01 Enables automatic screen-off function

3.2.9 Set subdivision interpolation function

The corresponding "MPlyer" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	89H	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	89H	00H	enable		

enable = 00 disables the internal 256 subdivision interpolation function.

enable = 01 enables the internal 256 subdivision interpolation function.

3.2.10 Set the serial port baud rate

The corresponding "UartBaud" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8AH	00H	baud		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8AH	00H	baud		

baud = 01 9600
baud = 02 19200
baud = 03 25000
baud = 04 38400
baud = 05 57600
baud = 06 115200
baud = 07 256000

3.2.11 Set slave address

The corresponding "UartAddr" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8BH	00H	addr		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8BH	00H	addr		

Note 1 : The address range is 00 ~ 0xFF , where 00 is the broadcast address and 01 is the default address.

Note 2 : Setting an address greater than 16 will also be displayed at the end of the UartAddr options.

3.2.12 Protocol Mode

The corresponding "MB_RTU" option on the screen

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8EH	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8EH	00H	enable		

enable = 00 disables the MODBUS-RTU communication protocol.

enable = 01 Enables the MODBUS-RTU communication protocol

After the MODBUS-RTU communication protocol is disabled, it can be enabled using the following command.

FA 01 8E 01 8A

3.2.13 Set key lock function

Downlink message（host computer → driver board）							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8FH	00H	enable		

Uplink message（host computer ← driver board）							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	8FH	00H	enable		

enable = 00 disables key lock function

enable = 01 Enables button lock function

3.2.14 Set heartbeat protection time

Heartbeat protection refers to the system that， if the motor does not receive any instructions from the host computer within the set protection time， it will control the motor to stop urgently， preventing abnormal accidents caused by communication interruption.

The configuration command is as follows:

Downlink message（host computer → driver board）									
Slave address	Function code	Starting address		Number of registers		byte count	Bytes 4-7 (uint32_t)	CRC16	
		Hi	Lo	Hi	Lo			Hi	Lo
01H	10H	00H	98H	00H	02H	04H	times		

times: Protection time， in milliseconds. （Default: 0， i.e.， heartbeat protection is off）

Note: When times=0， the heartbeat protection function is turned off.

Uplink message（host computer ← driver board）							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	98H	00H	02H	H	H

3.3 PID Parameter Setting Instructions

Note: MKS motors have their PID parameters pre-adjusted at the factory. Unless otherwise required, users are advised against adjusting the PID parameters themselves. If such adjustment is necessary, proceed with extreme caution to avoid damaging the motor!

3.3.1 Set PID parameters for vFOC mode

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	Byte 1-2	Byte 3-4	Byte 5-6	Byte 7-8	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	96H	00H	04H	08H	Kp	Ki	Kd	Kv		

Kp: Range 0-1024 (Default: 0xDC)
Ki: Range 0-1024 (Default: 0x64)
Kd: Range 0-1024 (Default: 0x10E)
Kv: Range 0-1024 (Default: 0x140)

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	96H	00H	04H	21H	E6H

3.3.2 Set CLOSE mode PID parameters

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	Byte 1-2	Byte 3-4	Byte 5-6	Byte 7-8	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	97H	00H	04H	08H	Kp	Ki	Kd	Kv		

Kp: Range 0-1024 (Default: 0xC8)
Ki: Range 0-1024 (Default: 0x50)
Kd: Range 0-1024 (Default: 0xFA)
Kv: Range 0-1024 (Default: 0x12C)

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	97H	00H	04H	70H	26H

3.4 Stall protection instructions

If the motor fails to reach the designated position within the response time due to overload or any other reason, the stall protection will be triggered, the motor will unlock and the shaft will be released to avoid damage.

There are two protection modes:

1. Current overload protection mode

   If motor overcurrent is detected, the stall protection will be activated.

2. Position out-of-tolerance protection mode

   If the motor position error exceeds y within time period x , the protection mechanism will be activated. (x and y are configurable)

These two protection modes can be turned on or off independently. They monitor the motor independently, and the motor protection will be activated when either protection condition is triggered.

Note: When the current overload protection is triggered, the screen will display "Wrong…"

When the position error protection is triggered, the screen displays "Wrong2…".

3.4.1 Set overcurrent protection

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	88H	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	88H	00H	enable		

enable = 00 disables stall protection.

enable = 01 Enables stall protection function

After the stall protection is activated , there are three ways to deactivate it:

1. Press the Enter key to release the stall protection;

The stall protection can be released via command (3D) ;

3. Loosening the motor shaft can release the stall protection .

3.4.2 Set position out-of-tolerance protection parameters

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	Byte 1	Byte2	Byte 3-4	Byte 5-6	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	9DH	00H	03H	06H	0	pEnble	Tim	Errs		

pEnble 0 : Disables position error protection (defaul)
1 : Enable position error protection function
Tim: uint16_t sets the error statistics time length.
Note: 1(Tim) unit is approximately equal to 15 ms.

Errs: uint16_t sets the number of boot protection errors
Note: When Errrors = 28000 , the motor is misaligned by 360 degrees.

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	9DH	00H	03H	11H	E6H

3.4.3 Read motor stall status

When a motor stalls, a stall flag will be set. This command allows you to determine if the motor has stalled. If the stall protection option is enabled, the motor will automatically unlock after a stall occurs.

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	3EH	00H	01H	50H	06H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	reserve	stalled state	CRC16	
					high byte	low byte
01H	04H	02H	00H	status		

status = 0 No blockage
status = 1 Blocked

3.4.4 Release the motor from stall state

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	3DH	00H	01H	D9H	C6H

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	3DH	00H	01H	D9H	C6H

When the motor stalls, sending this command can release the current stall state.
If a stall occurs again after the stall is cleared , the stall protection will still be triggered.
Loosening the motor shaft can also relieve the stall condition .

3.5 Recovery parameters and reset instructions

3.5.1 Factory reset command

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	3FH	00H	01H	78H	06H

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	3FH	00H	01H	78H	06H

Note 1 : After restoring the default parameters, the driver board will automatically restart, and there is no need to recalibrate the motor .

Note 2 : Press and hold the ″Next″ button before powering on . Wait for the LED light to illuminate to restore the default parameters.

3.5.2 Reset and restart the motor

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	41H	00H	01H	18H	1EH

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	41H	00H	01H	18H	1EH

3.6 IAP Firmware Online Upgrade Instructions

There are two IAP upgrade modes:

IAP Mode 1:

Press and hold the "Menu" button, then power on the device to enter IAP mode 1. the user configuration parameters will be retained after firmware upgrade.

IAP Mode 2:

Send a control command to enter IAP mode 2, the user configuration parameters will be retained after firmware upgrade..

Instructions for IAP upgrades can be found in "MKS SERVO42&57D IAP Upgrade Instructions.pdf".

The IAP upgrade operation video can be found in "MKS SERVO42&57D IAP Upgrade Operation Video.mp4".

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	50H	00H	cmd	48H	1BH

cmd = 01 Enter boot mode

cmd = 02 Enter silent state

cmd = 03 Exit silent state

Control word cmd description:

When there is only one motor on the bus, simply set cmd = 01 to enter boot mode;

When there are multiple motors on the bus, the following steps can be taken to avoid data interference:

a. First, send the command cmd = 02 to the other motors that are not being upgraded to enter silent mode;
b. Then send the command cmd = 01 to the motor to be upgraded to enter boot mode and upgrade the firmware;
c. After the upgrade is complete, send the command cmd = 03 to other motors to exit the silent state.

Note: In silent mode, the motor does not respond to commands other than 50H.

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	50H	00H	01H	48H	1BH

Note: After successful boot, the motor will automatically restart and enter IAP mode 2, waiting to receive the upgrade file.

Part 4. IO Port Operation Instructions

4.1 Read I/O port status

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	34H	00H	01H	70H	04H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	reserve	Port status	CRC16	
					high byte	low byte
01H	04H	02H	00H	status		

status							
Bit 7	…	…	bit4	bit3	bit2	bit1	bit0
Undefined				OUT_2	OUT_1	IN_2	IN_1

Note: After the limit remapping function is enabled, bit0 corresponds to the En state and bit1 corresponds to the Dir state .

4.2 Write IO port data

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	Register 1		Register 2		CRC16	
		Hi	Lo	Hi	Lo		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	36H	00H	02H	04H	OUT2_mask	OUT2	OUT1_mask	OUT1		

OUT2_mask 0: Do not write to OUT_2 IO port(default)
1: Write OUT_2 value to OUT_2 IO port
2: OUT_2 IO port value remains unchanged

OUT1_mask 0: Do not write to OUT_1 IO port (default)
1: Write OUT_1 value to OUT_1 IO port
2: OUT_1 IO port value remains unchanged

OUT_2 OUT_2 port write value (0/1)

OUT_1 OUT_1 port write value (0/1)

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high	low	high	low	high	low
01H	10H	00H	36H	00H	02H	A1H	C6H

4.3 Pulse frequency division output

Map the OUT_2 port to a pulse divider output port (only for 57D).

Downlink message (host computer → driver board)										
Slave address	Function code	Starting address		Number of registers		byte count	Byte 1-2	Bytes 3-6	CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo
01H	10H	00H	99H	00H	03H	06H	divLevel	divPeriod		

divLevel 0: Low start level; 1: High start level (default 0)

divPeriod Frequency division period (default 0)

When divPeriod < 100, no frequency divider output

When divPeriod >= 100, the PEND port toggles once every divPeriod pulse cycle.

For example, setting 16 subdivisions, divPeriod = 3200, then the PEND port flips once for every revolution of the motor.

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	99H	00H	03H	50H	27H

Note: To disable this feature, simply set divPeriod = 0.

Part 5. Motor return to zero instructions

There are two types of motor homing methods: "origin homing" and "coordinate homing".

5.1 Explanation of the method of returning to zero at the origin

There are two ways to "return to zero": "return to zero by origin switch" and "return to zero by mechanical limit switch".

1. Return the origin switch to zero.

Connect the origin switch to the corresponding port:

Origin switch access port	
Default port	Remapped port
IN1	EN

For details, please refer to: 1.2 IO Port DescriptionDescription

The process of returning to zero is as follows:

a. The motor searches for the switch based on the set "direction" and "speed " ;

b. Stop immediately upon encountering the rising edge of the switch signal;

c. Mark the current position as "coordinate zero point", and the origin has successfully returned to zero.

Note: If the motor starts in the closed position of the switch, it will first run in reverse to disengage from the switch.

2. Mechanical limit switch returns to zero.

The zero-return torque when the mechanical limit is returned to zero needs to be set in advance via command. The set torque should be sufficient to drive the load, but should not be too large to avoid damaging the equipment.

The process of returning to zero is as follows:

a. searches for the mechanical limit position using the set "speed " , "direction", and "torque " ;

b. When encountering a mechanical limit switch, the rotor will stop and then run to the preset "origin offset" position to stop.

c. Mark the current position as "coordinate zero point", and the origin has successfully returned to zero.

3. Single lap to zero

The "zero point" of the coordinates within a single circle needs to be set in advance via instructions.

The direction of the single-turn zeroing needs to be set in advance via command: "forward", "reverse", or "nearest".

The process of returning to zero is as follows:

a. The motor returns to the "coordinate zero point" position within a preset single revolution in the set zero-return direction and speed;

Upon arrival, the current position is reset to zero, and the single-lap coordinates have successfully returned to zero.

5.2 Explanation of coordinate zeroing method

To return to zero directly, you need to first execute the "Return to Zero" function to determine the "zero point".

The process of returning to zero is as follows:

No search process is required; it directly and quickly reaches the "coordinate zero point" position and successfully returns to zero.

5.3 Set parameters related to zero return.

5.3.1 Configure port mapping

(Bus control mode only)

28 / 35 / 42D motor, since it only has a left limit port, in bus control mode, the limit port remapping can be enabled to add a right limit port.

57D motors, the limit port can be remapped if wiring is convenient.

After enabling port forwarding:

Left limit switch -> En port

Right limit switch -> Dir port

The COM port must be connected to the corresponding high level.

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	9EH	00H	enable		

enable = 00 disables remapping

enable = 01 Enable remapping

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	9EH	00H	enable		

5.3.2 Set parameters such as zero-return direction and speed.

(Corresponding to the "**HmTrig**, **HmDir**, **HmSpeed**, **EndLimit**" options on the screen)

Downlink message (host computer → driver board)													
Slave address	Function code	Starting address		Number of registers		byte count	Level	dir	speed		Limit Enable	CRC16	
		Hi	Lo	Hi	Lo		Hi	Lo	Hi	Lo		Hi	Lo
01H	10H	00H	90H	00H	03H	05H	hmTrig	hmDir	HmSpeed		EndLimit		

HmTrig the effective level of the end stop

0: Low 1: High

HmDir the direction of go home

0: CW 1: CCW

HmSpeed the speed of go home

0~3000 (RPM)

EndLimit

0: disable endstop-limit

1: enable endstop-limit

Note 1: After using the limit function for the first time or changing the limit parameters, you need to perform a limit reset once.

(Menu -> GoHome or serial command "91")

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	90H	00H	03H	80H	25H

5.3.3 Set the zero-return torque and origin offset parameters.

Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	offset (uint32_t)	Zero mode (uint16_t)	current (uint16_t)	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	94H	00H	04H	08H	retValue	hm_mode	hm_ma		

hm_mode 0: Return to zero by endstop
1: Return to zero by mechanical limit
2: Single lap to zero

hm_ma Mechanical limit return-to-zero current

retValue Origin offset

For example:

retValue = 0x4000 (Returns one full rotation, 360 degrees)

retValue = 0x2000 (Returns half a circle, 180 degrees) (Default)

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	94H	00H	04H	80H	26H

5.3.4 Set single-cycle zero-return parameters

(Corresponding to the screen options ″ 0_Mode″, ″Set 0″, ″ 0_Speed″, ″ 0_Dir ″)

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	mode	Set 0 points	speed	direction	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	9AH	00H	02H	04H	mode	ena	speed	dir		

mode:

0: Disable do not go back to zero

1: DirMode go back to zero with direction

2: NearMode go back to zero with minimum angle

enable:

0: clean zero

1: set zero

2: not modify the zero

speed:

0 ~ 4 (0:slowest 4:fastest)

dir:

0: CW

1: CCW

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	9AH	00H	02H	61H	E7H

5.3.5 Set ″Zero Point″ command

This command directly sets the current position to ″zero″.

Inquire about the RTU frame (controller → closed-loop motor).							
Slave address	Function code	Register start address		Data to be written to register		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	92H	00H	01H	E9H	E7H

After successful writing, the response frame is as follows:

Uplink message (Controller ← Closed-loop motor)							
Slave address	Function code	Register start address		Data already written to registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	92H	00H	01H	E9H	E7H

5.3.6 Execute the zero-return instruction

Inquire about the RTU frame (controller → closed-loop motor).							
Slave address	Function code	Register start address		Data to be written to register		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	91H	00H	goZeroMode		

goZeroMode 01: Execute the "Return to Zero" function
02: Execute the "Coordinate Zeroing" function

Note 1: You can execute "Return to Zero" first to determine the "Zero Point Coordinates" before executing "Return to Zero".

After successful writing, the response frame is as follows:

Uplink message (Controller ← Closed-loop motor)							
Slave address	Function code	Register start address		Data already written to registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	91H	00H	01H	19H	E7H

Note: If the origin switch is already closed, the motor will rotate in the opposite direction of homeDir until the switch is open, and then return to zero.

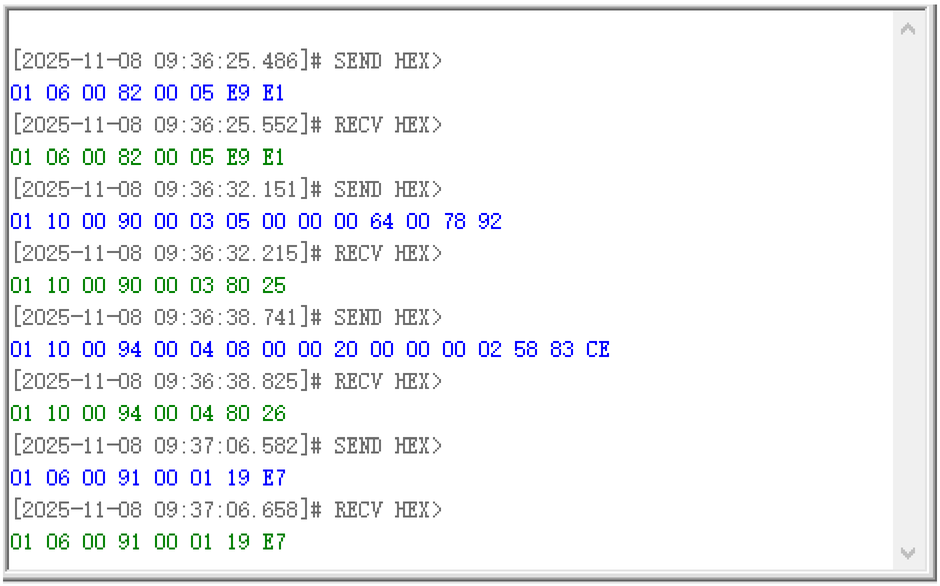
5.4 Origin switch zero-return configuration example

Connect the origin switch to the corresponding port.
When the DIP switches PIN3 and PIN2 of the 57D motor are switched to the ON position, 5V power is supplied to the external switch.

The configuration steps are as follows:

1. Configure working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set trigger level, homing direction, and homing speed.
   01 10 00 90 00 03 05 00 00 00 64 00 78 92
3. Set to zero mode
   01 10 00 94 00 04 08 00 00 20 00 00 00 02 58 83 CE
4. Perform ″ return to zero ″
   01 06 00 91 00 01 19 E7

It can be observed that: the motor (100RPM) rotates in the forward direction → the switch is triggered, and the return to zero is completed.



```
[2025-11-08 09:36:25.486]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 09:36:25.552]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 09:36:32.151]# SEND HEX>
01 10 00 90 00 03 05 00 00 00 64 00 78 92
[2025-11-08 09:36:32.215]# RECV HEX>
01 10 00 90 00 03 80 25
[2025-11-08 09:36:38.741]# SEND HEX>
01 10 00 94 00 04 08 00 00 20 00 00 00 02 58 83 CE
[2025-11-08 09:36:38.825]# RECV HEX>
01 10 00 94 00 04 80 26
[2025-11-08 09:37:06.582]# SEND HEX>
01 06 00 91 00 01 19 E7
[2025-11-08 09:37:06.658]# RECV HEX>
01 06 00 91 00 01 19 E7
```

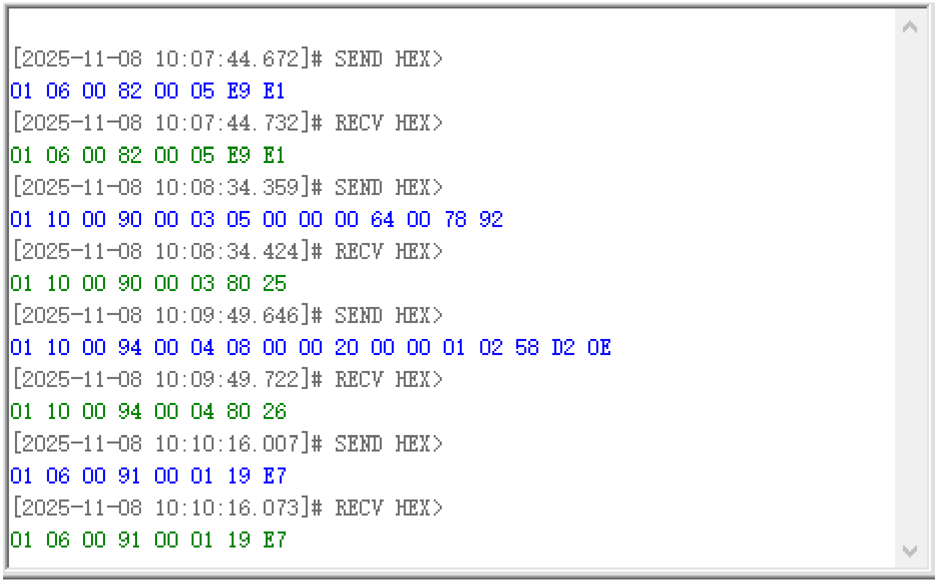
5.5 Example of mechanical limit switch homing configuration

A suitable zero-return current needs to be set for mechanical limit switches. The zero-return current should not be too large, just enough to drive the load, to avoid damaging the mechanical device.

The configuration steps are as follows:

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set the zero-return direction and zero-return speed.
   01 10 00 90 00 03 05 00 00 00 64 00 78 92
3. Set origin offset, zero-return mode, and zero-return current.
   01 10 00 94 00 04 08 00 00 20 00 00 0 1 02 58 D2 0E
4. Perform "return to zero"
   01 06 00 91 00 01 19 E7

It can be observed that: the motor (100RPM) rotates forward → after touching the mechanical limit switch → the motor stops → the motor runs in reverse to the position offset from the origin, and the return to zero is completed.



```
[2025-11-08 10:07:44.672]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:07:44.732]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:08:34.359]# SEND HEX>
01 10 00 90 00 03 05 00 00 00 64 00 78 92
[2025-11-08 10:08:34.424]# RECV HEX>
01 10 00 90 00 03 80 25
[2025-11-08 10:09:49.646]# SEND HEX>
01 10 00 94 00 04 08 00 00 20 00 00 01 02 58 D2 0E
[2025-11-08 10:09:49.722]# RECV HEX>
01 10 00 94 00 04 80 26
[2025-11-08 10:10:16.007]# SEND HEX>
01 06 00 91 00 01 19 E7
[2025-11-08 10:10:16.073]# RECV HEX>
01 06 00 91 00 01 19 E7
```

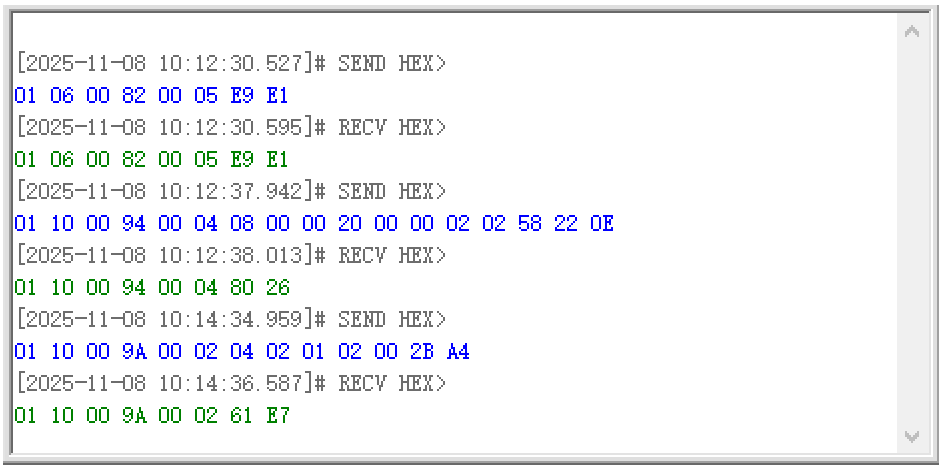
5.6 Single-lap zero-return configuration example

First, move the motor shaft to the appropriate "zero point" position.

The configuration steps are as follows:

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set to zero mode
   01 10 00 94 00 04 08 00 00 20 00 00 0 2 02 58 22 0E
3. Set the single-lap zero-return "mode", "Zero Point", "Speed", " Direction".
   01 10 00 9A 00 02 04 02 01 02 00 2B A4 (Nearest zero)

After power is cut off, move the motor shaft away from the "zero point" position.
Upon powering back on, it can be observed that the motor moves to the nearest "zero" position and completes the return to zero.



```
[2025-11-08 10:12:30.527]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:12:30.595]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:12:37.942]# SEND HEX>
01 10 00 94 00 04 08 00 00 20 00 00 02 02 58 22 0E
[2025-11-08 10:12:38.013]# RECV HEX>
01 10 00 94 00 04 80 26
[2025-11-08 10:14:34.959]# SEND HEX>
01 10 00 9A 00 02 04 02 01 02 00 2B A4
[2025-11-08 10:14:36.587]# RECV HEX>
01 10 00 9A 00 02 61 E7
```

5.7 Example of coordinate zeroing

Before performing "coordinate zeroing", the coordinates of the "zero point" must be determined:

For switch-based or mechanical limit switch-based zeroing, "origin zeroing" must be performed first.

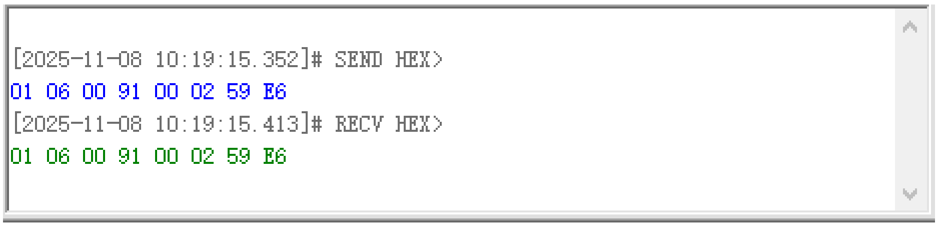
For the single-lap zero-return method, simply set the "zero point" coordinates.

1. Execute " Coordinate Zeroing "

01 06 00 91 00 02 59 E6

Once the "zero point" coordinates are determined, regardless of the motor's position, executing "coordinate return to zero" will cause the motor to return to the zero point position at high speed.

Note: The motor must be enabled before executing "coordinate return to zero".



```
[2025-11-08 10:19:15.352]# SEND HEX>
01 06 00 91 00 02 59 E6
[2025-11-08 10:19:15.413]# RECV HEX>
01 06 00 91 00 02 59 E6
```

Part 6. Left and right limit switch instructions

For limit switch wiring instructions, please refer to section 1.2 IO Port DescriptionDescription

Limit enable description:

1. When limit switch enable is enabled, in bus control mode
   Left limit switch triggered, motor stops moving to the left.
   Right limit switch triggered, motor stops moving to the right.
2. After using the limit function for the first time or changing the limit parameters, you must perform a "return to zero" operation once .
3. The limit function is invalid in pulse control mode.

6.1 Set limit switch parameters

Limit switch parameters include:

Whether the limit port is mapped:	set by 9EH, see 5.3.1.
Whether the limit function is enabled:	set by 90H, see 5.3.2 .

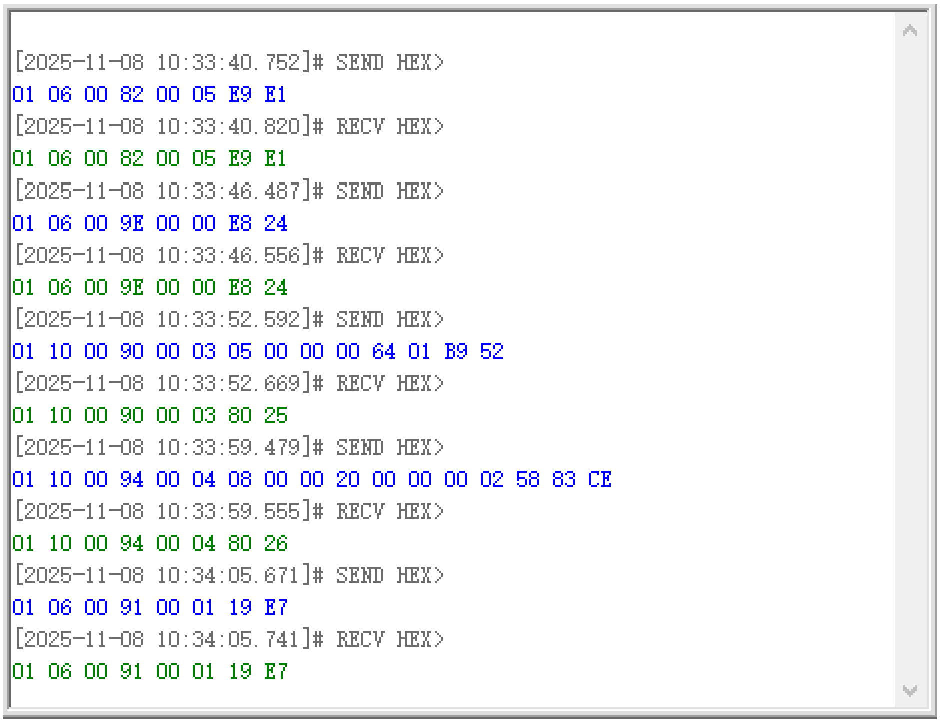
6.2 Limit switch configuration example

Taking 57D as an example, connect the limit switch to the corresponding port.
When the DIP switches PIN3 and PIN2 of the 57D motor are switched to the ON position, 5V power is supplied to the external switch.

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Disable port mapping
   01 06 00 9E 00 00 E8 24 (No mapping required)
3. Set the trigger level, homing direction, and homing speed, enable limit switch.
   01 10 00 90 00 03 05 00 00 00 64 01 B9 5 2
4. Set to zero mode
   01 10 00 94 00 04 08 00 00 20 00 00 00 02 58 83 CE
5. Perform "return to zero"
   01 06 00 91 00 01 19 E7

Once the origin switch has returned to zero, the limit switch function can be used normally.

If the above parameters remain unchanged, there is no need to reconfigure the parameters or execute "origin return to zero" on the next power-on; the limit switch function can be used directly.



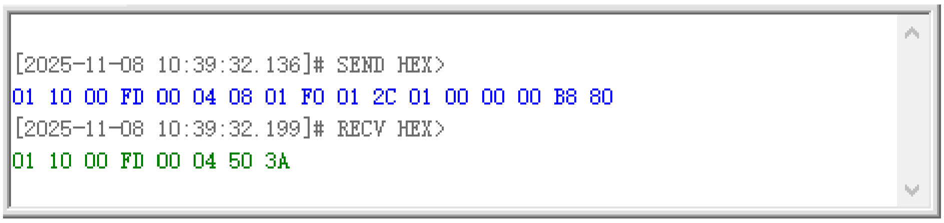
```
[2025-11-08 10:33:40.752]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:33:40.820]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 10:33:46.487]# SEND HEX>
01 06 00 9E 00 00 E8 24
[2025-11-08 10:33:46.556]# RECV HEX>
01 06 00 9E 00 00 E8 24
[2025-11-08 10:33:52.592]# SEND HEX>
01 10 00 90 00 03 05 00 00 00 64 01 B9 52
[2025-11-08 10:33:52.669]# RECV HEX>
01 10 00 90 00 03 80 25
[2025-11-08 10:33:59.479]# SEND HEX>
01 10 00 94 00 04 08 00 00 20 00 00 00 02 58 83 CE
[2025-11-08 10:33:59.555]# RECV HEX>
01 10 00 94 00 04 80 26
[2025-11-08 10:34:05.671]# SEND HEX>
01 06 00 91 00 01 19 E7
[2025-11-08 10:34:05.741]# RECV HEX>
01 06 00 91 00 01 19 E7
```

6.3 Left limit switch test example

After configuring the parameters according to 6.2

01 10 00 FD 00 04 08 01 F0 01 2C 01 00 00 00 B8 80

When the motor is running, triggering the left limit switch will stop the motor.



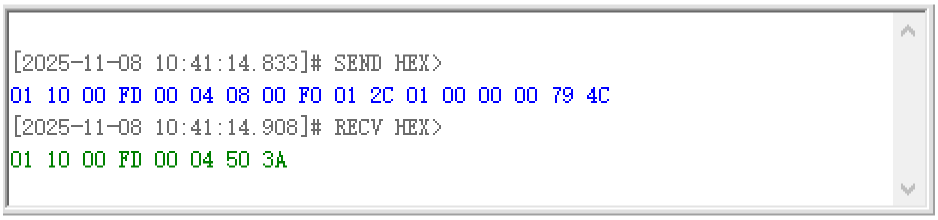
```
[2025-11-08 10:39:32.136]# SEND HEX>
01 10 00 FD 00 04 08 01 F0 01 2C 01 00 00 00 B8 80
[2025-11-08 10:39:32.199]# RECV HEX>
01 10 00 FD 00 04 50 3A
```

6.4 Right limit switch test example

After configuring the parameters according to 6.2

01 10 00 FD 00 04 08 00 F0 01 2C 01 00 00 00 79 4C

When the motor is running, triggering the left limit switch will stop the motor.



```
[2025-11-08 10:41:14.833]# SEND HEX>
01 10 00 FD 00 04 08 00 F0 01 2C 01 00 00 00 79 4C
[2025-11-08 10:41:14.908]# RECV HEX>
01 10 00 FD 00 04 50 3A
```

Part 7. Bus control mode parameter description

Note: The instructions in this section are only valid in bus control mode (SR_OPEN/SR_CLOSE/SR_vFOC) .

7.1 Description the parameters of speed and acceleration

1.speed

The speed parameter ranges from 0 to 3000. The larger the value, the faster the motor rotates.

When speed = 0, the motor stops rotating.

The maximum speed of the control mode is as follows:

	Control mode		Max speed
Open mode	Pulse interface	CR_OPEN	400(RPM)
	Serial interface	SR_OPEN	
Close mode	Pulse interface	CR_CLOSE	1500(RPM)
	Serial interface	SR_CLSOE	
FOC mode	Pulse interface	CR_vFOC	3000(RPM)
	Serial interface	SR_vFOC	

If the set speed is greater than the maximum speed of the control mode, the motor runs at the maximum speed of the control mode.

Note: The speed value is calibrated based on 16/32/64 subdivisions, and the speeds of other subdivisions need to be calculated based on 16 subdivisions.

For example, setting speed=1200

At 8 subdivisions, the speed is 2400 (RPM)

At 16/32/64 subdivisions, the speed is 1200 (RPM)

At 128 subdivisions, the speed is 150 (RPM)

2. acceleration

The value of the acceleration(acc) ranges from 0 to 255. The larger the value, the faster the motor accelerates/decelerates.

If acc=0, the motor runs without acceleration or deceleration, and runs directly at the set speed.

① accelerates

Suppose at time t1，the current speed is V_{t1} (V_{t1} < speed)
at time t2，the current speed is V_{t2}
t2 – t1 = （256-acc） * 50 (uS)

The relationship between the current speed V_{ti}, acc, and speed is as follows:

$V_{t2} = V_{t1} + 1$ (V_{t2} <= speed)

For example: acc = 236,speed = 3000

T(ms)	speed（RPM）
0	0
1	1
2	2
3	3
…	…

T(ms)	speed (RPM)
…	…
…	…
2998	2998
2999	2999
3000	3000

② decelerates

Suppose at time t1，the current speed is V_{t1} (V_{t1} > speed)
at time t2，the current speed is V_{t2}
t2 – t1 = （256-acc） * 50 (uS)

The relationship between the current speed V_{ti}, acc, and speed is as follows:

$V_{t2} = V_{t1} - 1$ (V_{t2} >= speed)

7.2 Bus control general instructions

7.2.1 Read motor operating status

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	04H	00H	F1H	00H	01H	60H	3H

Uplink message (host computer ← driver board)						
Slave address	Function code	byte count	reserve	Motor status	CRC16	
					high byte	low byte
01H	04H	02H	00H	status		

Motor operating status	status	Remark
Query failed	0	
stop	1	
speed up	2	
speed down	3	
full speed	4	
homing	5	
Calibration	6	

7.2.2 Set motor enable state

In bus control mode, the motor's enable state is no longer controlled by the level of the En pin, but is controlled by the command.

The configuration command is as follows:

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	F3H	00H	enable		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	F3H	00H	enable		

Note: This instruction is only valid in bus control mode.

7.2.3 Motor emergency stop command

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	F7H	00H	01H	F9H	F8H

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	F7H	00H	01H	F9H	F8H

Part 8. Speed Control Mode Description

Speed control mode

It can control the motor to run continuously at a set acceleration and speed.

It can control the motor to run at a set acceleration and speed for a set period of time.

It allows the motor to run at a set speed and acceleration as soon as it is turned on.

Note: The instructions in this section are only valid in bus control mode (SR_OPEN/SR_CLOSE/SR_vFOC) .

8.1 Speed control mode operation instructions

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	direction	acceleration	speed		CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	04H	dir	acc	speed			

dir: the value range is 0/1 (CCW/CW)

acc: the acceleration, the value range is 0-255

speed: the speed, the value range is 0-3000

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F6H	00H	02H	A1H	FAH

If you need the motor to automatically stop after running for a period of time, the operating command is as follows:

Downlink message (host computer → driver board)																
Slave address	Function code	Starting address		Number of registers		byte count	direction	acceleration	speed		runtime				CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo	Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	04H	08H	dir	acc	speed		runTime					

runTime Unit: 10ms Range (0 ~ 19990000H)

8.2 Speed control mode stop command

The stop command can control the motor to slow down and stop gradually, or it can control the motor to stop immediately.

When setting acc ≠ 0, the motor decelerates and stops slowly

When setting acc = 0, the motor stops immediately

Note: It is not recommended to use the immediate stop command when the motor speed exceeds 1000 RPM !

Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	direction	acceleration	speed		CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	04H	00H	acc	00H			

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F6H	00H	02H	A1H	FAH

8.3 Set automatic start command upon power-on

Downlink message (host computer → driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	FFH	00H	flag		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Register data		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	06H	00H	FFH	00H	flag		

flag = C8H Enables automatic operation upon power-on

flag = CAH disables automatic power-on operation

Note: This allows the motor to rotate continuously in the saved direction, speed, and acceleration every time it is powered on.

In other words, if you want the motor to run at a certain speed/acceleration as soon as it is powered on, you can first send a speed control mode operation command to make the motor run in the desired direction/speed/acceleration. Then, use this command to save the parameters. After powering on again, the motor will run according to the saved parameters.

If you don’t want it to run automatically upon power-on, simply send a clear command.

8.4 Speed mode running example

1. Set working mode

   01 06 00 82 00 05 E9 E1 （Bus FOC mode)

2. Set the running direction, acceleration, and speed parameters.

   01 10 00 F6 00 02 04 00 02 01 2C DD 4C

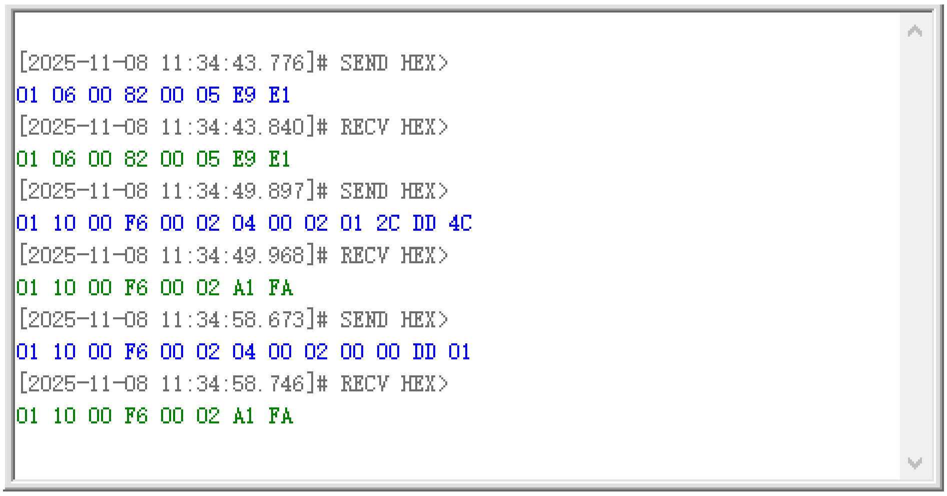
   At this point, the motor can be observed to run in the set direction, speed, and acceleration.

   To stop the operation, execute the stop command.

3. Stop command

   01 10 00 F6 00 02 04 00 02 00 00 DD 01

   At this point, the motor can be observed to stop at the set acceleration.



```
[2025-11-08 11:34:43.776]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 11:34:43.840]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 11:34:49.897]# SEND HEX>
01 10 00 F6 00 02 04 00 02 01 2C DD 4C
[2025-11-08 11:34:49.968]# RECV HEX>
01 10 00 F6 00 02 A1 FA
[2025-11-08 11:34:58.673]# SEND HEX>
01 10 00 F6 00 02 04 00 02 00 00 DD 01
[2025-11-08 11:34:58.746]# RECV HEX>
01 10 00 F6 00 02 A1 FA
```

8.5 Example of automatic operation upon power-on

1. Set working mode

   01 06 00 82 00 05 E9 E1 (Bus FOC mode)

2. Set the running direction, acceleration, and speed parameters.

   01 10 00 F6 00 02 04 00 02 01 2C DD 4C

   At this point, the motor can be observed to run in the set direction, speed, and acceleration.

3. Enable automatic operation upon power-on

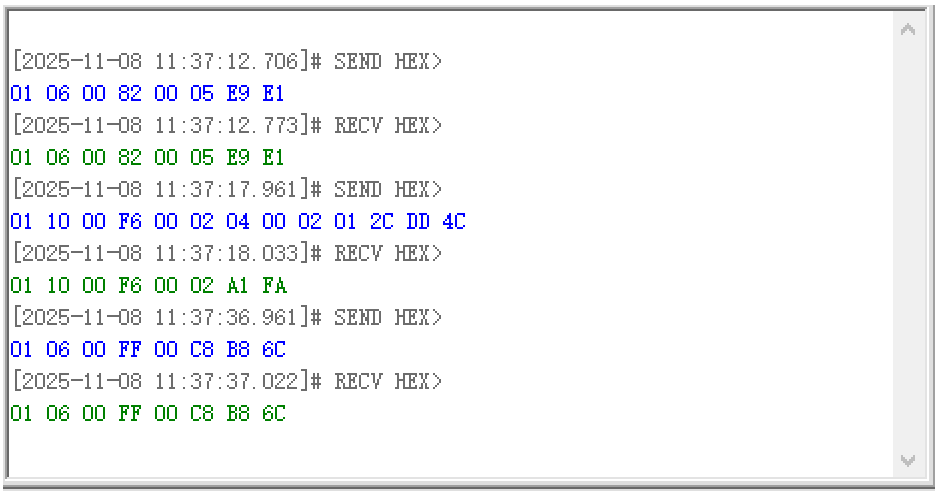
   01 06 00 FF 00 C8 B8 6C

   At this point, the motor will slow down and stop.

After powering on again, the motor can be observed to run in the set direction, speed, and acceleration.

To cancel the automatic start-up function, the command is as follows:

01 06 00 FF 00 CA 39 AD



```
[2025-11-08 11:37:12.706]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 11:37:12.773]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 11:37:17.961]# SEND HEX>
01 10 00 F6 00 02 04 00 02 01 2C DD 4C
[2025-11-08 11:37:18.033]# RECV HEX>
01 10 00 F6 00 02 A1 FA
[2025-11-08 11:37:36.961]# SEND HEX>
01 06 00 FF 00 C8 B8 6C
[2025-11-08 11:37:37.022]# RECV HEX>
01 06 00 FF 00 C8 B8 6C
```

Part 9. Position control mode description

9.1 Relative motion according to pulse number

In this mode, the motor can be controlled to run to a specified position relative to the set acceleration and speed, based on the number of pulses. It is suitable for controlling the number of revolutions required for the motor to run.

Note: The instructions in this section are only valid in bus control mode (SR_OPEN/SR_CLOSE/SR_vFOC) .

9.1.1 Pulse relative operation command

Downlink message（host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	direction	acceleration	speed	pulse count	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	FDH	00H	04H	08H	dir	acc	speed	pulses		

dir: the value range is 0/1 (CCW/CW) (uint8_t)
acc: the acceleration, the value range is 0-255 (uint8_t)
speed: the speed, the value range is 0-3000 (RPM) (uint16_t)
relpulses: the motor run steps (uint32_t)

Uplink message（host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	FDH	00H	04H	50H	3AH

9.1.2 Stop command

The stop command can control the motor to slow down and stop gradually, or it can control the motor to stop immediately.

When setting acc ≠ 0, the motor decelerates and stops slowly

When setting acc = 0, the motor stops immediately

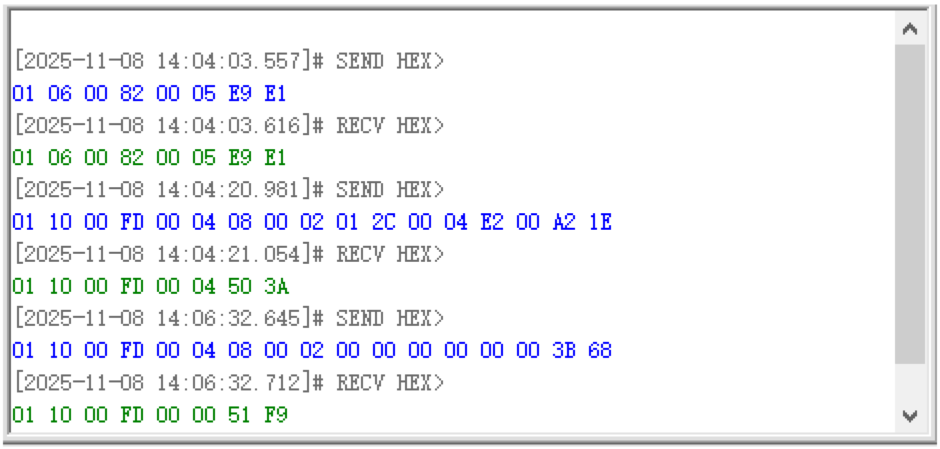
Downlink message (host computer → driver board)												
Slave address	Function code	Starting address		Number of registers		byte count	direction	acceleration	speed	pulse count	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	FDH	00H	04H	08H	00H	acc	00H	00H		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	FDH	00H	04H	50H	3AH

Note: You can also use the emergency stop command F7H to stop operation, see 7.2.3.

9.1.3 Pulse Relative Operation Example

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set the running direction , acceleration , speed , and relative pulse count.
   01 10 00 FD 00 04 08 00 02 01 2C 00 04 E2 00 A2 1E
   (At speed = 300 RPM , acce = 2, 100 revolutions clockwise (16 subdivisions))
3. If you need to slow down and stop the machine during operation, the command is as follows:
   01 10 00 FD 00 04 08 00 02 00 00 00 00 00 00 3B 68
4. If an emergency stop is required during operation, the command is as follows:
   01 06 00 F7 00 01 F9 F8



```
[2025-11-08 14:04:03.557]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:04:03.616]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:04:20.981]# SEND HEX>
01 10 00 FD 00 04 08 00 02 01 2C 00 04 E2 00 A2 1E
[2025-11-08 14:04:21.054]# RECV HEX>
01 10 00 FD 00 04 50 3A
[2025-11-08 14:06:32.645]# SEND HEX>
01 10 00 FD 00 04 08 00 02 00 00 00 00 00 00 3B 68
[2025-11-08 14:06:32.712]# RECV HEX>
01 10 00 FD 00 00 51 F9
```

9.2 Absolute motion based on pulse count

In this mode, the motor can be controlled to run to a specified position at a set acceleration and speed, based on the absolute number of pulses. It is suitable for controlling the number of revolutions required for the motor to run.

Note: Before using absolute motion, a zeroing process must be performed once (instruction 0x9 1 or 0x9 2) to mark the "zero point".

9.2.1 Pulse Absolute Operation Command

Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	Absolute Pulse	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	FEH	00H	04H	08H	acc	speed	absPulses		

speed: the speed, the value range is 0-3000(RPM) (uint16_t)

acc: the acceleration, the value range is 0-255 (uint16_t)

absPulses: the absolute pulses (int32_t)

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F EH	00H	04H		

9.2.2 Stop command

The stop command can control the motor to slow down and stop gradually, or it can control the motor to stop immediately.

When setting acc ≠ 0, the motor decelerates and stops slowly

When setting acc = 0, the motor stops immediately

Note: It is not recommended to use the immediate stop command when the motor speed exceeds 1000 RPM !

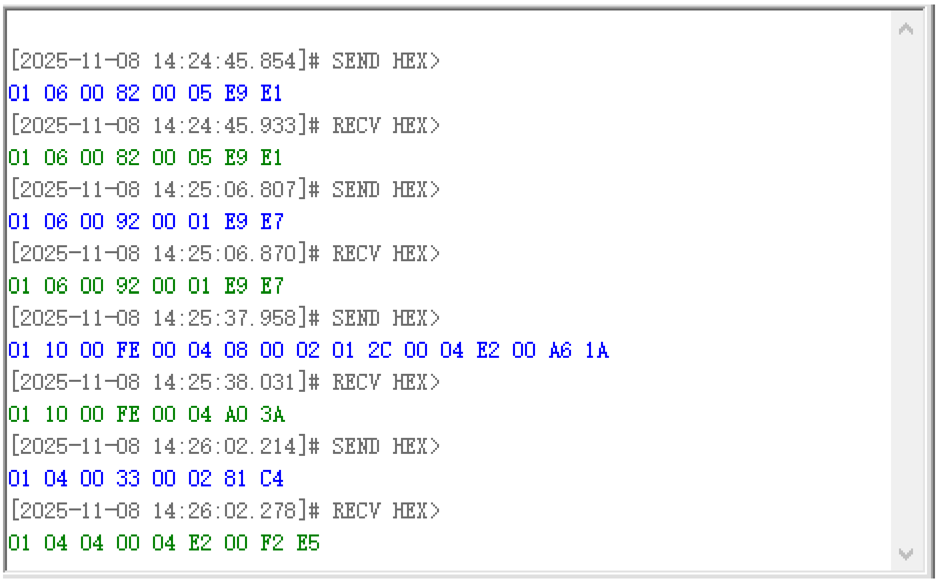
Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	absolute coordinates	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	FEH	00H	04H	08H	acc	00H	00H		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	FEH	00H	04H		

Note: You can also use the emergency stop command F7H to stop operation, see 7.2.3.

9.2.3 Pulse Absolute Operation Example

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set zero point
   01 06 00 92 00 01 E9 E7
3. Set the running acceleration, speed, and absolute pulse count.
   01 10 00 FE 00 04 08 00 02 01 2C 00 04 E2 00 A6 1A
   The motor can be observed to operate with the set parameters.
4. If you need to slow down and stop the machine during operation, the command is as follows:
   01 10 00 FE 00 04 08 00 02 00 00 00 00 00 00 3F 6C
5. If an emergency stop is required during operation, the command is as follows:
   01 06 00 F7 00 01 F9 F8
6. After the process is complete, you can view the number of pulses.
   01 04 00 33 00 02 81 C4



```
[2025-11-08 14:24:45.854]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:24:45.933]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:25:06.807]# SEND HEX>
01 06 00 92 00 01 E9 E7
[2025-11-08 14:25:06.870]# RECV HEX>
01 06 00 92 00 01 E9 E7
[2025-11-08 14:25:37.958]# SEND HEX>
01 10 00 FE 00 04 08 00 02 01 2C 00 04 E2 00 A6 1A
[2025-11-08 14:25:38.031]# RECV HEX>
01 10 00 FE 00 04 A0 3A
[2025-11-08 14:26:02.214]# SEND HEX>
01 04 00 33 00 02 81 C4
[2025-11-08 14:26:02.278]# RECV HEX>
01 04 04 00 04 E2 00 F2 E5
```

9.3 Relative motion according to coordinate values

In this mode, the motor can be controlled to move to a specified position relative to the coordinate values at a set acceleration and speed. It is suitable for controlling the motor to move to a specified coordinate position.

Note: The coordinate values are the cumulative multi-turn encoder values (16384/ turn), read using the command "31H".

9.3.1 Coordinate relative running instructions

Downlink message (host computer → driver board)										
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	relative coordinates	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F4H	00H	04H	08H	acc	speed	relAxis		

acc	(uint16_t)	Acceleration, value range 0 - 255
speed	(uint16_t)	value range is 0-3000(RPM)
relAxis	(int32_t)	represents the relative coordinates.

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F4H	00H	04H	80H	38H

9.3.2 Stop command

The stop command can control the motor to slow down and stop gradually, or it can control the motor to stop immediately.

When setting acc ≠ 0, the motor decelerates and stops slowly

When setting acc = 0, the motor stops immediately

Note: It is not recommended to use the immediate stop command when the motor speed exceeds 1000 RPM !

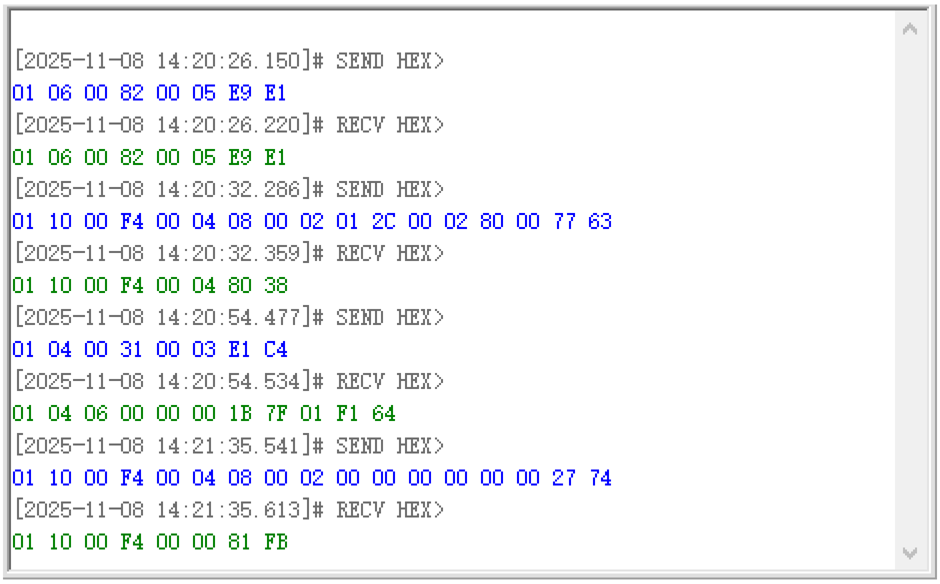
Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	relative coordinates	CRC16	
		Hi	L o	Hi	L o					Hi	L o
01H	10H	00H	F4H	00H	04H	08H	acc	00H	00H		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F4H	00H	04H	80H	38H

Note: You can also use the emergency stop command F7H to stop operation, see 7.2.3.

9.3.3 Example of coordinate relative operation

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set running acceleration, speed, and relative coordinates.
   01 10 00 F4 00 04 08 00 02 01 2C 00 02 80 00 77 63
   The motor can be observed to operate with the set parameters.
3. After the process is complete, you can view the coordinate values.
   01 04 00 31 00 03 E1 C4
4. If you need to slow down and stop the machine during operation, the command is as follows:
   01 10 00 F4 00 04 08 00 02 00 00 00 00 00 00 27 74
5. If an emergency stop is required during operation, the command is as follows:
   01 06 00 F7 00 01 F9 F8



```
[2025-11-08 14:20:26.150]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:20:26.220]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-08 14:20:32.286]# SEND HEX>
01 10 00 F4 00 04 08 00 02 01 2C 00 02 80 00 77 63
[2025-11-08 14:20:32.359]# RECV HEX>
01 10 00 F4 00 04 80 38
[2025-11-08 14:20:54.477]# SEND HEX>
01 04 00 31 00 03 E1 C4
[2025-11-08 14:20:54.534]# RECV HEX>
01 04 06 00 00 00 1B 7F 01 F1 64
[2025-11-08 14:21:35.541]# SEND HEX>
01 10 00 F4 00 04 08 00 02 00 00 00 00 00 00 27 74
[2025-11-08 14:21:35.613]# RECV HEX>
01 10 00 F4 00 00 81 FB
```

9.4 Absolute motion based on coordinate values

In this mode, the motor can be controlled to move to a specified position with a set acceleration and speed, based on the coordinate values. It is suitable for controlling the motor to move to a specified coordinate position.

Note 1: The coordinate values are the cumulative multi-turn encoder values (16384/ turn), read using the command ″3 1 ″.

Note 2 : Supports real-time updates of speed and coordinates, meaning that a new command can be issued to change the speed and coordinates while the previous command is running.

Note 3: Before using absolute motion, a zeroing process must be performed once (instruction 0x9 1 or 0x9 2) to mark the ″zero point″.

9.4.1 absolute coordinate execution command

Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	absolute coordinates	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F5H	00H	04H	08H	acc	speed	absAxis		

acc (uint16_t) Acceleration, value range 0 - 255

speed (uint16_t) represents the speed, ranging 0 to 3000 RPM.

absAxis (int32_t) the absolute coordinate

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F5H	00H	04H	D1H	F8H

9.4.2 Stop command

The stop command can control the motor to slow down and stop gradually, or it can control the motor to stop immediately.

When acc is set to ≠ 0 , the motor decelerates and stops slowly.

When acc is set to 0 , the motor stops immediately.

Note: It is not recommended to use the immediate stop command when the motor speed exceeds 1000 RPM !

Downlink message (host computer → driver board)											
Slave address	Function code	Starting address		Number of registers		byte count	acceleration	speed	absolute coordinates	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F5H	00H	04H	08H	acc	00H	00H		

Uplink message (host computer ← driver board)							
Slave address	Function code	Register start address		Number of registers		CRC16	
		high byte	low byte	high byte	low byte	high byte	low byte
01H	10H	00H	F5H	00H	04H	D1H	F8H

Note: You can also use the emergency stop command F7H to stop operation, see 7.2.3.

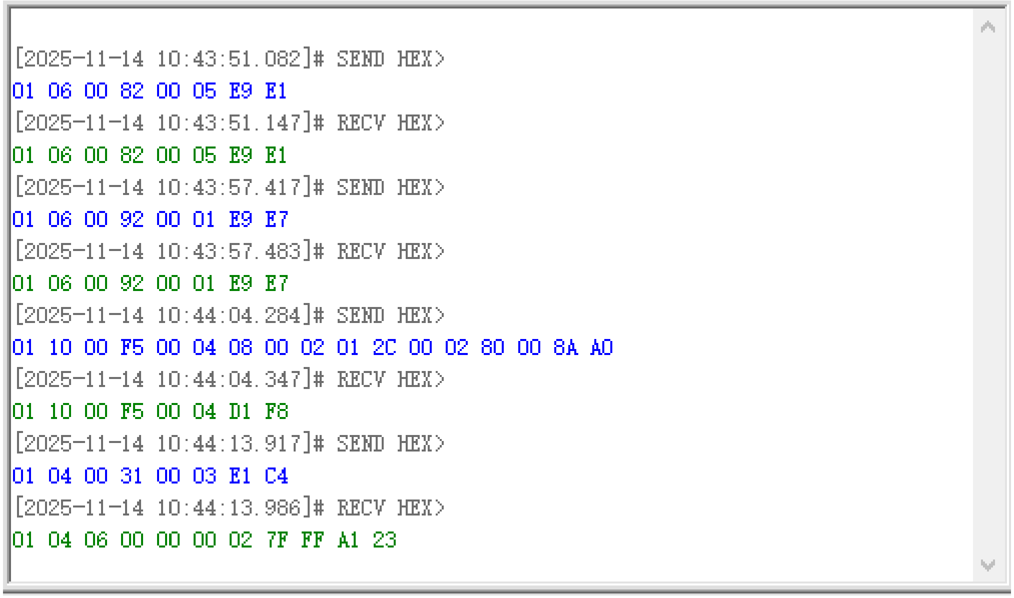
9.4.3 Example of running with absolute coordinates

You need to first set the "zero point" coordinates. There are two ways to do this:

1. "Return to Zero" setting;
2. Setting the "92H command mode";

If not set, the default power-on position is the "zero point" coordinate.

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set zero point
   01 06 00 92 00 01 E9 E7
3. Set runtime acceleration , speed, and absolute coordinates.
   01 10 00 F5 00 04 08 00 02 01 2C 00 02 80 00 8A A0
   The motor can be observed to operate with the set parameters.
4. If you need to slow down and stop the machine during operation, the command is as follows:
   01 10 00 F5 00 04 08 00 02 00 00 00 00 00 00 DA B7
5. If an emergency stop is required during operation, the command is as follows:
   01 06 00 F7 00 01 F9 F8
6. After the process is complete, you can view the coordinate values.
   01 04 00 31 00 03 E1 C4



```
[2025-11-14 10:43:51.082]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-14 10:43:51.147]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-14 10:43:57.417]# SEND HEX>
01 06 00 92 00 01 E9 E7
[2025-11-14 10:43:57.483]# RECV HEX>
01 06 00 92 00 01 E9 E7
[2025-11-14 10:44:04.284]# SEND HEX>
01 10 00 F5 00 04 08 00 02 01 2C 00 02 80 00 8A A0
[2025-11-14 10:44:04.347]# RECV HEX>
01 10 00 F5 00 04 D1 F8
[2025-11-14 10:44:13.917]# SEND HEX>
01 04 00 31 00 03 E1 C4
[2025-11-14 10:44:13.986]# RECV HEX>
01 04 06 00 00 00 02 7F FF A1 23
```

9.4.4 Real-time updates of running examples

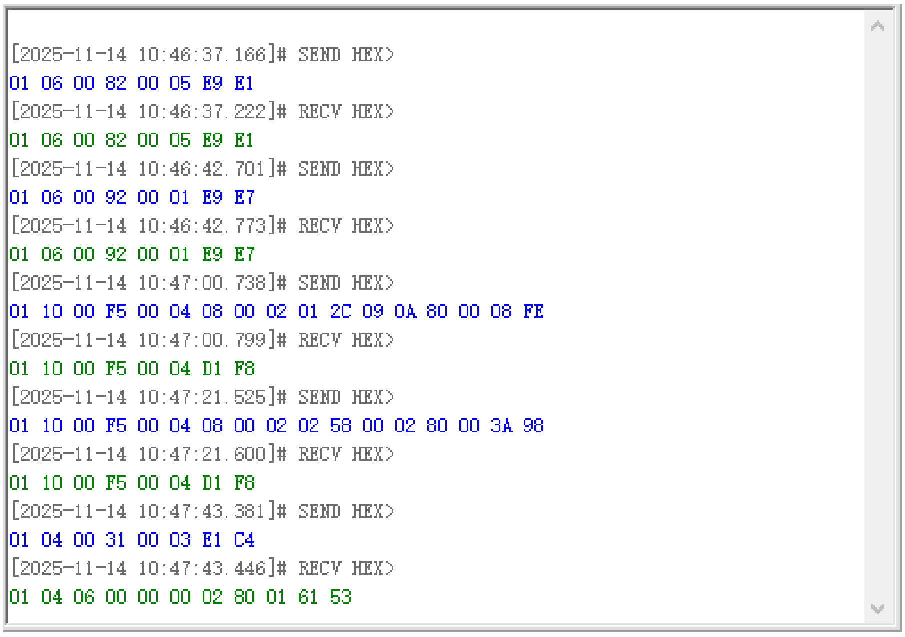
This mode supports real-time updates of speed and coordinates, meaning that new commands can be issued to change speed and coordinates while the previous command is running.

You need to first set the "zero point" coordinates. There are two ways to do this:

1. "Return to Zero" setting;
2. Setting the "92H command mode";

If not set, the default power-on position is the "zero point" coordinate.

1. Set working mode
   01 06 00 82 00 05 E9 E1 (Bus FOC mode)
2. Set zero point
   01 06 00 92 00 01 E9 E7
3. Set runtime acceleration , speed , and absolute coordinates.
   01 10 00 F5 00 04 08 00 02 01 2C 09 0A 80 00 08 FE
   Wait for the motor to run for about 20 seconds.
4. Real-time update speed , absolute coordinates
   01 10 00 F5 00 04 08 00 02 02 58 00 02 80 00 3A 98
   The motor update speed and coordinate operation can be observed.
5. After the process is complete, you can view the coordinate values.
   01 04 00 31 00 03 E1 C4
6. If you need to slow down and stop the machine during operation, the command is as follows:
   01 10 00 F5 00 04 08 00 02 00 00 00 00 00 00 DA B7
7. If an emergency stop is required during operation, the command is as follows:
   01 06 00 F7 00 01 F9 F8

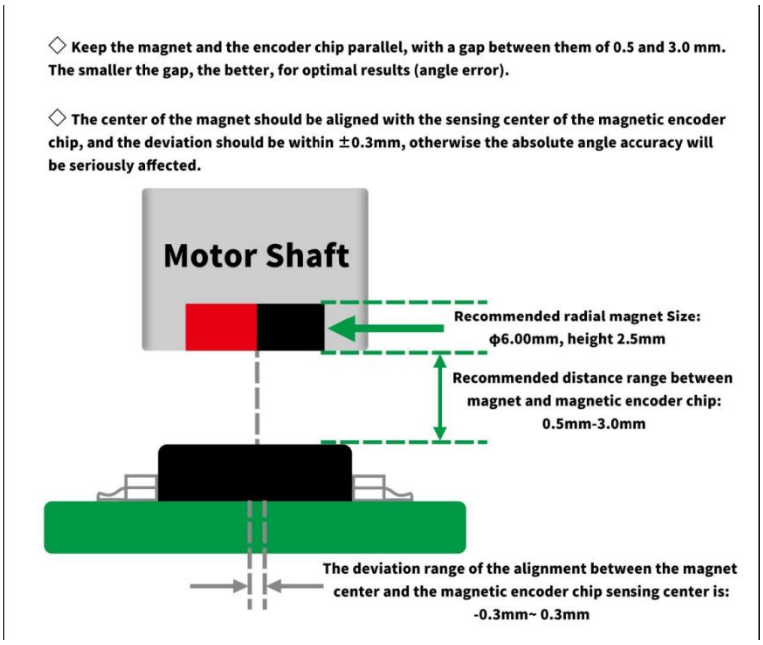


```
[2025-11-14 10:46:37.166]# SEND HEX>
01 06 00 82 00 05 E9 E1
[2025-11-14 10:46:37.222]# RECV HEX>
01 06 00 82 00 05 E9 E1
[2025-11-14 10:46:42.701]# SEND HEX>
01 06 00 92 00 01 E9 E7
[2025-11-14 10:46:42.773]# RECV HEX>
01 06 00 92 00 01 E9 E7
[2025-11-14 10:47:00.738]# SEND HEX>
01 10 00 F5 00 04 08 00 02 01 2C 09 0A 80 00 08 FE
[2025-11-14 10:47:00.799]# RECV HEX>
01 10 00 F5 00 04 D1 F8
[2025-11-14 10:47:21.525]# SEND HEX>
01 10 00 F5 00 04 08 00 02 02 58 00 02 80 00 3A 98
[2025-11-14 10:47:21.600]# RECV HEX>
01 10 00 F5 00 04 D1 F8
[2025-11-14 10:47:43.381]# SEND HEX>
01 04 00 31 00 03 E1 C4
[2025-11-14 10:47:43.446]# RECV HEX>
01 04 06 00 00 00 02 80 01 61 53
```

Part 10. Frequently Asked Questions and Precautions

10.1 Precautions

1. Power input voltage 12V-24V;
2. Do not plug or unplug the power cord or signal cable while the circuit is powered on, as this may damage the driver board.
3. Press and hold the "Next" button, then power on the device to quickly restore factory default settings.
4. Do not apply a load when calibrating the motor;
5. When the driver board is first installed on the motor, or after the motor wiring sequence is changed, the motor needs to be recalibrated.
6. If the system displays "Phase Line Error!" before powering on for calibration:
   a) Check the motor wiring sequence;
   b) Check the power supply voltage and output power (24V/1A, 12V/2A);
   c) If the MKS APT module is connected to the motherboard power supply, try connecting the MKS APT module to ports X, Y, Z, E, etc., and then power on for calibration.
   d) Before calibration, do not use the MKS APT module for power supply; connect the power supply directly to V+ and Gnd.
7. If the LED light stays on after powering on, or if the screen displays an error message, please refer to the "Frequently Asked Questions" for troubleshooting.
8. We recommend that users purchase our matching motors directly to avoid the risk of incompatibility. If you choose to use a custom motor in closed-loop mode instead of our motors, the following conditions must be met:
   (1) The motor step distance is 1.8 degrees.
   (2) The motor's internal resistance is less than 10 ohms.
   (3) A radial magnet can be installed on the back of the motor.
   (4) When installing magnets, please note the following diagram:



◇ Keep the magnet and the encoder chip parallel, with a gap between them of 0.5 and 3.0 mm. The smaller the gap, the better, for optimal results (angle error).

◇ The center of the magnet should be aligned with the sensing center of the magnetic encoder chip, and the deviation should be within ±0.3mm, otherwise the absolute angle accuracy will be seriously affected.

Motor Shaft
Recommended radial magnet Size: φ6.00mm, height 2.5mm
Recommended distance range between magnet and magnetic encoder chip: 0.5mm-3.0mm
The deviation range of the alignment between the magnet center and the magnetic encoder chip sensing center is: -0.3mm~ 0.3mm

10.2 Frequently Asked Questions

No	Question	Solution
1	Not Cal	Calibrate the motor.
2	Reverse Lookup Error!	Calibrate Fail, Check magnet and motor shaft.
3	Magnet Loss!	Not install the magent.
4	Magnet Strong!	the magnet too near.
5	Magnet Weak!	the magnet too far.
6	Encoder Error!	Check magnet and motor shaft.
7	Offset Current Error!	Reference voltage error.
8	Phase Line Error!	The motor line sequence is wrong or the power supply is not enough.
9	Wrong Protect!	Locked-rotor protection.
10	Coming Back to Origin..	Going back to zero.
11	Reboot Again	The motor need to be restart.
12	Press Next Key To Fixed	Press Next Key， until it reboot.
13	Low Voltage Error!	The supply voltage is too low.

Part 11.　Schematic

Please download 《MKS SERVO42D/57D V1.0 Schematic.pdf》 in
https://github.com/makerbase-motor/MKS-SERVO42D
https://github.com/makerbase-motor/MKS-SERVO57D

Part 12.　contact us

https://makerbase.aliexpress.com/
https://www.youtube.com/channel/UC2i5I1tcOXRJ2ZJiRxwpCUQ
https://github.com/makerbase-motor