# MKS SmartMotor

# RS485 User Manual V1.0.0

Note: This manual corresponds to the firmware version V 1.0.0

| MKS SmartMotor_RS485 Version Description | | | |
|---|---|---|---|
| Manual Version | content | Firmware version | Date |
| V1.0.0 | First release | V1.0.0 | AUG-2025 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Part1. Product Overview

## 1.1 Product Introduction

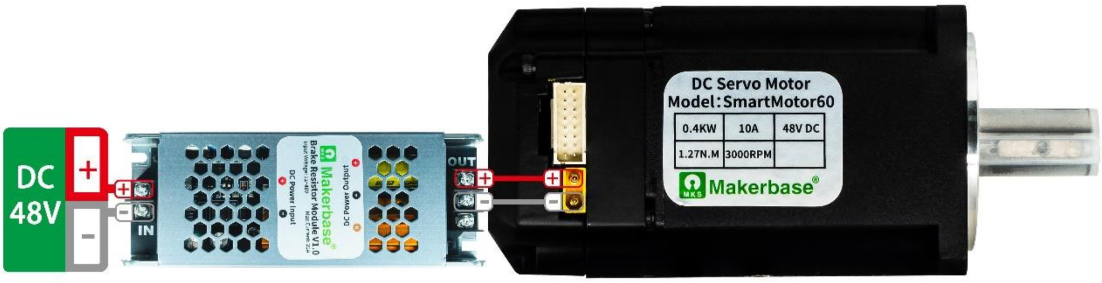
MKS SmartMotor The permanent magnet DC servo motor is a product independently developed by the Maker Base to meet market demand and in accordance with industrial standards . It supports pulse interface , RS485 interface and CAN interface , has a built-in efficient FOC vector algorithm , uses a high-precision encoder , has high positioning accuracy and fast response speed . It is suitable for applications such as small robotic arms , medical equipment , engraving machines, automation products and electronic competitions .

## 1.2 Technical Parameters

| | |
|---|---|
| Motor performance | Maximum power 400W / 750W, rated torque 1.27NM / 2.40NM |
| | Rated speed 3000RPM, response speed less than 0.5ms |
| | Speed fluctuation rate: <±0.03 (load 0~100%): <±0.02×(0.9~1.1) power supply voltage |
| Position control mode | Pulse mode: Maximum receiving frequency 100KHZ (duty ratio 1:1) |
| | Pulse mode: Pulse + direction; A + B orthogonal pulse; double pulse (CW/CCW) |
| | Bus mode: supports multi-segment position automatic cycle operation, relative position operation, absolute position operation, etc. |
| Speed control mode | Bus mode: supports multi-speed automatic cycle operation, etc. |
| Torque control mode | Bus Mode |
| Return to origin function | Supports multiple return to origin modes such as switch, torque, single turn, and origin offset function |
| Gain Adjustment | Supports manual adjustment and internal rigidity grade table adjustment |
| Monitoring parameters | Motor phase current, bus voltage, module temperature, fault alarm, operating status and other parameters |
| Protection parameters | Over/under voltage, over current, overload, encoder abnormality, position error, stall, etc. |
| Input port function | Motor enable, fault alarm reset, emergency stop, return to zero enable, origin switch, overtravel switch, etc. |
| Output port function | Fault alarm, positioning completed, speed reached, torque reached, zero return completed, pulse frequency division output, etc. |
| RS485 communication | Support MKS custom protocol and Modbus-RTU protocol |
| | The speed is optional from 2400bps to 115200bps, the default is 57600bps |
| | The address is 0~255, 0 is the broadcast address, 1 is the default address |
| CAN communication | Use standard frames and support MKS custom protocols |
| | The speed is optional: 125K/250K/500K/1000K, the default is 500K |
| | Address 0~2047 is optional, 0 is the broadcast address, 1 is the default address |
| Power supply | DC48V |
| Usage Environment | Working temperature: 0℃ ~ 55℃ Storage temperature: -20℃ ~ +80℃ |

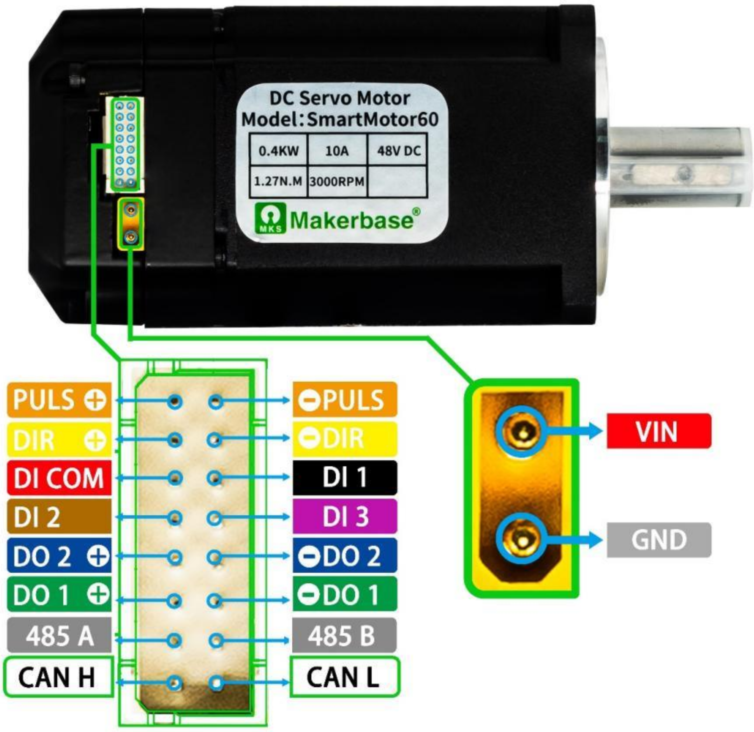
## 1.3 Braking mode

SmartMotor60 does not have an external brake resistor interface. When the load inertia is large, it is recommended to purchase a dedicated brake module to ensure the quality of the bus voltage. The brake module wiring is as shown below:



SmartMotor80 has an external brake resistor interface. You only need to select a suitable brake resistor. No brake module is required.

## 1.4 Interface Description



## 1.5 Indicator light status table

The status indicator light is divided into a green operating indicator light and a red fault indicator light, and the flashing frequency is 0.5HZ

| Green indicator light | Motor status |
|---|---|
| Flash | Motor enable is ON |
| Always on | Motor enable OFF |
| Constantly extinct | The drive is not powered on or is faulty, and the red light flashes; |

For the status of the red fault indicator light, see: Part 11 "Fault Code Correspondence Table"

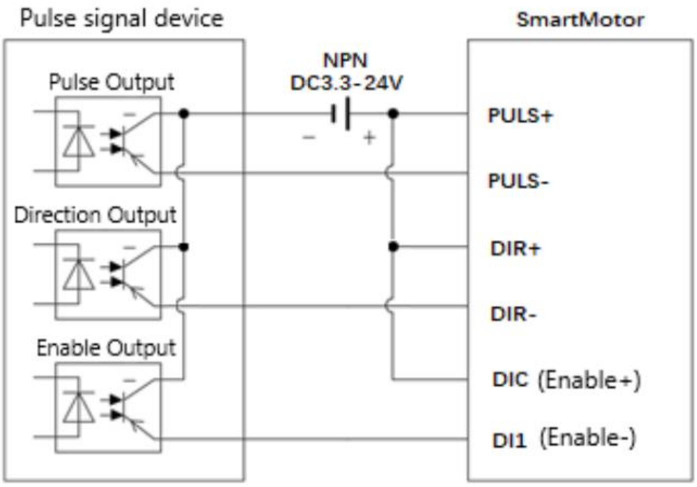
# Part2. Wiring method

## 2.1 Pulse control wiring method

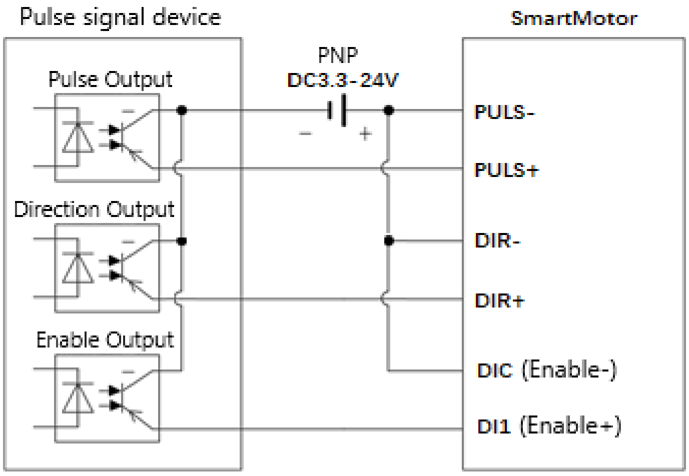
The pulse interface supports 3.3V-24V input, the maximum receiving pulse frequency is 100KHZ, and the duty cycle is as close to 1:1 as possible, otherwise the pulse may be lost, resulting in abnormal positioning.

The PULS/DIR port has a 10mA current limiting resistor, and the DI1 port has a 2KΩ current limiting resistor, which can directly input 3.3V-24V signals without the need for an external current limiting resistor.

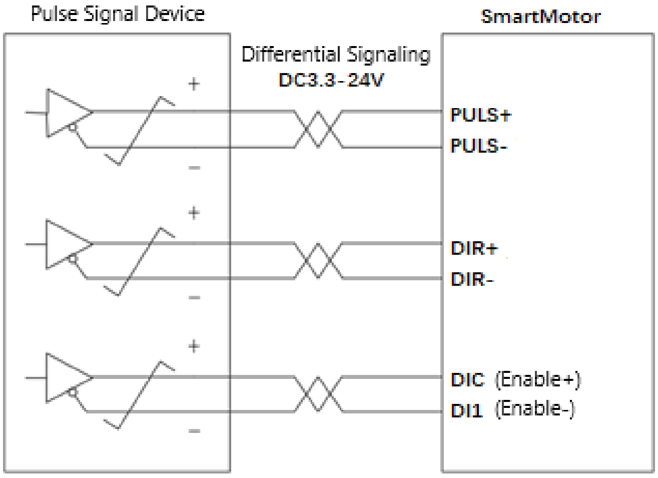
### 2.1.1 Common anode wiring method



### 2.1.2 Common cathode wiring method



### 2.1.3 Differential Wiring Method



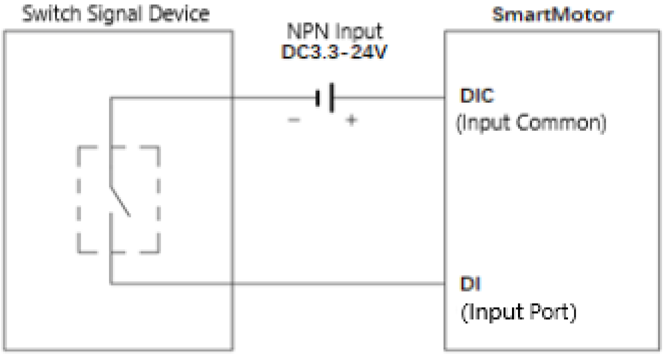
## 2.2 Input port wiring method

The input port includes the common port DIC, input ports DI1, DI2, and DI3, among which DI1 can be used as a motor enable signal.

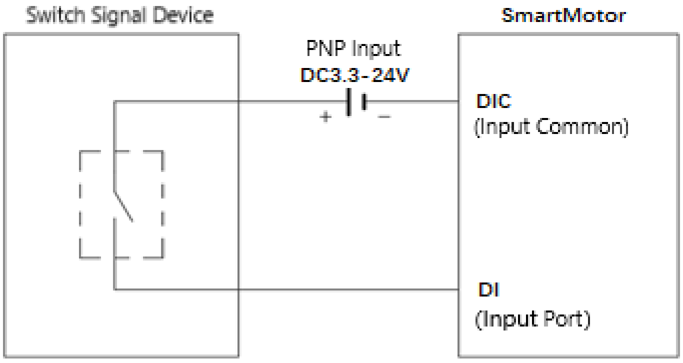
The input voltage range is DC3.3-24V, and the maximum input signal frequency is 100Hz.

DI1, DI2, and DI3 ports are all equipped with 2KΩ current -limiting resistors, and the wiring methods are consistent.

### 2.2.1 Common anode wiring method

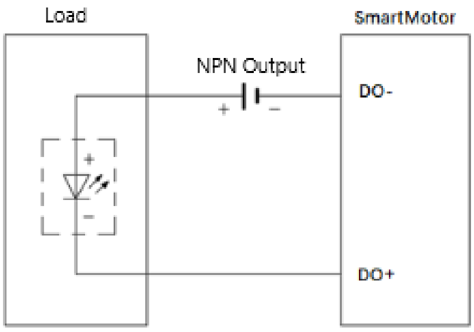


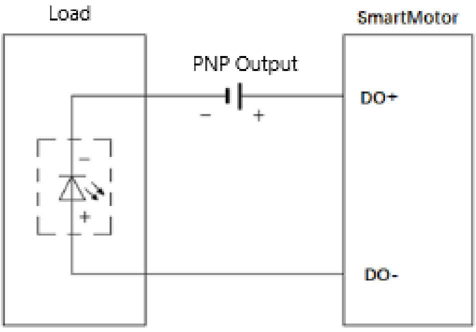
### 2.2.2 Common cathode wiring method



## 2.3 Output port wiring method

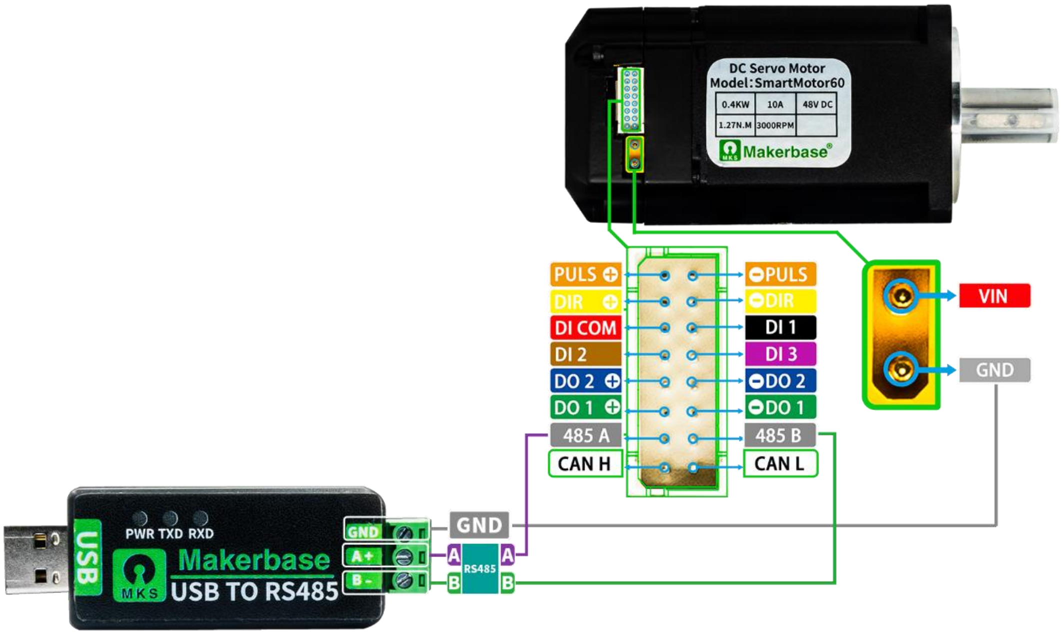
There are 2 output ports: DO1+/DO1-, DO2+/DO2-.
DO1 maximum driving current is 800mA.
The maximum driving current of DO2 is 400mA.
If you need to drive a larger load, please use a relay for conversion.



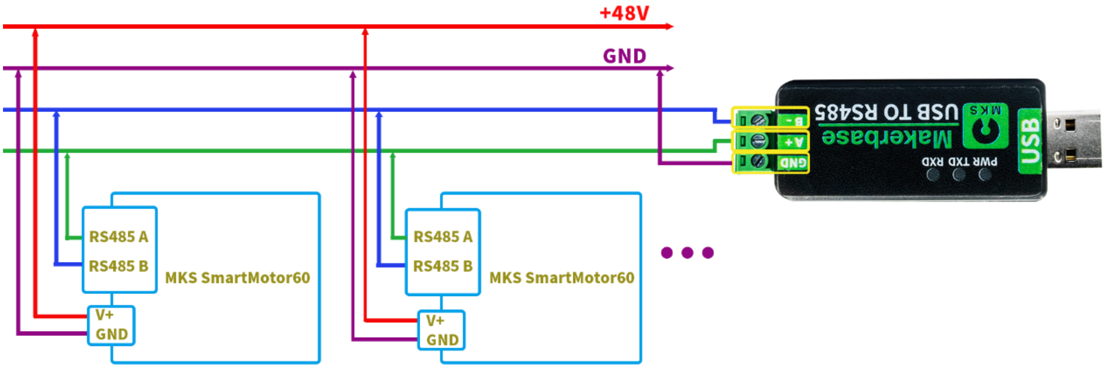


## 2.4 RS485 Wiring Method

### 2.4.1 Single machine wiring method



### 2.4.2 Multi-machine wiring method



When connecting multiple machines, if the communication signal noise is large, it is recommended to add a 120 ohm terminal resistor to the last motor to ensure signal quality.

# Part3. Serial instruction format description

| Downlink frame (host computer → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Frame Header | Slave Address | Function code | Instruction data | | | | CRC checksum |
| FA | addr | code | | | | | CRC |
| | | | | | | | |
| Uplink frame (host computer ← SmartMotor ) | | | | | | | |
| Frame Header | Slave Address | Function code | Return data | | | | CRC checksum |
| FB | addr | code | | | | | CRC |

The command data and return data are in big-endian mode.

Downlink frame header FA , Uplink frame header FB

The slave address ( addr ) range is 0 0~255 , and the default address is 01.

Among them, 00 is the broadcast address ;

The function code ( code ) executes the corresponding instruction , for example, 0x82 sets the working mode .

command data or return data , see "Serial Port Command Description ".

The CRC check code is CHECKSUM 8bit

For example, the command "FA 01 80 00 CRC"

CRC = (0xFA + 0x01 + 0x80 + 0x00) & 0xFF = 0x17B & 0xFF = 0x7B

When the host computer sends a command, the timing between the bytes of a single command (FA ... CRC) must be continuous, and there cannot be more than one byte delay, otherwise the lower computer may fail to receive the command.

In position control mode, there are two types of position counting units: "pulse unit" and "encoder unit".

Pulse unit: the angle the motor rotates when it receives one pulse.

For example, if the 84H instruction sets 3200 pulses/circle, then 1 pulse unit $=\frac{1}{3200}\times 360^\circ \quad = 0.1125^\circ$

Encoder unit: the physical unit of the motor encoder.

The encoder resolution is 17 bits ( $2^{17} = 131072$), 1 encoder unit $=\frac{1}{131072}\times 360^\circ$ $= 0.00275^\circ$

**If the slave receives an undefined function code, the returned data is as follows:**

| Uplink frame (host computer ← SmartMotor ) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Frame Header | Slave Address | Function code | Return data | | | | | CRC checksum |
| FB | addr | code | FB | FF | FF | FF | FF | CRC |

Note: When sending commands using the broadcast address or group address, the slave will not respond.

# Part4. Serial Command Description

Note 1: When a command is sent using a broadcast address or a group address, the slave will not respond.

Note 2: For MODBUS-RTU protocol instructions, see in 《RTU manual v1.0.0》 .

Note 3: In the following sections, the default slave address is 01.

## 4.1 Read-only parameter instructions

### 1. Read absolute position

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 31H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Command absolute position | Checksum |
| FB | addr | 31H | (int32_t) | CRC |

Record the absolute position in pulse units after power-on ( enabled or disabled ) .

The single-turn value range is variable and is related to the number of pulses per turn (subdivision) set by the 84H instruction.

For example, set the number of pulses per circle to 3200

Calculation rule: The motor shaft rotates counterclockwise for one circle, and the pulse unit is + 3200 ;

The motor shaft rotates one circle clockwise, pulse unit - 3200 ;

Note: The command absolute position can be cleared by the 92H command.

## 2. Reading the absolute position of the encoder

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 35H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Bytes 8-11 | Byte 12 |
| Frame Header | Slave Address | Function code | High 32 bits | Lower 32 bits | Checksum |
| FB | addr | 35H | (int32_t) | (int32_t) | CRC |

Records the absolute position in encoder units after power-on ( enabled or disabled ) .

Using 17-bit encoder, single-turn value range 0~0x20000

Calculation rule: The motor shaft rotates counterclockwise for one circle, and the encoder value is +0x20000;

The motor shaft rotates one circle clockwise, and the encoder value is -0x 20000;

For example :

current encoder value is 0x 1492A. After one clockwise rotation (- 0x20000), the encoder value is 0xFFFFFFFFFFFF492A.

current encoder value is 0x 1492A. After one counterclockwise rotation (+0x20000), the encoder value is 0x3492A.

Note: The absolute position of the encoder cannot be cleared by the 92H instruction.

## 3. Read the real-time speed of the motor

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 32H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Real-time speed | Checksum |
| FB | addr | 32H | (int16_t) | CRC |

Note: The speed unit is RPM , counterclockwise speed is greater than 0, and clockwise speed is less than 0.

## 4. Read the cumulative number of input pulses

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 33H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Number of pulses | Checksum |
| FB | addr | 33H | (int32_t) | CRC |

## 5. Reading fault codes

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 3BH | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Fault Codes | Checksum |
| FB | addr | 3BH | (uint16_t) | CRC |

Note 1: For the definition of fault codes, see Part 11 "Fault Code Correspondence Table"

Note 2: The fault alarm can be reset by command 41H

## 6. Read real-time torque output value

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 36H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Torque output value | Checksum |
| FB | addr | 36H | (int16_t) | CRC |

Note: Unit: 0.1% (100.0% corresponds to the rated torque of the motor)

## 7. Read bus voltage and phase current values

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 37H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Bus voltage | Phase current | Checksum |
| FB | addr | 37H | (uint16_t) | (int16_t) | CRC |

Note: Bus voltage unit is 0.1V, phase current unit is 0.01A

## 8. Reading temperature value

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 38H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Temperature value | Checksum |
| FB | addr | 38H | (int16_t) | CRC |

Note: Unit: ° C

### 9. Read encoder position deviation

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 39H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Position deviation | Checksum |
| FB | addr | 39H | (int32_t) | CRC |

### 10. Read servo enable status

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 3AH | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Enable status | Checksum |
| FB | addr | 3AH | (uint8_t) | CRC |

Note: 1: Enabled 0: Note enabled

## 11. Read version information

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 40H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Bytes 5-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Hardware version | Firmware version | Checksum |
| FB | addr | 40H | hardVer | firmVer [ 3] | CRC |

Firmware version firmVer [0] = 1  firmVer [1] = 0  firmVer [2] = 0

Corresponding version V1.0.0

The hardware versions correspond to the following：

| Board Type | hardVer |
|---|---|
| 60 Series (400W) | 60H |
| 80 Series (750W) | 80H |

## 4.2 Set general parameter command

Note: After setting the parameters, do not save them directly. After all the parameters are set, use the 42H instruction to save them uniformly.

### 1. Set the working mode

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
| Frame Header | Slave Address | Function code | Main Mode | Secondary Mode | Checksum |
| FA | addr | 82H | major | minor | CRC |

| Working Mode | major | minor | Mode Description |
|---|---|---|---|
| Position Mode | 00 | 00 | Pulse+direction CW (default) |
| | 00 | 01 | Pulse+direction CCW |
| | 00 | 02 | AB phase pulse |
| | 00 | 03 | CW/CCW double pulse |
| | 00 | 04 | Bus control |
| Speed Mode | 01 | 00 | Bus Control Single Speed Mode |
| | 01 | 01 | Bus Controlled Multi-Speed Mode |
| Torque mode | 02 | 00 | Bus control |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 82H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

## 2. Set the number of pulses per revolution

how many pulse inputs are needed for the motor to rotate 1 circle. (Default value is 1000)

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Number of pulses per revolution | Checksum |
| FA | addr | 84H | pulsPR (uint32_t) | CRC |

For example: pulsPR = 3200, which means that 3200 pulses are required for the motor to rotate 1 circle.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 84H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 3. Set the En pin effective level

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Enable level | Checksum |
| FA | addr | 85H | uint8_t | CRC |

00 corresponds to low level enable (L) (default value)

01 corresponds to high level enable (H)

02 corresponds to always enabled (Hold)

Note: When the enable level is set to L or H, DI1 automatically functions as En

When the enable level is set to Hold, DI1 can be used for other functions.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 85H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

Note: This command is only valid in non-bus control mode. In bus control mode, use F3 command to enable the motor.

### 4. Set the motor rotation direction

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Direction of rotation | Checksum |
| FA | addr | 86H | uint8_t | CRC |

00: CCW is positive direction and CW is negative direction (default value)

01: CW is positive direction and CCW is negative direction

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 86H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

Note: This instruction needs to be saved with the 42H instruction and will take effect after reset and restart.

### 5. Enable disconnect shutdown mode selection

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Shutdown mode | Checksum |
| FA | addr | 87H | uint8_t | CRC |

00: Free stop, keep free state after stop (default value)

01: Zero speed stop, keep free state after stop

02: Zero speed stop, keep damping state after stop (recommended for vertical load)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 87H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 6. Set the encoder frequency division pulse number

That is, set the number of pulses output by the DO port when the motor rotates 1 circle.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Output pulse number | Checksum |
| FA | addr | 88H | outPulses | CRC |

outPulses : Output pulse number (range 1~50, default value 1)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 88H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 7. Set the serial port baud rate

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Baud rate | Checksum |
| FA | addr | 8A | baudrate (uint8_t) | CRC |

| | |
|---|---|
| 00 | 2400 |
| 01 | 4800 |
| 02 | 9600 |
| 03 | 19200 |
| 04 | 38400 |
| 05 | 57600 (default) |
| 06 | 115200 |

For example: send FA 01 8A 0 6 8B and set the baud rate to 115200 .

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 8A | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 8. Set the slave address

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Slave Address | Checksum |
| FA | addr | 8BH | address(uint8_t) | CRC |

(Default slave address 0x01)

for example:

Send FA 01 8 B 02 88 to set the slave address to 0x02.

Send FA 01 8 B 50 D7 to set the slave address to 0x50.

…

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 8BH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

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

## 9. Set group address

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Group Address | Checksum |
| FA | addr | 8DH | address(uint8_t) | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 8DH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

Group address description:

Assuming there are 6 motors, the address settings are as follows

| | Broadcast Address | Slave Address | Group Address |
|---|---|---|---|
| Motor 1 | 0 | 1 | 0x50 |
| Motor 2 | 0 | 2 | 0x50 |
| Motor 3 | 0 | 3 | 0x50 |
| Motor 4 | 0 | 4 | 0x51 |
| Motor 5 | 0 | 5 | 0x51 |
| Motor 6 | 0 | 6 | 0x51 |

Send FA 01 FD 00 C8 00 64 00 00 7D 00 CRC Motor 1 run
Send FA 00 FD 00 C8 00 64 00 00 7D 00 CRC Motor 1 -6 running
Send FA 50 FD 00 C8 00 64 00 00 7D 00 CRC Motor 1 -3 running
Send FA 51 FD 00 C8 00 64 00 00 7D 00 CRC Motor 4-6 running

Note: When sending commands using the group address, the slave will not respond.

## 10. Set the slave response mode

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Answer Mode | Checksum |
| FA | addr | 8CH | ackMode (uint8_t) | CRC |

| | |
|---|---|
| ackMode = 0 | slave no response mode |
| ackMode = 1 | The slave does not actively initiate data mode |
| ackMode = 2 | slave actively initiates data mode (default value) |

Answer mode description:
Take single position control operation as an example:
The host sends FA 01 FD 00 C8 00 64 00 00 7D 00 CRC

**a. No response mode ( ackMode = 0 ）**

The slave does not return any information

**b. Do not actively initiate data mode ( ackMode = 1 )**

slave immediately returns the position control start 0 1 or fails 00

**c. Default mode ( ackMode = 2)**

slave immediately returns the position control start 0 1 or fails 00
After the motor is finished running, it returns to 0 2

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 8CH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 11. Setting the MODBUS-RTU communication protocol

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | RTU Mode | Checksum |
| FA | addr | 8E | isRtuMode | CRC |

isRtuMode = 0 sets MKS command mode (default value)

isRtuMode = 1 sets MODBUS-RTU mode

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 8E | status<br>(uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

Note 1: This instruction needs to be saved with the 42H instruction and will take effect after reset and restart.

Note 2: For instructions on using MODBUS-RTU mode, see "".

In MODBUS-RTU mode, to return to MKS command mode, use the following command:

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Slave Address | Function code | Register start address high | Register start address low | Write register data high bit | Write register data low bit | CRC check high bit | CRC check low bit |
| 01H | 06H | 0C H | 20H | 00H | 00H | 8B | 50H |

## 12. Set user-defined parameters

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | User Parameters | Checksum |
| FA | addr | 43H | USER ID | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | 43H | status<br>(uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

# 4.3 Set performance parameter command

## 4.3.1 Gain parameter

### 1) Position loop gain Kp

The Kp value affects the motor's response speed and regulation ability to position errors.

Kp value can enable the motor to respond quickly to position errors and reduce position tracking errors.

the Kp value is too large, it will cause oscillation or even instability .

### 2) Speed loop gain Kv

The Kv value affects the motor's response speed to speed errors.

Kv value can enable the motor to respond quickly to speed deviations, adjust the speed, and reduce speed errors.

A Kv value that is too large will cause oscillation or even instability, and the speed fluctuation will be aggravated.

### 3) Speed loop integral time constant Tvi

Tvi refers to the time scale parameter of the integral action in the integral link of the speed loop control system.

A smaller Tvi can enable the motor to eliminate steady-state errors faster, but it may cause the system overshoot to increase and even cause system oscillation; a larger Tvi will slow down the process of eliminating steady-state errors, but the system stability is relatively good and the overshoot is smaller.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | | |
|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Byte 10 |
| Frame Header | Slave Address | Function code | Kp | Kv | Tvi | Checksum |
| FA | addr | 70H | Kp | Kv | Tvi | CRC |

| | | | |
|---|---|---|---|
| Kp | range: 0~20000 | Unit: 0.1 Hz | Default value: 80 |
| Kv | range: 1~20000 | Unit: 0.1Hz | Default value: 100 |
| Tvi | range: 15~51200 | Unit: 0.01ms | Default value: 1800 |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 70H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 4.3.2 Rigidity Grade Selection Table

The rigidity level refers to the performance of the motor in maintaining its own motion state and position accuracy when subjected to external forces.

position loop gain and speed loop gain will change the response characteristics and output torque of the motor, thereby affecting the rigidity of the motor.

A higher stiffness level helps improve the position control accuracy of the servo motor, but too high a gain may cause system instability, oscillation or overshoot.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
| Frame Header | Slave Address | Function code | Control Word | Rigidity level | Checksum |
| FA | addr | 71H | Enable | Level | CRC |

Enable = 00 Do not use the stiffness level table, and adjust the gain parameters manually (default value)

Enable = 01 Use the rigidity level table and automatically adjust the gain according to the level

Level range: 0~41 Default value: 10 (the larger the value, the higher the rigidity level)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 71H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

### 4.3.3 Overload protection gain

The motor may be overloaded during operation. Long-term overload will cause the motor to overheat and be damaged.

Overload protection gain (OLP) is a parameter used to set the overload protection startup sensitivity.

OLP = 100 means 10S to start overload protection.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Bytes 5-6 | Byte 7 |
| Frame Header | Slave Address | Function code | Control Word | Protection gain | Checksum |
| FA | addr | 72H | cmd | OLP | CRC |

cmd = 00 Use overload protection (default)

cmd = 01 turns off overload protection, and the motor automatically reduces current when overloaded

OLP range: 10~3000 Default value: 100 Unit: 100ms

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 72H | status<br>(uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

### 4.3.4 Load moment of inertia ratio

The load moment of inertia ratio ( Kj ) refers to the ratio of the load moment of inertia to the motor rotor moment of inertia in the system. It has an important impact on the control accuracy, response speed and stability of the motor.

The calculation formula is $K_j = \frac{J_L}{J_m}$, $J_L$is the moment of inertia of the load,$J_m$ is the moment of inertia of the motor rotor .

The rotor inertia of the SmartMoto60A motor is $0.52\mathrm{Kg} \cdot \mathrm{cm}^2$.

The rotor inertia of the SmartMoto80A motor is $1.48\mathrm{Kg} \cdot \mathrm{cm}^2$.

Kj value, the slower the system response speed is usually and the lower the stability is.

the Kj value is too large, the system will experience unstable phenomena such as oscillation, overshoot, or even loss of control.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Moment of inertia ratio | Checksum |
| FA | addr | 73H | Kj | CRC |

Kj range: 0~12000 Unit: 0.01 times Default value: 0

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 73H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 4.3.5 Position filter time constant

Set the low-pass filter time constant of the position command.

Setting appropriate parameters can reduce motor impact, but excessively large parameters will also increase positioning response delay.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Time constant | Checksum |
| FA | addr | 74H | delayTime | CRC |

delayTime range: 0~65535 unit: 0.1ms default value: 500

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 74H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

### 4.3.6 Position deviation fault threshold

Set the position deviation fault alarm threshold in position mode.

When the deviation between the actual motor position and the command position exceeds this parameter value, a fault alarm will be triggered.

Use encoder units to trigger an alarm when the motor deviates by 1 turn.

Then set the threshold pDeviation = 131072 X 1 = 131072

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Position deviation threshold | Checksum |
| FA | addr | 75H | pDeviation | CRC |

pDeviation (range: 1~ 1073741824 Default value: 1310720)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 75H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

## 4.4 IO port operation instructions

The motor has 3 input ports (DI) and 2 output ports (DO), which can be set with different function options.

### 4.4.1 Input port settings

| Input port DI function option table | |
|---|---|
| **Input function options** | **Functional Description** |
| 0 | invalid |
| 1 | Fault alarm reset |
| 2 | Emergency Stop |
| 3 | reserve |
| 4 | Positive overtravel switch |
| 5 | Reverse overtravel switch |
| 6 | External origin switch ( Hm_Switch ) |
| 7 | Return to zero enable ( Hm_en ) |
| 8 | Motor enable (reserved) |

Note 1: One function option can only be associated with one DI port.

Note 2: For function option "8", when the "85H" instruction is used to set the effective level of the En pin to L or H, DI1 is automatically set to the "motor enable" function. The "21H" instruction cannot be used to set this function.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Byte 9 | Byte 10 |
| Frame Header | Slave Address | Function code | DI1 Function | DI1 logic | DI2 Function | DI2 logic | DI3 Function | DI3 logic | Checksum |
| FA | addr | 21H | Fun1 | Logic1 | Fun2 | Logic2 | Fun3 | Logic3 | CRC |

DIx functions, see the "Input Port DI Function Option Table" above.Funx 00-08: Corresponding to DI function option table function

Funx FF: Keep the original functions without any changes

DIx Logic：

Logicx 00: Indicates that the signal is valid when it is turned on and invalid when it is turned off (default value)

Logicx 01: Indicates that the signal is disconnected and valid, and is not connected.

Logicx FF: Keep the original logic, no changes

Note: The factory default functions of the input ports are as follows

DI1 Motor enable DI2 fault alarm reset DI3 Emergency Stop

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 21H | status (uint8_t) | CRC |

status = 0 Setting failed status = 1 Setting successful

Note: One function option can only be associated with one DI port, otherwise the setting will fail.

## 4.4.2 Output port settings

| Output port DO function option table | |
|---|---|
| **Output function options** | **Functional Description** |
| 0 | User-defined output |
| 1 | Servo motor ready |
| 2 | Fault alarm output |
| 3 | Positioning completion signal output |
| 4 | Torque arrival signal output |
| 5 | Speed arrival signal output |
| 6 | Output when the origin is returned to zero |
| 7 | Electrical zero return completion output |
| 8 | Pulse frequency division output |

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| Frame Header | Slave Address | Function code | DO1 Function | DO1 logic | DO2 Function | DO2 logic | Checksum |
| FA | addr | 22H | Fun1 | Logic1 | Fun2 | Logic2 | CRC |

DOx functions, see the "Output Port DO Function Option Table" above.

Funx 00-08: Corresponding to DO function option table function

Funx FF: Keep the original functions without any changes

DOx Logic

Logicx 00: When the signal is valid, the optocoupler is turned on (default value)

Logicx 01: When the signal is valid, the optocoupler is turned off.

Logicx FF: Keep the original logic, no changes

Note: The factory default functions of the output ports are as follows

DO1 Servo motor ready

DO2 fault alarm output

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 22H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

### 4.4.3 Read IO port status

Read instruction:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | 23H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
| Frame Header | Slave Address | Function code | DO port status | DI port status | Checksum |
| FB | addr | 23H | (uint8_t) | (uint8_t) | CRC |

| DO port status | | | | | | | |
|---|---|---|---|---|---|---|---|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| 0 | 0 | 0 | 0 | 0 | 0 | DO2 | DO1 |

| DI port status | | | | | | | |
|---|---|---|---|---|---|---|---|
| bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| 0 | 0 | 0 | 0 | 0 | DI3 | DI2 | DI1 |

Note: If the port is valid, the corresponding bit is set to 1; if the port is invalid, the corresponding bit is set to 0.

For example, if DO1 is valid and the other DOs are invalid, the DO port status value is 00000001.

### 4.4.4 Write output port

Write output port instruction:

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 |
| Frame Header | Slave Address | Function code | DO1 | DO2 | Checksum |
| FA | addr | 24H | data1 | data2 | CRC |

data = 00 The corresponding port optocoupler is disconnected

data = 01 The corresponding port optocoupler is turned on

data = FF The corresponding port function and status remain unchanged

Note: After writing 00 or 01 to the output port, the port function will automatically change to "user-defined output"

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 24H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 4.5 EEPROM operation instructions

### 4.5.1 Save parameters to EEPROM command

All changed parameters are saved in EEPROM.

Save instructions:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | 42H | 01 | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | 42H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Save failed |
| status = 1 | Saved successfully |
| status = 2 | Data error |

Note: If the data is returned as "Data Error", you need to reset and restart, then save again.

It is best to save data when the motor is disabled to avoid data errors.

### 4.5.2 Restore factory parameters command

Save instructions:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | 3FH | mode | CRC |

| | |
|---|---|
| mode = 0 | restores only user parameters (recommended) |
| mode = 1 | restores user parameters and system parameters |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | 3FH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Recovery failed |
| status = 1 | Recovery successful |

Note: After restoring the factory parameters, the motor will restart automatically.

## 4.6 Reset control instructions

Reset instruction:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Reset Mode | Checksum |
| FA | addr | 41H | Mode | CRC |

Mode = 01 Software reset, restart the motor

Mode = 02 Fault alarm reset, do not restart the motor

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 41H | status (uint8_t) | CRC |

status = 0 Reset failed

status = 1 Reset successful

## 4.7 Read configuration parameter instructions

If you need to read a configured parameter value, add the control word 00H before the function code corresponding to the parameter.

The read instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Control Word | Function code | Checksum |
| FA | addr | 00H | code | CRC |

code is the function code corresponding to the parameter to be read

(For example, to read the "working mode" parameter value, the corresponding function code is "82H")

Return data format:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4-n | Byte n+1 |
| Frame Header | Slave Address | Function code | Parameter Value | Checksum |
| FB | addr | code | param | CRC |

param : corresponding parameter value

Note: The param data format should be consistent with the data format when setting this parameter.

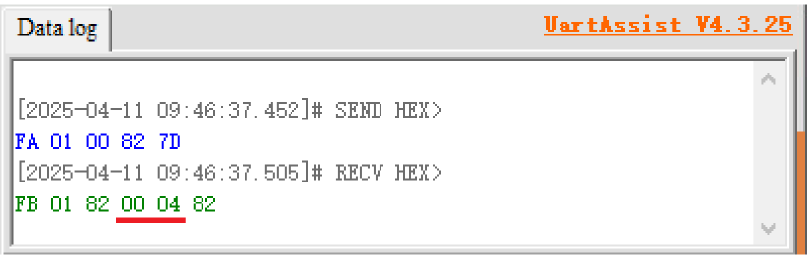
If the parameter does not support reading, the returned data is as follows:

| Uplink frame (PC ← SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
| Frame Header | Slave Address | Function code | | | Checksum |
| FB | addr | code | FFH | FFH | CRC |

For example, read "working mode (code = 82H)"

Read instruction FA 01 00 82 CRC

Return data FB 01 82 00 04 CRC



```
[2025-04-11 09:46:37.452]# SEND HEX>
FA 01 00 82 7D
[2025-04-11 09:46:37.505]# RECV HEX>
FB 01 82 00 04 82
```

## 4.8 IAP firmware upgrade

For IAP upgrade instructions, please refer to "MKS SmartMotor IAP Upgrade Instructions.pdf"

For the IAP upgrade operation video, please refer to "MKS SmartMotor IAP upgrade operation video.mp4"

The firmware upgrade instructions are as follows:

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
|---|---|---|---|---|
| Frame Header | address | Function code | Control Word | Checksum |
| FA | addr | 50 | cmd | CRC |

| | |
|---|---|
| cmd = 01 | to enter boot mode |
| cmd = 02 | Enter silent mode |
| cmd = 03 | exit silent mode |

Control word cmd description:

When there is only one motor on the bus, directly send cmd = 01 command to enter boot mode;

When there are multiple motors on the bus, to avoid data interference, you can do the following:

a. First, broadcast cmd = 02 command (FA 00 50 02 CRC) to make all motors enter silent state;

b. Then send cmd = 01 command to the motor to be upgraded to enter boot mode and upgrade the firmware;

c. After the upgrade is complete, broadcast cmd = 03 command (FA 00 50 03 CRC) to make all motors exit the silent state.

Note 1: In silent mode, the motor does not respond to commands other than 50.

If the setting fails, FB 01 50 00 CRC is returned.

If the setting is successful, FB 01 50 01 CRC is returned.

Note 2: After cmd = 01 is successful, the motor automatically restarts and enters boot mode, waiting to receive the upgrade file.

# Part5. Motor zero return instructions

zero return method is divided into two categories: "origin zero return" and "coordinate zero return".

Note 1: The motor return to zero function is only valid in "position mode".

Note 2: The motor return to zero function is executed only when the motor is enabled.

## 5.1 Description of origin return method

There are two types of “origin return”: “switch origin return” and “mechanical limit origin return”.

### 5.1.1 Switch origin return to zero

You need to first use the 21H instruction to set the function and logic of the corresponding port.

The switch can be selected from "external origin switch", "forward overtravel switch" and "reverse overtravel switch".

The zero return process is as follows:

a. The motor first searches for the switch in the set "direction" and "high speed " ;
b. When encountering the rising edge of the switch signal , it starts to run at "low speed " and disengage the switch;
c. When the motor reaches the falling edge of the switch signal, it commutates and continues to search for the rising edge of the switch signal at a "low speed " ;
d. When the switch signal rises, it runs to the preset "origin offset" position and stops;
e. Mark the current position as "coordinate zero point" and the origin return is successful.

### 5.1.2 Mechanical limit origin return to zero

The zero return torque for the mechanical limit return to zero must be set in advance through instruction 94H . The set torque can drive the load and should not be too large to avoid damaging the equipment.

The zero return process is as follows:

a. searches for the mechanical limit position with the set "direction", "torque " and "low speed " ;
b. When encountering a mechanical limit, it will stop and then run to the preset "origin offset" position and stop;
c. Mark the current position as "coordinate zero point" and the origin return is successful.

## 5.2 Coordinate return to zero method description

Coordinate zero return includes "direct coordinate zero return" and "single-turn coordinate zero return"

### 5.2.1 Direct coordinate return to zero

To directly return the coordinates to zero, you need to first execute the "origin return" function to determine the "coordinate zero point".

The zero return process is as follows:

No need for searching process, it can directly run to the "coordinate zero point" position at high speed and return to zero successfully.

### 5.2.2 Single-turn coordinate return to zero

The "coordinate zero point" within a single circle must be set in advance through instruction 92H.

The single-turn zero return direction must be set in advance through command 93H: "Forward", "Reverse", or "Nearest".

The zero return process is as follows:

a. The motor returns to the "coordinate zero point" position within the preset single circle at "high speed" in the set zero return direction;

b. After arriving, the current position will be cleared and the single-circle coordinate will return to zero successfully.

## 5.3 Set the parameters related to zero return

### 1. Set the return to zero mode, direction, trigger mode, and timeout parameters

The setting instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Bytes 7-8 | Byte 9 |
| Frame Header | Slave Address | Function code | Zero return mode | Return to zero direction | Zero return trigger mode | Return to zero timeout ( ms ) | Checksum |
| FA | addr | 93H | HmMode | HmDir | HmTrig | HmTimOut (uint16_t) | CRC |

Hm Mode : Set the return to zero mode

00: External origin switch returns to zero (21H instruction is required to configure 1 DI port as function 6)

01: Positive overtravel switch returns to zero (needs 21H instruction to configure 1 DI port as function 4)

02: Reverse overtravel switch return to zero (need 21H command to configure 1 DI port as function 5)

03: Mechanical limit return to zero

04: Single turn back to zero

FF: Turn off the origin return function (default value)

Hm Dir : Set the return direction

00: Return to zero in positive direction (default value)

01: Reverse return to zero

02: Nearest zero return (only applicable to single-turn zero return)

Hm Trig : Set the zero return trigger mode

00: Command triggers return to zero (default value)

01: IO signal triggers return to zero (21H instruction is required to configure 1 DI port as function 7)

02: Automatically return to zero after power on

Hm TimOut : Set the return to zero timeout (default = 60000 )

If the motor does not reach zero within this time, alarm code "0801H" will be triggered.

Note: If the corresponding function of the required DI port is not configured, the zero return will fail.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 93H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

## 2. Set the origin offset and return to zero torque parameters

The setting instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-9 | Byte 10 |
| Frame Header | Slave Address | Function code | Zero return torque | Origin offset | Checksum |
| FA | addr | 94H | HmTorque<br>(uint16_t) | Orgoffset<br>(int32_t) | CRC |

HmTorque : Set the torque when the mechanical limit returns to zero (range 0~3000, default value = 500)

Orgoffset : Set the offset position after returning to zero (default = 0, command unit)

When Orgoffset = 0, the motor stops after returning to zero.

When Orgoffset ≠0, after the motor returns to zero, it will continue to run to the offset position before stopping.

Note:When returning to zero by mechanical limit, the Orgoffset value (positive value = positive direction, negative value = reverse direction) and the return to zero direction (positive direction, reverse direction) must match (ie, Orgoffset >=0 HmDir =1 or Orgoffset <=0 HmDir =0 ), otherwise the return to zero will fail.

When returning to zero in a single turn, the Orgoffset value is invalid.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 94H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 3. Set the return speed and acceleration parameters

The setting instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | | | |
|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Byte 10 |
| Frame Header | Slave Address | Function code | "High" speed | "Low" speed | Acceleration and deceleration time | Checksum |
| FA | addr | 95H | HiSpeed (uint16_t) | LoSpeed (uint16_t) | AccTim (uint16_t) | CRC |

HiSpeed : Set the "high speed" speed when returning to zero (range 0~3000RPM, default value = 100)

LoSpeed : Set the "low speed" when returning to zero (range 0~100RPM, default = 10)

AccTim : Set the speed change time of the motor 0-1000RPM when returning to zero (range 0~ 200ms , default value = 200)

the AccTim value, the slower the acceleration and deceleration.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 95H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

### 4. Set the current position as zero point command

The setting instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | 92H | 00H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 92H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

## 5. Execute the return to zero command

The execution instructions are as follows:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Execution | Checksum |
| FA | addr | 91H | goZeroMode | CRC |

| | |
|---|---|
| goZeroMode | 00: Execute the “origin return” function |
| | 01: Execute the "coordinate return to zero" function |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 91H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Return to zero failed |
| status = 1 | returns to zero |
| status = 2 | Return to zero completed |
| status = 3 | Return to zero timeout failure |

Note: When executing the 91H instruction, the motor must be enabled ( shaft locked ) first.

## 5.4 Switch origin return configuration example

Taking "external origin switch" as an example, the configuration methods of "forward overtravel switch" and "reverse overtravel switch" are similar.

### 5.4.1 Command triggers return to zero

**1. Configure DI2 function as "external origin switch"**

FA 01 21 FF FF 06 00 FF FF CRC

**2. Set 93H parameters: external origin switch return to zero, forward, command trigger, timeout 60000ms**

FA 01 93 00 00 00 **EA 60** CRC

**3. Set 94H parameters: zero return torque 400, origin offset 0**

FA 01 94 **01 90** 00 00 00 00 CRC

**4. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 **00 32 00 0A 00 C8** CRC

**5. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

**6. Execute origin return**

FA 01 91 00 CRC

If the origin switch DI2 signal is not triggered (the switch is disconnected), the motor operates as follows:

Forward (counterclockwise) high speed operation → switch closed → reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

If the origin switch DI2 signal is triggered (switch closed), the motor runs as follows:

Reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

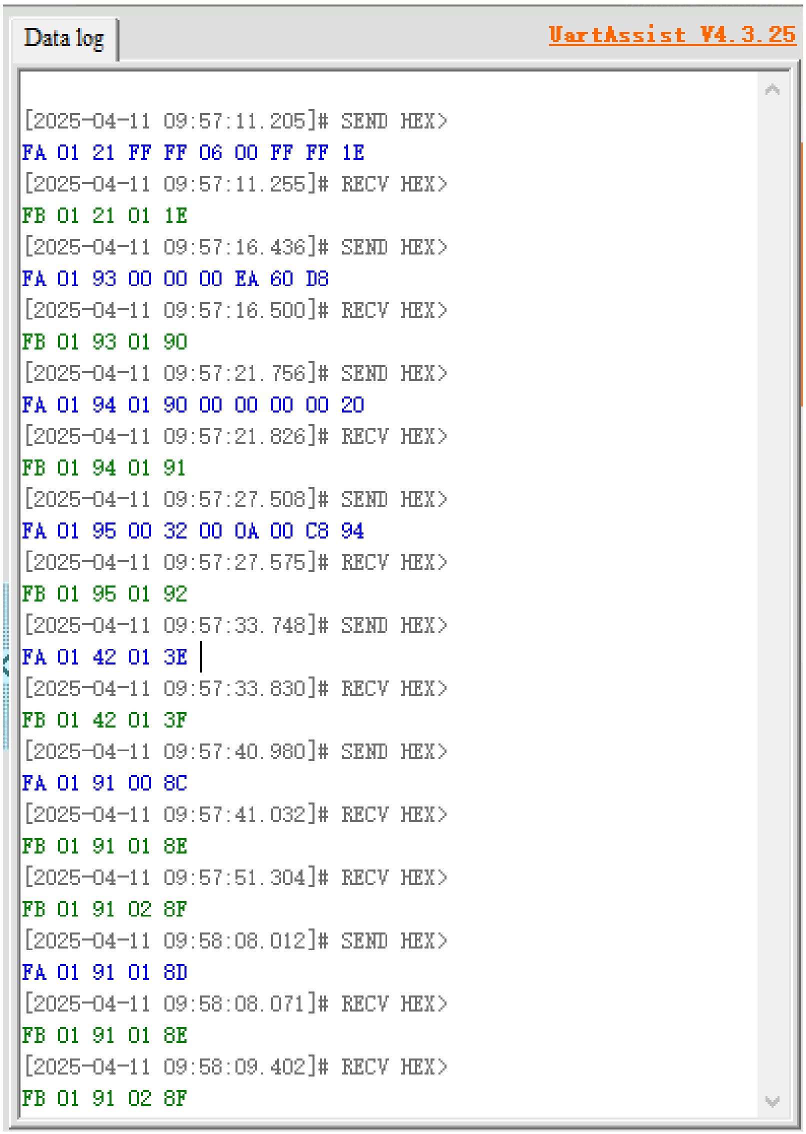
**7. Execute coordinate return to zero**

FA 01 91 01 CRC

You must first execute step 5 to determine the zero point coordinates before you can execute the coordinate return to zero.

If the motor is not at the zero position, it will run directly to the zero position at high speed.

The serial port assistant configuration data is shown in the figure below:



```
[2025-04-11 09:57:11.205]# SEND HEX>
FA 01 21 FF FF 06 00 FF FF 1E
[2025-04-11 09:57:11.255]# RECV HEX>
FB 01 21 01 1E
[2025-04-11 09:57:16.436]# SEND HEX>
FA 01 93 00 00 00 EA 60 D8
[2025-04-11 09:57:16.500]# RECV HEX>
FB 01 93 01 90
[2025-04-11 09:57:21.756]# SEND HEX>
FA 01 94 01 90 00 00 00 00 20
[2025-04-11 09:57:21.826]# RECV HEX>
FB 01 94 01 91
[2025-04-11 09:57:27.508]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 09:57:27.575]# RECV HEX>
FB 01 95 01 92
[2025-04-11 09:57:33.748]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 09:57:33.830]# RECV HEX>
FB 01 42 01 3F
[2025-04-11 09:57:40.980]# SEND HEX>
FA 01 91 00 8C
[2025-04-11 09:57:41.032]# RECV HEX>
FB 01 91 01 8E
[2025-04-11 09:57:51.304]# RECV HEX>
FB 01 91 02 8F
[2025-04-11 09:58:08.012]# SEND HEX>
FA 01 91 01 8D
[2025-04-11 09:58:08.071]# RECV HEX>
FB 01 91 01 8E
[2025-04-11 09:58:09.402]# RECV HEX>
FB 01 91 02 8F
```

### 5.4.2 IO signal triggers return to zero

**1. Configure DI2 function as "external origin switch" and DI3 function as "zero return enable"**

FA 01 21 FF FF 06 00 07 00 CRC

**2. Set 93H parameters: external origin switch return to zero, forward, IO trigger, timeout 60000ms**

FA 01 93 00 00 01 EA 60 CRC

**3. Set 94H parameters: zero return torque 400, origin offset 0**

FA 01 94 01 90 00 00 00 00 CRC

**4. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 00 32 00 0A 00 C8 CRC

**5. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

**6. If the motor is not enabled, enable the motor**

**7. Execute origin return and make DI3 signal valid**

If the switch signal DI2 is not triggered (the switch is open), the motor operates as follows:

Forward (counterclockwise) high speed operation → switch closed → reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

If the switch signal DI2 is triggered (switch closed), the motor runs as follows:

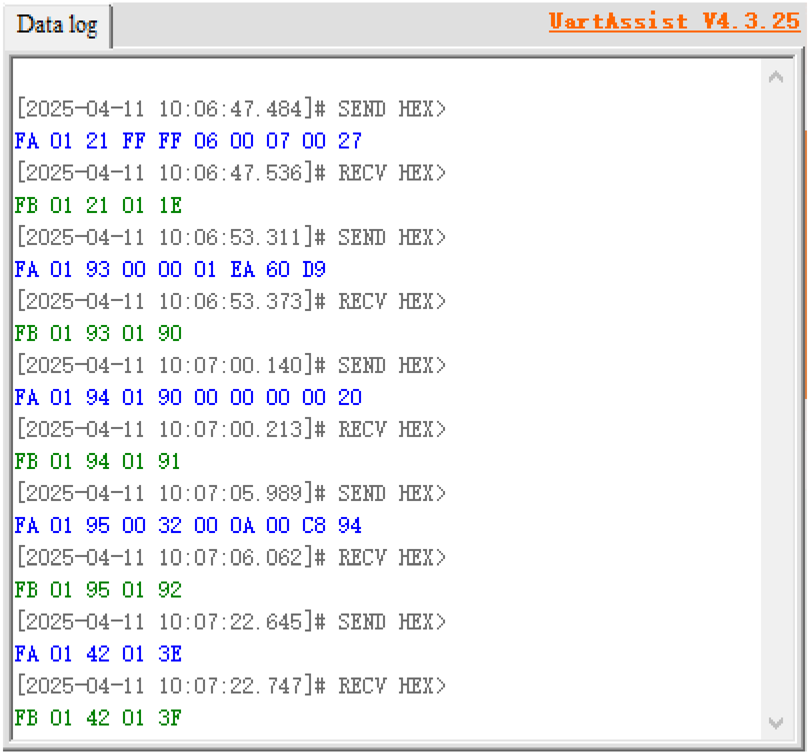
Reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

**8. Execute coordinate return to zero**

Make DI3 signal effective again Or execute instruction： FA 01 91 01 CRC

If the motor is not at the zero position, it will run directly to the zero position at high speed.

The serial port assistant configuration data is shown in the figure below:



```
[2025-04-11 10:06:47.484]# SEND HEX>
FA 01 21 FF FF 06 00 07 00 27
[2025-04-11 10:06:47.536]# RECV HEX>
FB 01 21 01 1E
[2025-04-11 10:06:53.311]# SEND HEX>
FA 01 93 00 00 01 EA 60 D9
[2025-04-11 10:06:53.373]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:07:00.140]# SEND HEX>
FA 01 94 01 90 00 00 00 00 20
[2025-04-11 10:07:00.213]# RECV HEX>
FB 01 94 01 91
[2025-04-11 10:07:05.989]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:07:06.062]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:07:22.645]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:07:22.747]# RECV HEX>
FB 01 42 01 3F
```

### 5.4.3 Automatically return to zero after power on

**1. Configure DI2 function as "external origin switch"**

FA 01 21 FF FF 06 00 FF FF CRC

**2. Set 94H parameters: zero return torque 400, origin offset 0**

FA 01 94 **01 90** 00 00 00 00 CRC

**3. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 **00 32 00 0A 00 C8** CRC

**4. Set 93H parameters: external origin switch return to zero, positive, automatic return to zero after power on, timeout 60000ms**

FA 01 93 00 00 02 **EA 60** CRC

**5. Save Parameters**

FA 01 42 01 CRC

Note 1: The above parameters only need to be set once and do not need to be set repeatedly.

Note 2: After setting the 93H parameter, if the motor is enabled, it will automatically return to zero.

**6. After the motor is turned on and enabled, it will automatically return to zero**

If the origin switch DI2 signal is not triggered (the switch is disconnected), the motor operates as follows:

Forward (counterclockwise) high speed operation → switch closed → reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

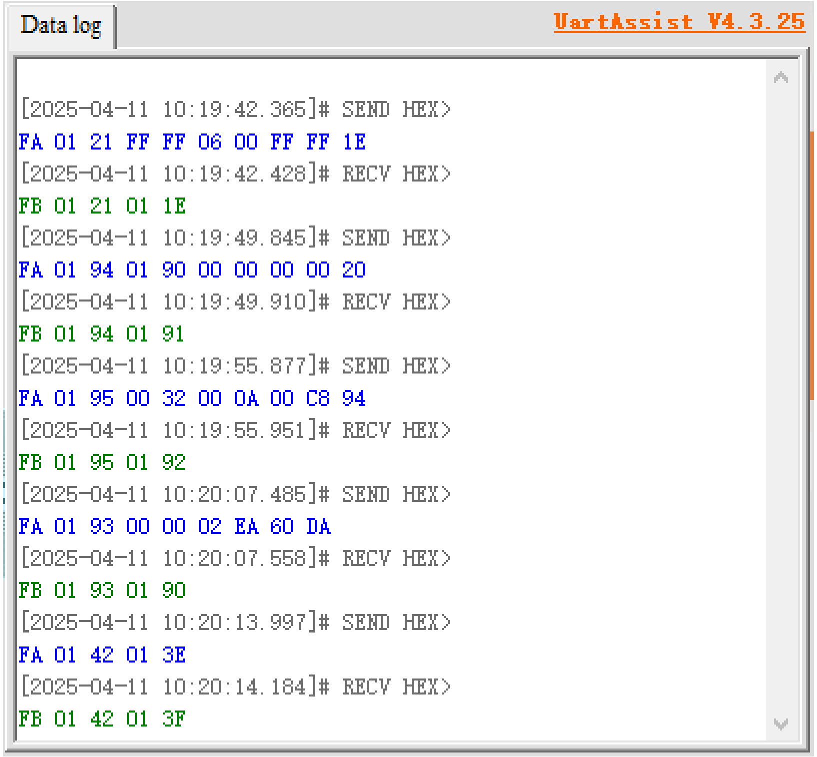
If the origin switch DI2 signal is triggered (switch closed), the motor runs as follows:

Reverse ( clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

**7. Execute coordinate return to zero**

FA 01 91 01 CRC

If the motor is not at the zero position, it will run directly to the zero position at high speed.



```
[2025-04-11 10:19:42.365]# SEND HEX>
FA 01 21 FF FF 06 00 FF FF 1E
[2025-04-11 10:19:42.428]# RECV HEX>
FB 01 21 01 1E
[2025-04-11 10:19:49.845]# SEND HEX>
FA 01 94 01 90 00 00 00 00 20
[2025-04-11 10:19:49.910]# RECV HEX>
FB 01 94 01 91
[2025-04-11 10:19:55.877]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:19:55.951]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:20:07.485]# SEND HEX>
FA 01 93 00 00 02 EA 60 DA
[2025-04-11 10:20:07.558]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:20:13.997]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:20:14.184]# RECV HEX>
FB 01 42 01 3F
```

## 5.5 Mechanical limit origin return configuration example

### 5.5.1 Command trigger return to zero

**1. Set 93H parameters: mechanical limit return to zero, forward, command trigger, timeout 60000ms**

FA 01 93 03 00 00 EA 60 CRC

**2. Set 94H parameters: zero return torque 100, origin offset 0**

FA 01 94 **00 64** 00 00 00 00 CRC

**3. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 **00 32 00 0A 00 C8** CRC

**4. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**5. Execute origin return**

FA 01 91 00 CRC

The motor operates as follows:

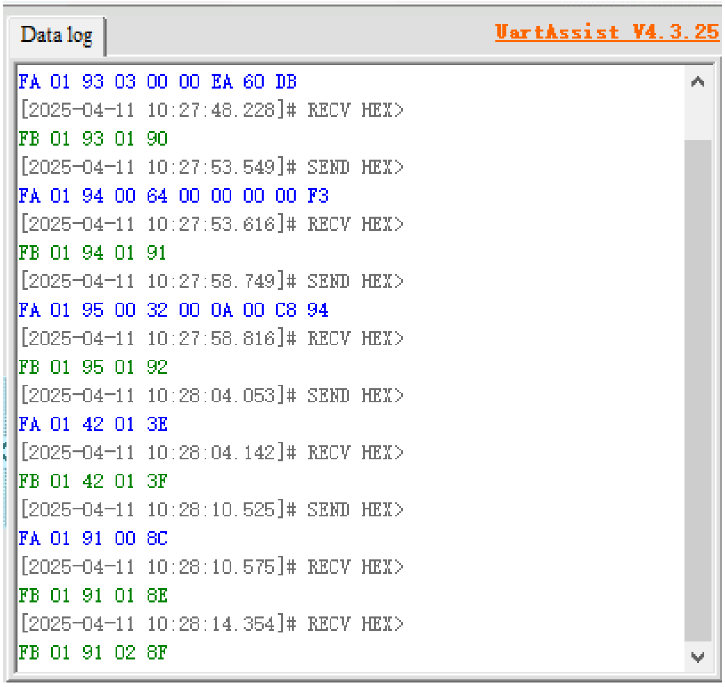
Run forward (counterclockwise) at low speed → reach mechanical limit → stop.

**6. Execute coordinate return to zero**

FA 01 91 01 CRC

You must first execute step 5 to determine the zero point coordinates before you can execute the coordinate return to zero.

If the motor is not at the zero position, it will run directly to the zero position at high speed.



```
FA 01 93 03 00 00 EA 60 DB
[2025-04-11 10:27:48.228]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:27:53.549]# SEND HEX>
FA 01 94 00 64 00 00 00 00 F3
[2025-04-11 10:27:53.616]# RECV HEX>
FB 01 94 01 91
[2025-04-11 10:27:58.749]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:27:58.816]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:28:04.053]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:28:04.142]# RECV HEX>
FB 01 42 01 3F
[2025-04-11 10:28:10.525]# SEND HEX>
FA 01 91 00 8C
[2025-04-11 10:28:10.575]# RECV HEX>
FB 01 91 01 8E
[2025-04-11 10:28:14.354]# RECV HEX>
FB 01 91 02 8F
```

## 5.5.2 IO signal triggers return to zero

**1. Configure DI3 function to "return to zero enable"**

FA 01 21 FF FF FF FF 07 00 CRC

**2. Set 93H parameters: mechanical limit return to zero, positive, IO trigger, timeout 60000ms**

FA 01 93 03 00 01 EA 60 CRC

**3. Set 94H parameters: zero return torque 100, origin offset 0**

FA 01 94 00 64 00 00 00 00 CRC

**4. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 00 32 00 0A 00 C8 CRC

**5. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

**6. If the motor is not enabled, enable the motor**

**7. Execute origin return and make DI3 signal valid**

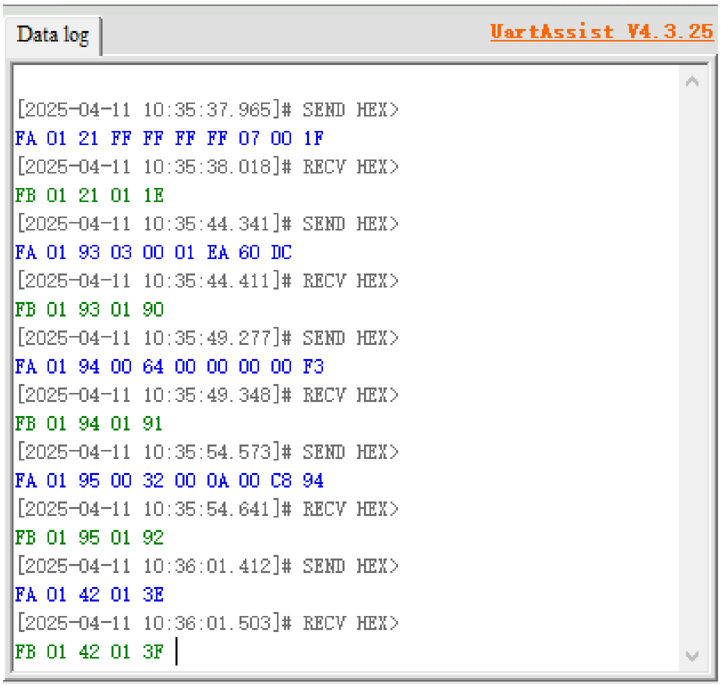
The motor operates as follows:

Run forward (counterclockwise) at low speed → reach mechanical limit → stop.

**8. Execute coordinate return to zero**

Make DI3 signal effective again Or execute instruction： FA 01 91 01 CRC

If the motor is not at the zero position, it will run directly to the zero position at high speed.



```
[2025-04-11 10:35:37.965]# SEND HEX>
FA 01 21 FF FF FF FF 07 00 1F
[2025-04-11 10:35:38.018]# RECV HEX>
FB 01 21 01 1E
[2025-04-11 10:35:44.341]# SEND HEX>
FA 01 93 03 00 01 EA 60 DC
[2025-04-11 10:35:44.411]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:35:49.277]# SEND HEX>
FA 01 94 00 64 00 00 00 00 F3
[2025-04-11 10:35:49.348]# RECV HEX>
FB 01 94 01 91
[2025-04-11 10:35:54.573]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:35:54.641]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:36:01.412]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:36:01.503]# RECV HEX>
FB 01 42 01 3F
```

### 5.5.3 Automatically return to zero after power on

**1. Set 94H parameters: zero return torque 100, origin offset 0**

FA 01 94 00 64 00 00 00 00 CRC

**2. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 00 32 00 0A 00 C8 CRC

**3. Set 93H parameters: Mechanical limit return to zero, positive, automatic return to zero after power on, timeout 60000ms**

FA 01 93 03 00 02 EA 60 CRC

**4. Save Parameters**

FA 01 42 01 CRC

Note 1: The above parameters only need to be set once and do not need to be set repeatedly.

Note 2: After setting the 93H parameters, if the motor is enabled, it will automatically return to zero.

**5. After the motor is turned on and enabled, it automatically returns to zero**

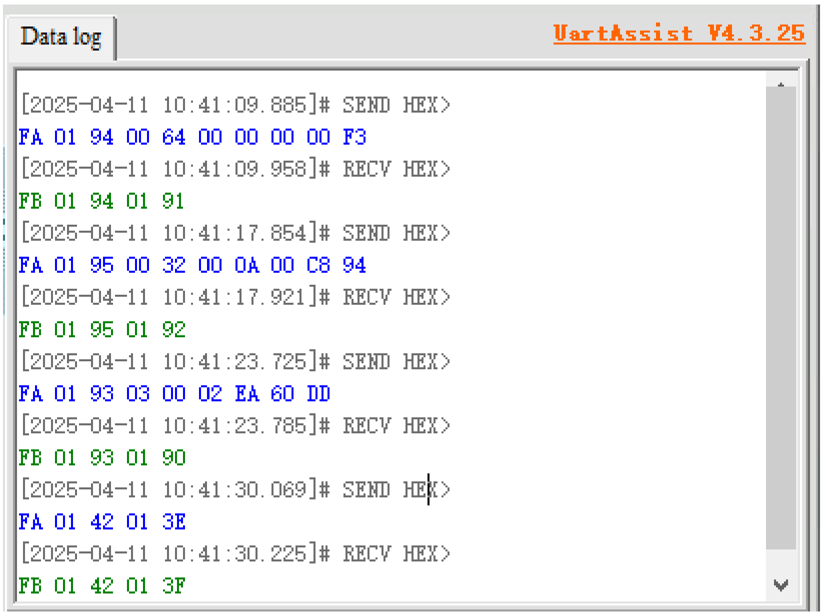
The motor operates as follows:

Run forward (counterclockwise) at low speed → reach mechanical limit → stop.

**6. Execute coordinate return to zero**

FA 01 91 01 CRC

If the motor is not at the zero position, it will run directly to the zero position at high speed.



```
[2025-04-11 10:41:09.885]# SEND HEX>
FA 01 94 00 64 00 00 00 00 F3
[2025-04-11 10:41:09.958]# RECV HEX>
FB 01 94 01 91
[2025-04-11 10:41:17.854]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:41:17.921]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:41:23.725]# SEND HEX>
FA 01 93 03 00 02 EA 60 DD
[2025-04-11 10:41:23.785]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:41:30.069]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:41:30.225]# RECV HEX>
FB 01 42 01 3F
```

## 5.6 Single-turn coordinate return to zero configuration example

### 5.6.1 Command triggers return to zero

**1. Move the motor shaft to the appropriate position and set the zero point coordinate**

FA 01 92 00 CRC

**2. Set 93H parameters: single-turn return to zero, nearest, command trigger, timeout parameter 60000ms**

FA 01 93 04 02 00 EA 60 CRC

**3. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 **00 32 00 0A 00 C8** CRC

**4. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

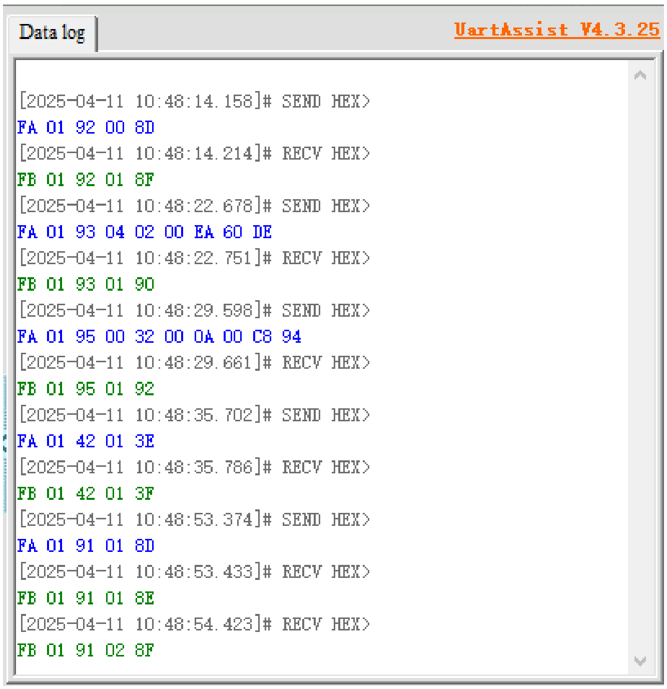
**5. If the motor is not enabled, enable the motor**

**6. Execute single-turn coordinate return to zero**

FA 01 91 01 CRC

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



```
[2025-04-11 10:48:14.158]# SEND HEX>
FA 01 92 00 8D
[2025-04-11 10:48:14.214]# RECV HEX>
FB 01 92 01 8F
[2025-04-11 10:48:22.678]# SEND HEX>
FA 01 93 04 02 00 EA 60 DE
[2025-04-11 10:48:22.751]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:48:29.598]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:48:29.661]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:48:35.702]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:48:35.786]# RECV HEX>
FB 01 42 01 3F
[2025-04-11 10:48:53.374]# SEND HEX>
FA 01 91 01 8D
[2025-04-11 10:48:53.433]# RECV HEX>
FB 01 91 01 8E
[2025-04-11 10:48:54.423]# RECV HEX>
FB 01 91 02 8F
```

### 5.6.2 IO signal triggers return to zero

**1. Move the motor shaft to the appropriate position and set the zero point coordinate**

FA 01 92 00 CRC

**2. Configure DI3 function to "return to zero enable"**

FA 01 21 FF FF FF FF 07 00 CRC

**3. Set 93H parameters: single-turn return to zero, nearest, IO trigger, timeout parameter 60000ms**

FA 01 93 04 02 01 EA 60 CRC

**4. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 00 32 00 0A 00 C8 CRC

**5. Save Parameters**

FA 01 42 01 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

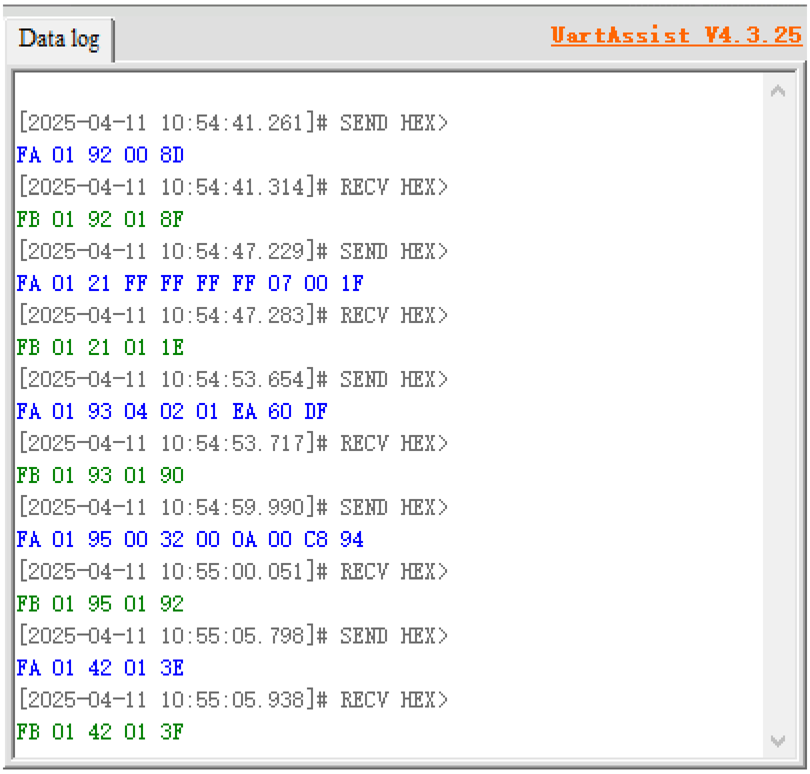
**6. If the motor is not enabled, enable the motor**

**7. Execute single-turn return to zero**

Make DI3 signal effective Or execute instruction： FA 01 91 01 CRC

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



```
[2025-04-11 10:54:41.261]# SEND HEX>
FA 01 92 00 8D
[2025-04-11 10:54:41.314]# RECV HEX>
FB 01 92 01 8F
[2025-04-11 10:54:47.229]# SEND HEX>
FA 01 21 FF FF FF FF 07 00 1F
[2025-04-11 10:54:47.283]# RECV HEX>
FB 01 21 01 1E
[2025-04-11 10:54:53.654]# SEND HEX>
FA 01 93 04 02 01 EA 60 DF
[2025-04-11 10:54:53.717]# RECV HEX>
FB 01 93 01 90
[2025-04-11 10:54:59.990]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 10:55:00.051]# RECV HEX>
FB 01 95 01 92
[2025-04-11 10:55:05.798]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 10:55:05.938]# RECV HEX>
FB 01 42 01 3F
```

### 5.6.3 Automatically return to zero after power on

**1. Move the motor shaft to the appropriate position and set the zero point coordinate**

FA 01 92 00 CRC

**2. Set 93H parameters: single-turn zero return, nearest, automatic zero return on power-up, timeout parameter 60000ms**

FA 01 93 04 02 02 EA 60 CRC

**3. Set 95H parameters: high speed 50, low speed 10, acceleration and deceleration time 200**

FA 01 95 **00 32 00 0A 00 C8** CRC

**4. Save Parameters**

FA 01 42 01 CRC

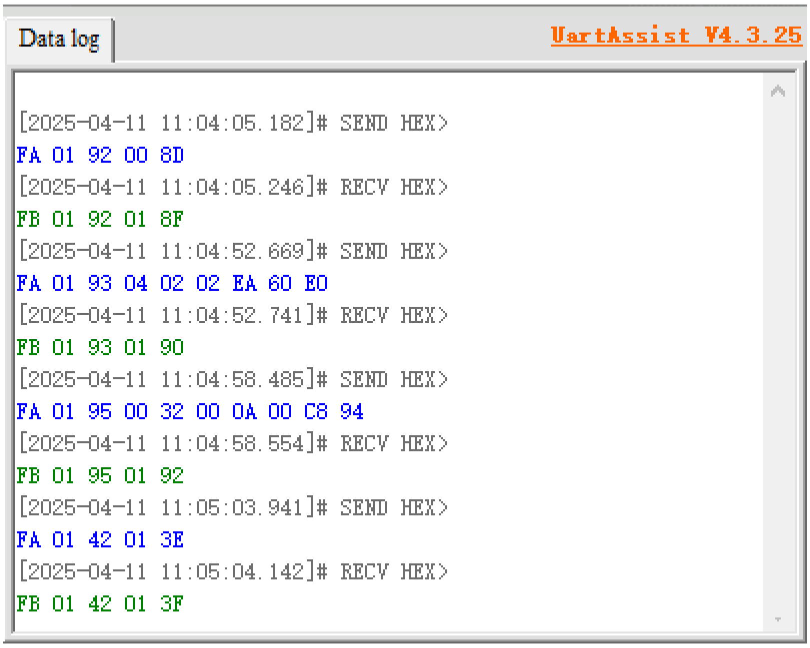
Note: The above parameters only need to be set once and there is no need to set them repeatedly.

**5. After the motor is turned on and enabled, it automatically returns to zero Or execute instruction：**

FA 01 91 01 CRC

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



```
[2025-04-11 11:04:05.182]# SEND HEX>
FA 01 92 00 8D
[2025-04-11 11:04:05.246]# RECV HEX>
FB 01 92 01 8F
[2025-04-11 11:04:52.669]# SEND HEX>
FA 01 93 04 02 02 EA 60 E0
[2025-04-11 11:04:52.741]# RECV HEX>
FB 01 93 01 90
[2025-04-11 11:04:58.485]# SEND HEX>
FA 01 95 00 32 00 0A 00 C8 94
[2025-04-11 11:04:58.554]# RECV HEX>
FB 01 95 01 92
[2025-04-11 11:05:03.941]# SEND HEX>
FA 01 42 01 3E
[2025-04-11 11:05:04.142]# RECV HEX>
FB 01 42 01 3F
```

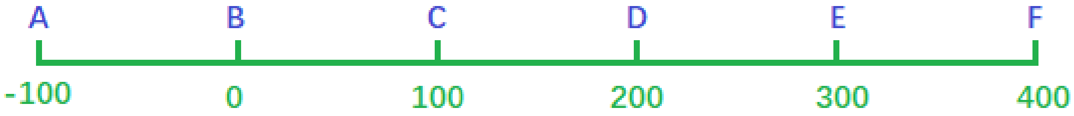
# Part6. Position Control Mode Description

The position control mode uses "pulse unit" as the position counting unit.

Position control modes are differentiated by coordinate position: relative position and absolute position.

Position control mode is divided into two types according to the operation mode: single operation and cyclic operation.

## 6.1 Relative and absolute position description



Assume the current position of the motor is at point C

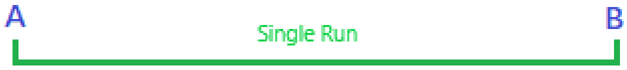
After running 200 at the relative position, the motor reaches point E.

After running at a relative position of -200, the motor reaches point A.

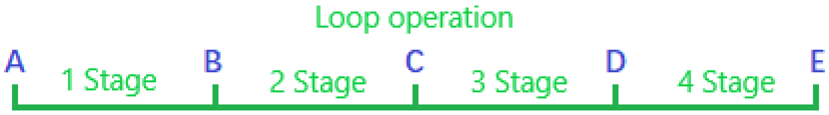
After running 100 at the absolute position, the motor reaches point C.

After running at an absolute position of -100, the motor reaches point A.

## 6.2 Description of single position operation and multi-segment position operation



Single position operation means that the motor runs from point A to point B at the set speed and acceleration, and then stops.



When operating in multiple positions, up to 4 positions can be configured.

Each segment can independently configure the displacement, speed, acceleration, and waiting time parameters after running.

Take 4-stage position operation as an example:

During single multi-stage position operation, the motor operation sequence is:

A→B→C→D→E→Stop.

When the cyclic multi-stage position is running, the motor running sequence is:

A→B→C→D→E→A→B→ … and the cycle continues.

## 6.3 General control instructions

General control instructions, applicable to position control mode, speed control mode and torque control mode.

### 6.3.1 Stop command

Stop command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | F7H | stopMode | CRC |

stopMode = 0 sets normal stop

stopMode = 1 sets emergency stop

stopMode = 2 to release the emergency stop

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | F7H | status (uint8_t) | CRC |

status = 0 Setting failed

status = 1 Setting successful

Note : After the emergency stop is successful, the emergency stop must be released before the motor can be run again.

### 6.3.2 Query the motor running status command

Query command:

| Downlink frame (PC → SmartMotor ) | | | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Frame Header | Slave Address | Function code | Checksum |
| FA | addr | F1H | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | F1H | status (uint8_t) | CRC |

status = 0 Query failed

status = 1 The motor stops running

status = 2 Motor acceleration operation

status = 3 Motor deceleration operation

status = 4 The motor runs at full speed

### 6.3.3 Motor enable command

Enable instruction:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | F3H | enable | CRC |

| | |
|---|---|
| enable = 1 | enables the motor |
| enable = 0 | releases the motor |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | F3H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 6.3.4 Set the power-on automatic run command

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | FFH | cmd | CRC |

| | |
|---|---|
| cmd = 00 | cancels automatic operation at power on |
| cmd = 01 | Set to run automatically after power on |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | FFH | status | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

Note 1: In position control mode, if the power-on automatic return to zero function is set, after power-on, wait for the zero return to complete before executing the power-on automatic operation function.

Note 2: If the motor is running, first use the F7H command to stop the motor, then use the 42H command to save the parameters.

## 6.3.5 Set the threshold value instruction

The threshold value (threshold ) is described as follows:

**1. Position control mode**

Positioning completion threshold

When the motor position deviation is less than thereshold , the positioning completion signal output is valid (DO function option 3).

thereshold Range: 0~65535 Unit: Pulse unit Default value: 100

**2. Speed control mode**

The speed reaches the threshold

When the actual motor speed is greater than or equal to the reshold , the speed arrival signal output is valid (DO function option 5).

thereshold Range: 10~3000 Unit: RPM Default value: 1000

**3. Torque control mode**

That is, the torque reaches the threshold

When the actual motor torque is greater than or equal to t hreshold , the torque arrival signal output is valid (DO function option 4).

thereshold Range: 0~3000 Unit: 0.1% rated torque Default value: 200

Note: When the control mode is changed, thereshold will automatically return to the default value of that control mode.

Set the command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Byte 6 |
| Frame Header | Slave Address | Function code | Threshold | Checksum |
| FA | addr | F2H | thereshold | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | state | Checksum |
| FB | addr | F2H | status | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 6.4 Position control configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
| Frame Header | Slave Address | Function code | How it works | Displacement method | Displacement Segment Number | Residue processing method | Checksum |
| FA | addr | 96H | runMode | shiftMode | shiftNum | remainMode | CRC |

runMode : 00 single position run (default value)
01 Single multi-stage position operation
02 Cyclic multi-segment position operation

shiftMode : 00 relative shift (default)
01 Absolute displacement
shiftNum : 01~04 shift segment number (default value 01)

remianMode : 00 The loop operation is paused, and after restarting, it continues to run the remaining number of segments (default value)
01 The cycle operation is paused, and after restarting, it starts from the first section

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 96H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 6.5 Single position run

### 6.5.1 Single position operation command

Run command:

| Downlink frame (PC → SmartMotor ) | | | | | | |
|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-11 | Byte 12 |
| Frame Header | Slave Address | Function code | Maximum speed | Acceleration and deceleration time | Displacement | Checksum |
| FA | addr | FDH | Speed(uint16_t) | accTime (uint16_t) | pulses(int32_t) | CRC |

Speed: Maximum speed (range 1~3000, unit: RPM)

accTime : acceleration/deceleration time (range 1~65535, unit: ms )

pulses: displacement (range -10000000~10000000, unit: pulse unit)

Note 1: accTime is the acceleration time from 0 to 1000RPM and the deceleration time from 1000 to 0RPM, the same below.

Note 2: pulses are positive/negative numbers that determine the direction of motor operation.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Running status | Checksum |
| FB | addr | FDH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Run failed |
| status = 1 | Run started |
| status = 2 | Run completed |

Note 3: You can use the command "8CH" to set whether to return to the running state.

### 6.5.2 Single relative position run configuration example

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 **0C 80** CRC

**3. Set 96H parameters: single position operation, relative displacement, displacement segment number**

FA 01 96 00 00 01 01 CRC

**4. Save Parameters**

FA 01 42 01 CRC

**5. Coordinates cleared for easy observation (optional)**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and there is no need to set them repeatedly.

**6. Control motor operation and set FDH parameters**

Maximum speed **200**, acceleration/deceleration time **100**ms, displacement **32000** (10 turns)

FA 01 FD **00 C8 00 64 00 00 7D 00** CRC

**7. The motor starts running.**

**8. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**9. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

FA 01 F1 CRC command can check whether the motor is in place

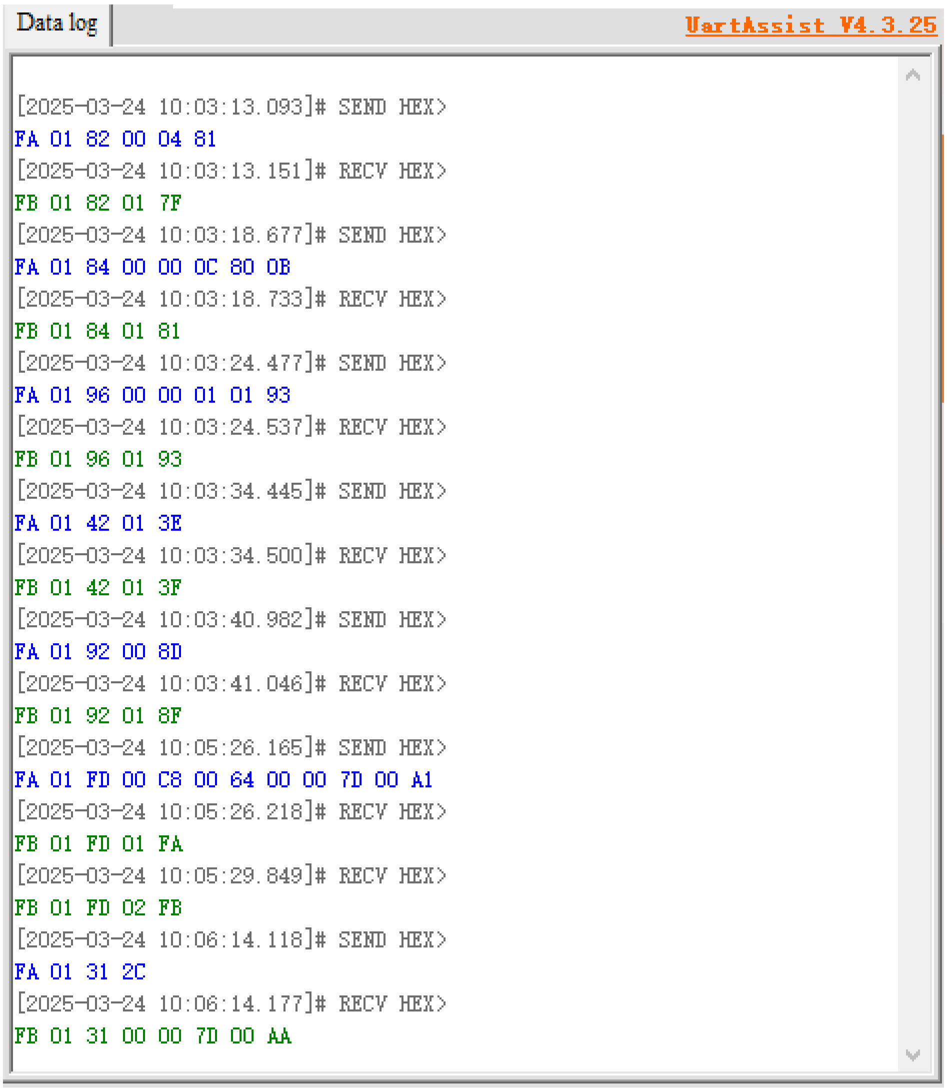
If you need to run automatically after power on, add the following configuration:

**10. Set automatic operation on power on**

FA 01 FF 01 CRC

**11. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-24 10:03:13.093]# SEND HEX>
FA 01 82 00 04 81
[2025-03-24 10:03:13.151]# RECV HEX>
FB 01 82 01 7F
[2025-03-24 10:03:18.677]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-24 10:03:18.733]# RECV HEX>
FB 01 84 01 81
[2025-03-24 10:03:24.477]# SEND HEX>
FA 01 96 00 00 01 01 93
[2025-03-24 10:03:24.537]# RECV HEX>
FB 01 96 01 93
[2025-03-24 10:03:34.445]# SEND HEX>
FA 01 42 01 3E
[2025-03-24 10:03:34.500]# RECV HEX>
FB 01 42 01 3F
[2025-03-24 10:03:40.982]# SEND HEX>
FA 01 92 00 8D
[2025-03-24 10:03:41.046]# RECV HEX>
FB 01 92 01 8F
[2025-03-24 10:05:26.165]# SEND HEX>
FA 01 FD 00 C8 00 64 00 00 7D 00 A1
[2025-03-24 10:05:26.218]# RECV HEX>
FB 01 FD 01 FA
[2025-03-24 10:05:29.849]# RECV HEX>
FB 01 FD 02 FB
[2025-03-24 10:06:14.118]# SEND HEX>
FA 01 31 2C
[2025-03-24 10:06:14.177]# RECV HEX>
FB 01 31 00 00 7D 00 AA
```

## 6.5.3 Single absolute position run configuration example

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 **0C 80** CRC

**3. Set 96H parameters: single position operation, absolute displacement, displacement segment number**

FA 01 96 00 01 01 01 CRC

**4. Save Parameters**

FA 01 42 01 CRC

**5. Coordinates cleared for easy observation**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**6. Control motor operation and set FDH parameters**

Maximum speed **200**, acceleration/deceleration time **100**ms, displacement **32000** (10 turns)

FA 01 FD **00 C8 00 64 00 00 7D** 00 CRC

**7. The motor starts running.**

**8. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**9. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

FA 01 F1 CRC command can check whether the motor is in place

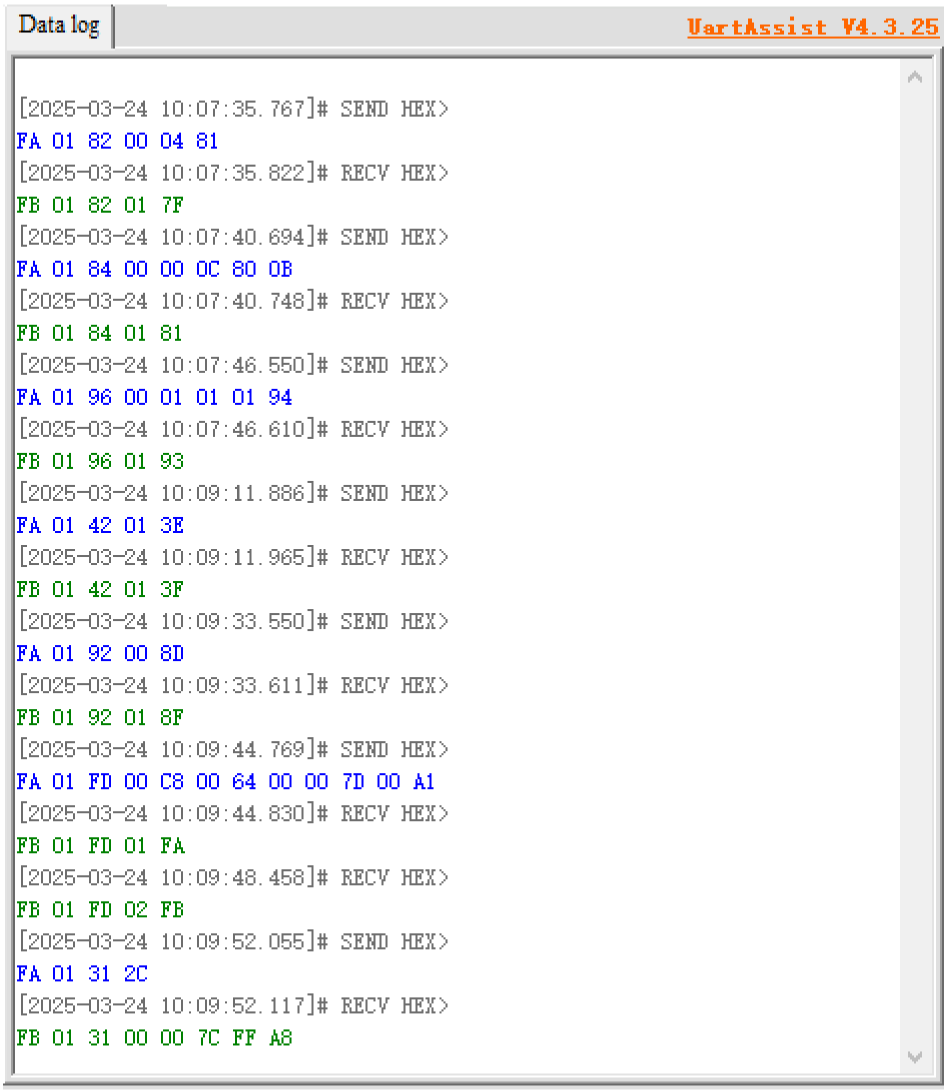
If you need to run automatically after power on, add the following configuration:

**10. Set automatic operation on power on**

FA 01 FF 01 CRC

**11. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-24 10:07:35.767]# SEND HEX>
FA 01 82 00 04 81
[2025-03-24 10:07:35.822]# RECV HEX>
FB 01 82 01 7F
[2025-03-24 10:07:40.694]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-24 10:07:40.748]# RECV HEX>
FB 01 84 01 81
[2025-03-24 10:07:46.550]# SEND HEX>
FA 01 96 00 01 01 01 94
[2025-03-24 10:07:46.610]# RECV HEX>
FB 01 96 01 93
[2025-03-24 10:09:11.886]# SEND HEX>
FA 01 42 01 3E
[2025-03-24 10:09:11.965]# RECV HEX>
FB 01 42 01 3F
[2025-03-24 10:09:33.550]# SEND HEX>
FA 01 92 00 8D
[2025-03-24 10:09:33.611]# RECV HEX>
FB 01 92 01 8F
[2025-03-24 10:09:44.769]# SEND HEX>
FA 01 FD 00 C8 00 64 00 00 7D 00 A1
[2025-03-24 10:09:44.830]# RECV HEX>
FB 01 FD 01 FA
[2025-03-24 10:09:48.458]# RECV HEX>
FB 01 FD 02 FB
[2025-03-24 10:09:52.055]# SEND HEX>
FA 01 31 2C
[2025-03-24 10:09:52.117]# RECV HEX>
FB 01 31 00 00 7C FF A8
```

## 6.6 Multi- position operation

### 6.6.1 Bit field parameter configuration instructions

**1. Section 1 parameter configuration instructions**

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Bytes 10-13 | Byte 14 |
| Frame Header | Slave Address | Function code | Maximum speed | Acceleration and deceleration time | Waiting time after the run is completed | Displacement | Checksum |
| FA | addr | 97H | speed1 | accTime1 | waitTime1 | Pulses1 | CRC |

Speed 1: Maximum speed (range 1~3000, unit: RPM)

accTime 1: Acceleration and deceleration time (range 1~65535, unit: ms )

waitTime 1: Waiting time after the operation is completed (range 0~10000, unit: ms )

pulses1: displacement (range -10000000~10000000, unit: pulse unit)

Note 1: accTime is the acceleration time from 0 to 1000RPM and the deceleration time from 1000 to 0RPM, and the other segments are the same.

Note 2: relPulses is a positive/negative number that determines the direction of motor operation. The other segments are the same.

Note 3: The above parameters can be modified during operation. The parameters will take effect the next time the operation reaches this section. The same applies to other sections.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame | Slave | Function | Set Status | Checksum |
| FB | addr | 97H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 2. Section 2 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Bytes 10-13 | Byte 14 |
| Frame Header | Slave Address | Function code | Maximum speed | Acceleration and deceleration time | Waiting time after the run is completed | Displacement | Checksum |
| FA | addr | 99H | speed2 | accTime2 | waitTime2 | pulses2 | CRC |

speed 2: maximum speed(range 1~3000, unit: RPM)

accTime2: acceleration and deceleration time (range 1~65535, unit: ms )

waitTime2: Waiting time after the operation is completed (range 0~10000, unit: ms )

pulses2: displacement (range -10000000~10000000, unit: pulse unit)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 99H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 3. Section 3 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Bytes 10-13 | Byte 14 |
| Frame Header | Slave Address | Function code | Maximum speed | Acceleration and deceleration time | Waiting time after the run is completed | Displacement | Checksum |
| FA | addr | 9BH | speed3 | accTime3 | waitTime3 | pulses3 | CRC |

speed 3: maximum speed (range 1~3000, unit: RPM)

accTime 3: Acceleration and deceleration time (range 1~65535, unit: ms )

waitTime 3: Waiting time after the operation is completed (range 0~10000, unit: ms )

pulses3: displacement (range -10000000~10000000, unit: pulse unit)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 9BH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 4. Section 4 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Bytes 10-13 | Byte 14 |
| Frame Header | Slave Address | Function code | Maximum speed | Acceleration and deceleration time | Waiting time after the run is completed | Displacement | Checksum |
| FA | addr | 9DH | speed4 | accTime4 | waitTime4 | pulses4 | CRC |

speed 4: Maximum speed (range 1~3000, unit: RPM)

accTime 4: Acceleration and deceleration time (range 1~65535, unit: ms )

waitTime 4: Waiting time after the operation is completed (range 0~10000, unit: ms )

pulses4: displacement(range -10000000~10000000, unit: pulse unit)

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 9DH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 6.6.2 Multi-segment position operation command

Run command:

| Downlink frame (PC → SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Control Word | Checksum |
| FA | addr | FEH | 01 | CRC |

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Running status | Checksum |
| FB | addr | FEH | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Run failed |
| status = 1 | Run started |

Note: You can use the command "8CH" to set whether to return to the running state.

### 6.6.3 single multi-segment relative position operation

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 **0C 80** CRC

**3. Set 96H parameters: single multi-stage position operation, relative displacement, 4-stage displacement**

FA 01 96 01 00 04 01 CRC

**4. Set 97H parameters: Parameters for the first section ( 200 RPM, 1000 ms, 2000 ms, 32000 pulses)**

FA 01 97 **00 C8 03 E8 07 D0 00 00 7D 00** CRC

**5. Set 99H parameters: Parameters for the second section ( 300 RPM, 100 ms, 2000 ms, -32000 pulses)**

FA 01 99 **01 2C 00 64 07 D0 FF FF 83 00** CRC

**6. Set 9BH parameters: 3rd section parameters ( 600 RPM, 2000 ms, 500 ms, 128000 pulses)**

FA 01 9B **02 58 07 D0 01 F4 00 01 F4 00** CRC

**7. Set 9DH parameters: 4th section parameters ( 1000 RPM, 100 ms, 10 ms, -128000 pulses)**

FA 01 9D **03 E8 00 64 00 0A FF FE 0C 00** CRC

**8. Save Parameters**

FA 01 42 01 CRC

**9. Coordinates cleared for easy observation (optional)**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**10. Control motor operation**

FA 01 FE 01 CRC

**11. The motor starts running.**

**12. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**13. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

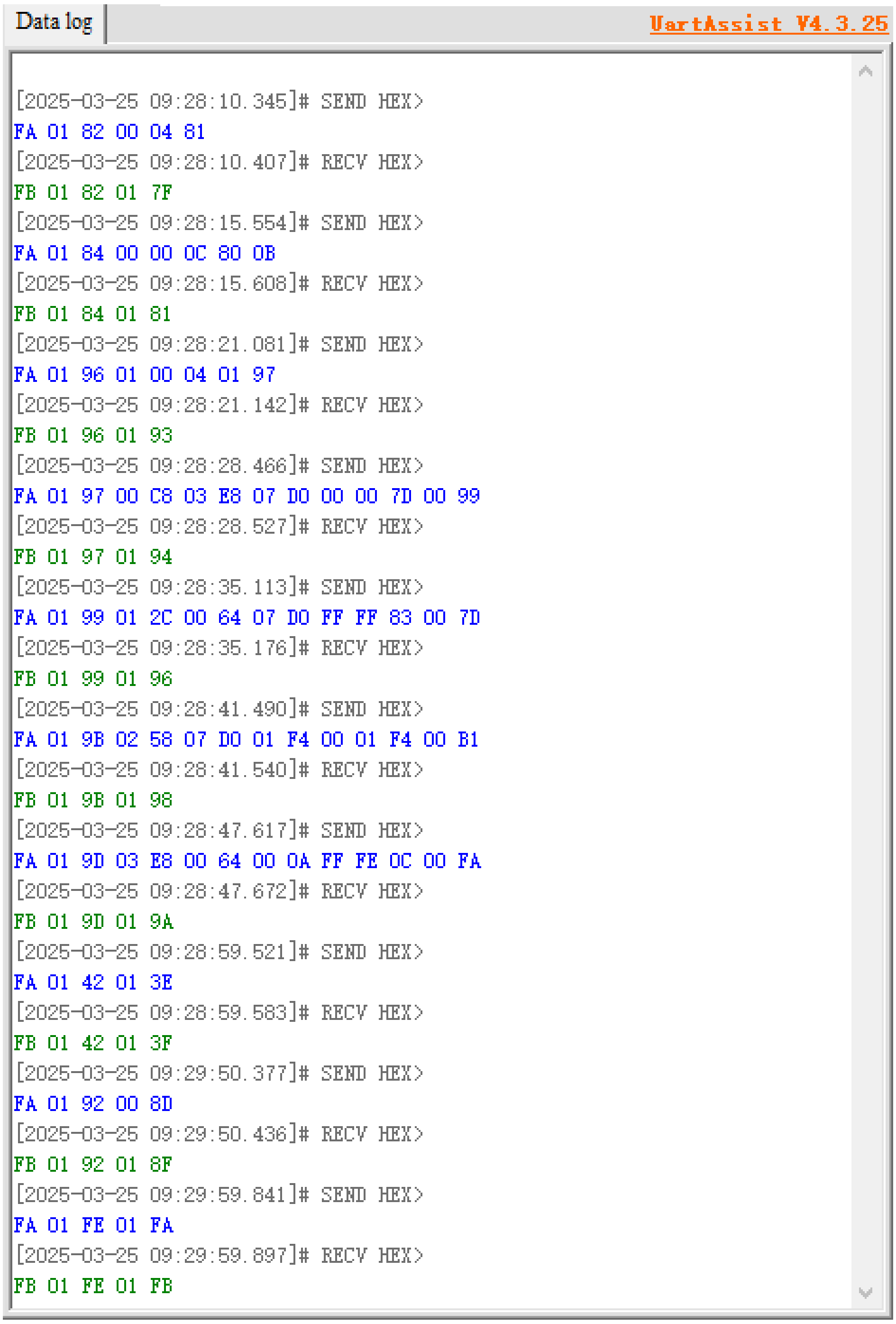
If you need to run automatically after power on, add the following configuration:

**14. Set automatic operation on power on**

FA 01 FF 01 CRC

**15. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-25 09:28:10.345]# SEND HEX>
FA 01 82 00 04 81
[2025-03-25 09:28:10.407]# RECV HEX>
FB 01 82 01 7F
[2025-03-25 09:28:15.554]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-25 09:28:15.608]# RECV HEX>
FB 01 84 01 81
[2025-03-25 09:28:21.081]# SEND HEX>
FA 01 96 01 00 04 01 97
[2025-03-25 09:28:21.142]# RECV HEX>
FB 01 96 01 93
[2025-03-25 09:28:28.466]# SEND HEX>
FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 99
[2025-03-25 09:28:28.527]# RECV HEX>
FB 01 97 01 94
[2025-03-25 09:28:35.113]# SEND HEX>
FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 7D
[2025-03-25 09:28:35.176]# RECV HEX>
FB 01 99 01 96
[2025-03-25 09:28:41.490]# SEND HEX>
FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 B1
[2025-03-25 09:28:41.540]# RECV HEX>
FB 01 9B 01 98
[2025-03-25 09:28:47.617]# SEND HEX>
FA 01 9D 03 E8 00 64 00 0A FF FE 0C 00 FA
[2025-03-25 09:28:47.672]# RECV HEX>
FB 01 9D 01 9A
[2025-03-25 09:28:59.521]# SEND HEX>
FA 01 42 01 3E
[2025-03-25 09:28:59.583]# RECV HEX>
FB 01 42 01 3F
[2025-03-25 09:29:50.377]# SEND HEX>
FA 01 92 00 8D
[2025-03-25 09:29:50.436]# RECV HEX>
FB 01 92 01 8F
[2025-03-25 09:29:59.841]# SEND HEX>
FA 01 FE 01 FA
[2025-03-25 09:29:59.897]# RECV HEX>
FB 01 FE 01 FB
```

### 6.6.4 Single multi-segment absolute position operation configuration example

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 **0C 80** CRC

**3. Set 96H parameters: single multi-stage position operation, absolute displacement, 4-stage displacement**

FA 01 96 01 01 04 01 CRC

**4. Set 97H parameters: Parameters for the first section ( 200 RPM, 1000 ms, 2000 ms, 32000 pulses)**

FA 01 97 **00 C8 03 E8 07 D0 00 00 7D 00** CRC

**5. Set 99H parameters: Parameters for the second section ( 300 RPM, 100 ms, 2000 ms, -32000 pulses)**

FA 01 99 **01 2C 00 64 07 D0 FF FF 83 00** CRC

**6. Set 9BH parameters: 3rd section parameters ( 600 RPM, 2000 ms, 500 ms, 128000 pulses)**

FA 01 9B **02 58 07 D0 01 F4 00 01 F4 00** CRC

**7. Set 9DH parameters: 4th section parameters ( 1000 RPM, 100 ms, 10 ms, 0 pulses)**

FA 01 9D **03 E8 00 64 00 0A 00 00 00 00** CRC

**8. Save Parameters**

FA 01 42 01 CRC

**9. Coordinates cleared for easy observation (optional)**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**10. Control motor operation**

FA 01 FE 01 CRC

**11. The motor starts running.**

**12. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**13. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

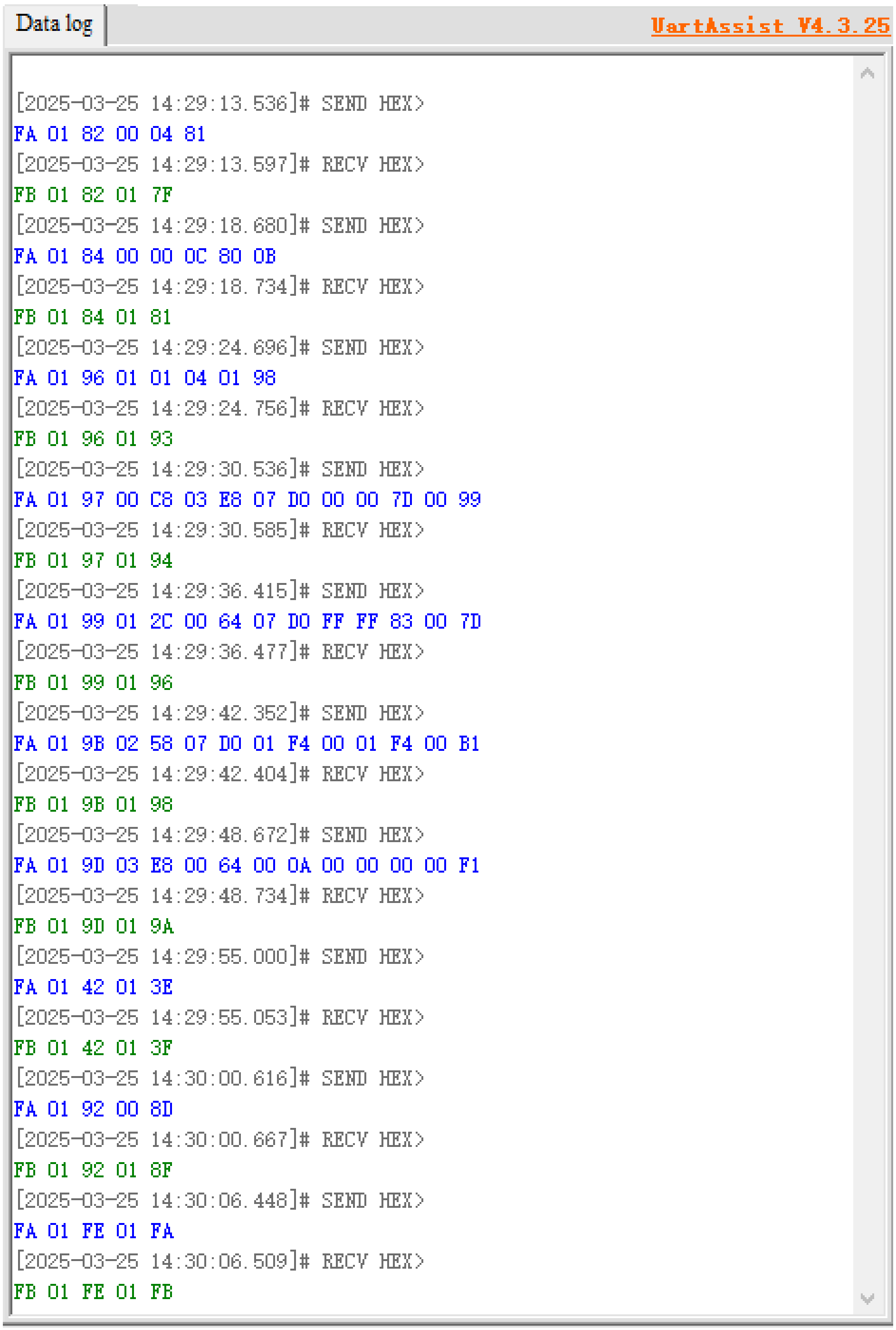
If you need to run automatically after power on, add the following configuration:

**14. Set automatic operation on power on**

FA 01 FF 01 CRC

**15. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-25 14:29:13.536]# SEND HEX>
FA 01 82 00 04 81
[2025-03-25 14:29:13.597]# RECV HEX>
FB 01 82 01 7F
[2025-03-25 14:29:18.680]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-25 14:29:18.734]# RECV HEX>
FB 01 84 01 81
[2025-03-25 14:29:24.696]# SEND HEX>
FA 01 96 01 01 04 01 98
[2025-03-25 14:29:24.756]# RECV HEX>
FB 01 96 01 93
[2025-03-25 14:29:30.536]# SEND HEX>
FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 99
[2025-03-25 14:29:30.585]# RECV HEX>
FB 01 97 01 94
[2025-03-25 14:29:36.415]# SEND HEX>
FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 7D
[2025-03-25 14:29:36.477]# RECV HEX>
FB 01 99 01 96
[2025-03-25 14:29:42.352]# SEND HEX>
FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 B1
[2025-03-25 14:29:42.404]# RECV HEX>
FB 01 9B 01 98
[2025-03-25 14:29:48.672]# SEND HEX>
FA 01 9D 03 E8 00 64 00 0A 00 00 00 00 F1
[2025-03-25 14:29:48.734]# RECV HEX>
FB 01 9D 01 9A
[2025-03-25 14:29:55.000]# SEND HEX>
FA 01 42 01 3E
[2025-03-25 14:29:55.053]# RECV HEX>
FB 01 42 01 3F
[2025-03-25 14:30:00.616]# SEND HEX>
FA 01 92 00 8D
[2025-03-25 14:30:00.667]# RECV HEX>
FB 01 92 01 8F
[2025-03-25 14:30:06.448]# SEND HEX>
FA 01 FE 01 FA
[2025-03-25 14:30:06.509]# RECV HEX>
FB 01 FE 01 FB
```

### 6.6.5 Example of loop multi-segment relative position operation configuration

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 0C 80 CRC

**3. Set 96H parameters: cyclic multi-stage position operation, relative displacement, 4-stage displacement**

FA 01 96 02 00 04 01 CRC

**4. Set 97H parameters: Parameters for the first section ( 200 RPM, 1000 ms, 2000 ms, 32000 pulses)**

FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 CRC

**5. Set 99H parameters: Parameters for the second section ( 300 RPM, 100 ms, 2000 ms, -32000 pulses)**

FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 CRC

**6. Set 9BH parameters: 3rd section parameters ( 600 RPM, 2000 ms, 500 ms, 128000 pulses)**

FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 CRC

**7. Set 9DH parameters: 4th section parameters ( 1000 RPM, 100 ms, 10 ms, -128000 pulses)**

FA 01 9D 03 E8 00 64 00 0A FF FE 0C 00 CRC

**8. Save Parameters**

FA 01 42 01 CRC

**9. Coordinates cleared for easy observation (optional)**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**10. Control motor operation**

FA 01 FE 01 CRC

**11. The motor starts running.**

**12. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**13. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

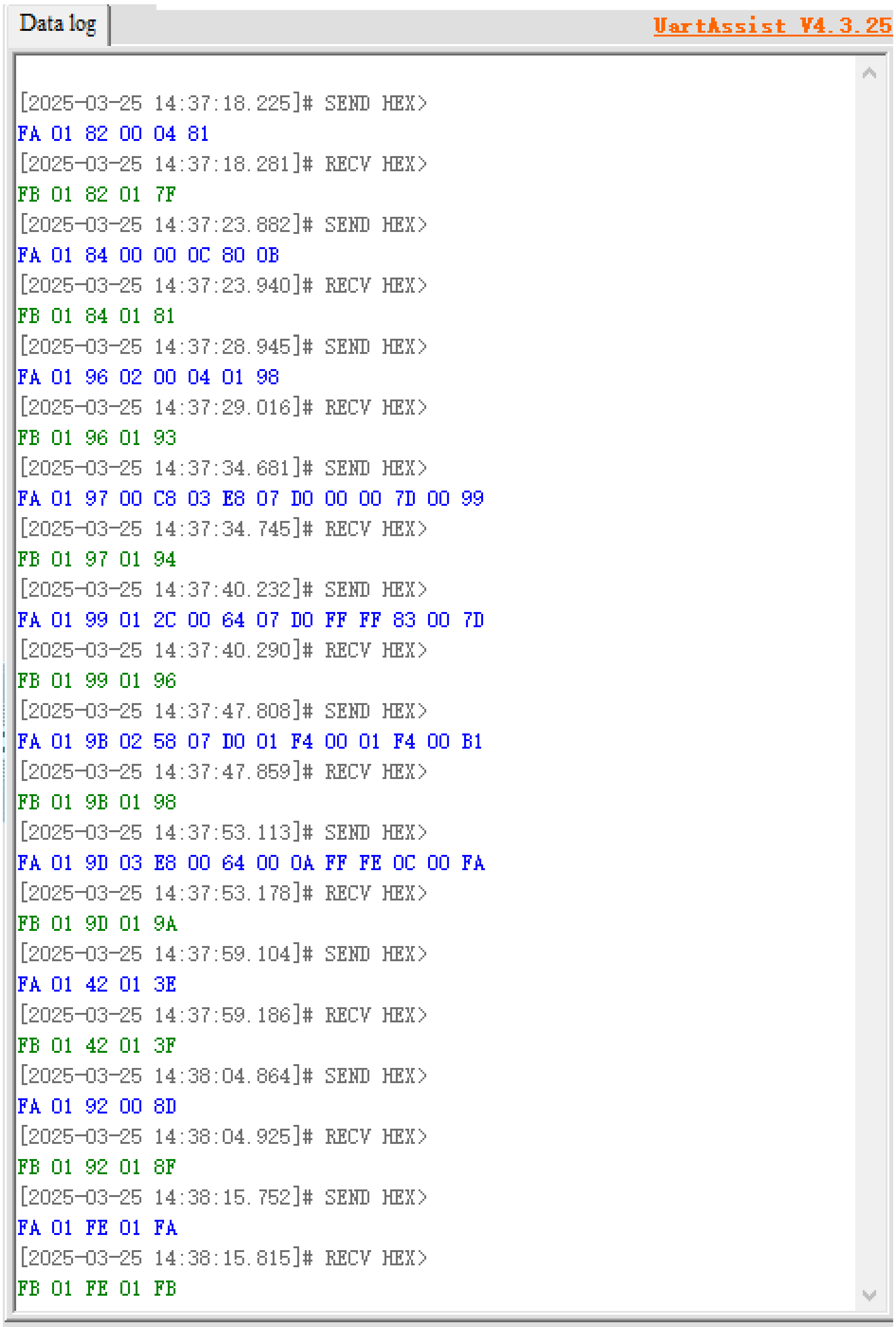
If you need to run automatically after power on, add the following configuration:

**14. Set automatic operation on power on**

FA 01 FF 01 CRC

**15. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-25 14:37:18.225]# SEND HEX>
FA 01 82 00 04 81
[2025-03-25 14:37:18.281]# RECV HEX>
FB 01 82 01 7F
[2025-03-25 14:37:23.882]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-25 14:37:23.940]# RECV HEX>
FB 01 84 01 81
[2025-03-25 14:37:28.945]# SEND HEX>
FA 01 96 02 00 04 01 98
[2025-03-25 14:37:29.016]# RECV HEX>
FB 01 96 01 93
[2025-03-25 14:37:34.681]# SEND HEX>
FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 99
[2025-03-25 14:37:34.745]# RECV HEX>
FB 01 97 01 94
[2025-03-25 14:37:40.232]# SEND HEX>
FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 7D
[2025-03-25 14:37:40.290]# RECV HEX>
FB 01 99 01 96
[2025-03-25 14:37:47.808]# SEND HEX>
FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 B1
[2025-03-25 14:37:47.859]# RECV HEX>
FB 01 9B 01 98
[2025-03-25 14:37:53.113]# SEND HEX>
FA 01 9D 03 E8 00 64 00 0A FF FE 0C 00 FA
[2025-03-25 14:37:53.178]# RECV HEX>
FB 01 9D 01 9A
[2025-03-25 14:37:59.104]# SEND HEX>
FA 01 42 01 3E
[2025-03-25 14:37:59.186]# RECV HEX>
FB 01 42 01 3F
[2025-03-25 14:38:04.864]# SEND HEX>
FA 01 92 00 8D
[2025-03-25 14:38:04.925]# RECV HEX>
FB 01 92 01 8F
[2025-03-25 14:38:15.752]# SEND HEX>
FA 01 FE 01 FA
[2025-03-25 14:38:15.815]# RECV HEX>
FB 01 FE 01 FB
```

### 6.6.6 Example of cyclic multi-segment absolute position operation configuration

**1. Set 82H parameters: position mode, bus control**

FA 01 82 00 04 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 0C 80 CRC

**3. Set 96H parameters: cyclic multi-segment position operation, absolute displacement, 4-segment displacement**

FA 01 96 02 01 04 01 CRC

**4. Set 97H parameters: Parameters for the first section ( 200 RPM, 1000 ms, 2000 ms, 32000 pulses)**

FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 CRC

**5. Set 99H parameters: Parameters for the second section ( 300 RPM, 100 ms, 2000 ms, -32000 pulses)**

FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 CRC

**6. Set 9BH parameters: 3rd section parameters ( 600 RPM, 2000 ms, 500 ms, 128000 pulses)**

FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 CRC

**7. Set 9DH parameters: 4th section parameters ( 1000 RPM, 100 ms, 10 ms, 0 pulses)**

FA 01 9D 03 E8 00 64 00 0A 00 00 00 00 CRC

**8. Save Parameters**

FA 01 42 01 CRC

**9. Coordinates cleared for easy observation (optional)**

FA 01 92 00 CRC

Note: The above parameters only need to be set once and no need to be set repeatedly.

**10. Control motor operation**

FA 01 FE 01 CRC

**11. The motor starts running.**

**12. After the run is completed, check the coordinates and they are consistent with the set displacement**

FA 01 31 CRC

**13. During operation**

FA 01 F7 00 CRC command can terminate the motor operation

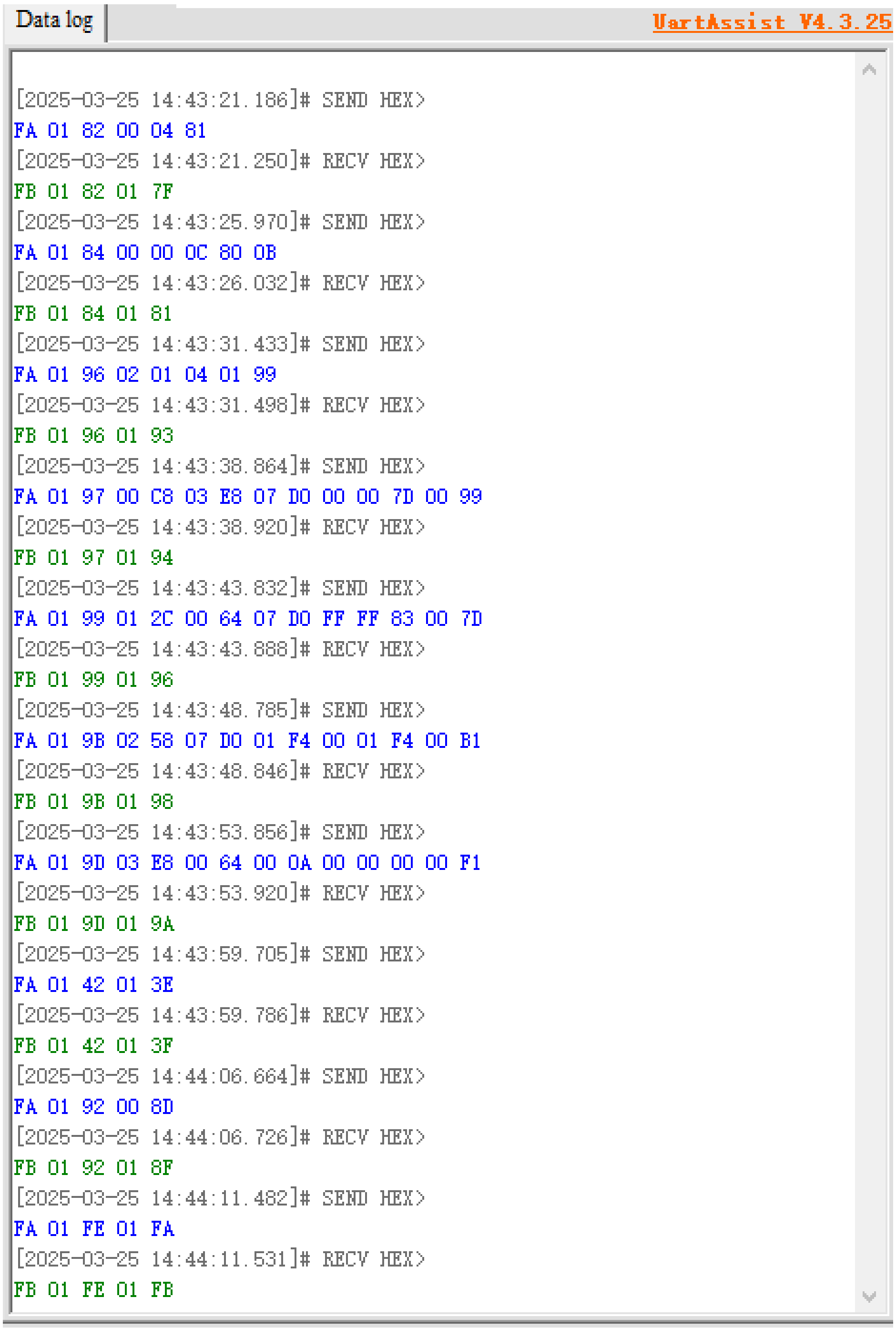
If you need to run automatically after power on, add the following configuration:

**14. Set automatic operation on power on**

FA 01 FF 01 CRC

**15. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-25 14:43:21.186]# SEND HEX>
FA 01 82 00 04 81
[2025-03-25 14:43:21.250]# RECV HEX>
FB 01 82 01 7F
[2025-03-25 14:43:25.970]# SEND HEX>
FA 01 84 00 00 0C 80 0B
[2025-03-25 14:43:26.032]# RECV HEX>
FB 01 84 01 81
[2025-03-25 14:43:31.433]# SEND HEX>
FA 01 96 02 01 04 01 99
[2025-03-25 14:43:31.498]# RECV HEX>
FB 01 96 01 93
[2025-03-25 14:43:38.864]# SEND HEX>
FA 01 97 00 C8 03 E8 07 D0 00 00 7D 00 99
[2025-03-25 14:43:38.920]# RECV HEX>
FB 01 97 01 94
[2025-03-25 14:43:43.832]# SEND HEX>
FA 01 99 01 2C 00 64 07 D0 FF FF 83 00 7D
[2025-03-25 14:43:43.888]# RECV HEX>
FB 01 99 01 96
[2025-03-25 14:43:48.785]# SEND HEX>
FA 01 9B 02 58 07 D0 01 F4 00 01 F4 00 B1
[2025-03-25 14:43:48.846]# RECV HEX>
FB 01 9B 01 98
[2025-03-25 14:43:53.856]# SEND HEX>
FA 01 9D 03 E8 00 64 00 0A 00 00 00 00 F1
[2025-03-25 14:43:53.920]# RECV HEX>
FB 01 9D 01 9A
[2025-03-25 14:43:59.705]# SEND HEX>
FA 01 42 01 3E
[2025-03-25 14:43:59.786]# RECV HEX>
FB 01 42 01 3F
[2025-03-25 14:44:06.664]# SEND HEX>
FA 01 92 00 8D
[2025-03-25 14:44:06.726]# RECV HEX>
FB 01 92 01 8F
[2025-03-25 14:44:11.482]# SEND HEX>
FA 01 FE 01 FA
[2025-03-25 14:44:11.531]# RECV HEX>
FB 01 FE 01 FB
```

## 6.7 Multi-machine synchronous position operation

The synchronous operation of multiple machines can be achieved through multi-segment position operation, as shown below:

Assume that n motors (corresponding to addresses 1, 2, … , n) run in absolute position synchronization.

**1. Set the first motor 96H parameters: single multi-stage position operation, absolute displacement, 1-stage displacement**

FA 01 96 01 01 01 01 CRC

**2. Set the first motor 97H parameters: the first section parameters ( 200 , 1000 , 2000 , 32000 )**

FA 01 97 **00 C8 03 E8 07 D0 00 00 7D 00** CRC

**3. Set the second motor 96H parameters: single multi-stage position operation, absolute displacement, 1-stage displacement**

FA 02 96 01 01 01 01 CRC

**4. Set the second motor 97H parameters: Parameters of the first section ( 200 , 1000 , 2000 , 32000 )**

FA 02 97 **00 C8 03 E8 07 D0 00 00 7D 00** CRC

**5. Set the nth motor 96H parameters: single multi-stage position operation, absolute displacement, 1-stage displacement**

FA 0n 96 01 01 01 01 CRC

**6. Set the nth motor 97H parameters: Parameters of the first section ( 200 , 1000 , 2000 , 32000 )**

FA 0n 97 **00 C8 03 E8 07 D0 00 00 7D 00** CRC

**7. Broadcast command controls motor operation**

FA 00 FE 01 CRC

It can be observed that 1,2, … ,n motors are running simultaneously.

# Part7. Speed Control Mode Description

The speed control mode is divided into single speed mode and multi-speed mode.

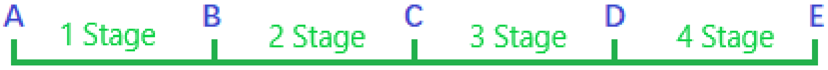
## 7.1 Single- speed mode and multi- speed mode description

A
Single speed operation

### 1. Single speed mode

In single-speed mode, the motor runs at the set speed and acceleration all the time .

### 2. Multi-speed mode



multi -speed mode , up to 4 speed parameters can be configured.

Each segment can be independently configured with speed and running time parameters.

Take 4-speed operation as an example:

single multi- speed operation , the motor operation sequence is:

A→B→C→D→E→Stop.

the cyclic multi- speed operation is in progress , the motor operation sequence is:

A→B→C→D→E→A→B→ … and the cycle continues.

## 7.2 Single speed mode operation

| Downlink frame (PC → SmartMotor ) | | | | | | |
|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Byte 10 |
| Frame Header | Slave Address | Function code | Speed | Acceleration time | Deceleration time | Checksum |
| FA | addr | F6H | speed | accTime | decTime | CRC |

speed: Maximum speed (range -3000~3000, unit: RPM)

accTime : 0~1000RPM acceleration time (range 0~65535, unit: ms )

decTime : 1000~0RPM deceleration time (range 0~65535, unit: ms )

Note: speed is a positive/negative number that determines the direction of the motor.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | F6H | status | CRC |

| | |
|---|---|
| status = 0 | Run failed |
| status = 1 | Successful operation |

Note: F7H command can stop speed mode operation.

## 7.3 Single -speed mode operation configuration example

**1. Set 82H parameters: speed mode, bus control**

FA 01 82 01 00 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 0C 80 CRC

**3. Set F6H parameters: speed 200RPM, acceleration time 1000ms, deceleration time 1000ms**

FA 01 F6 00 C8 03 E8 03 E8 CRC

**4. Observe the motor operation**

**5. Stop the motor**

FA 01 F7 00 CRC

**6. Save Parameters**

FA 01 42 01 CRC

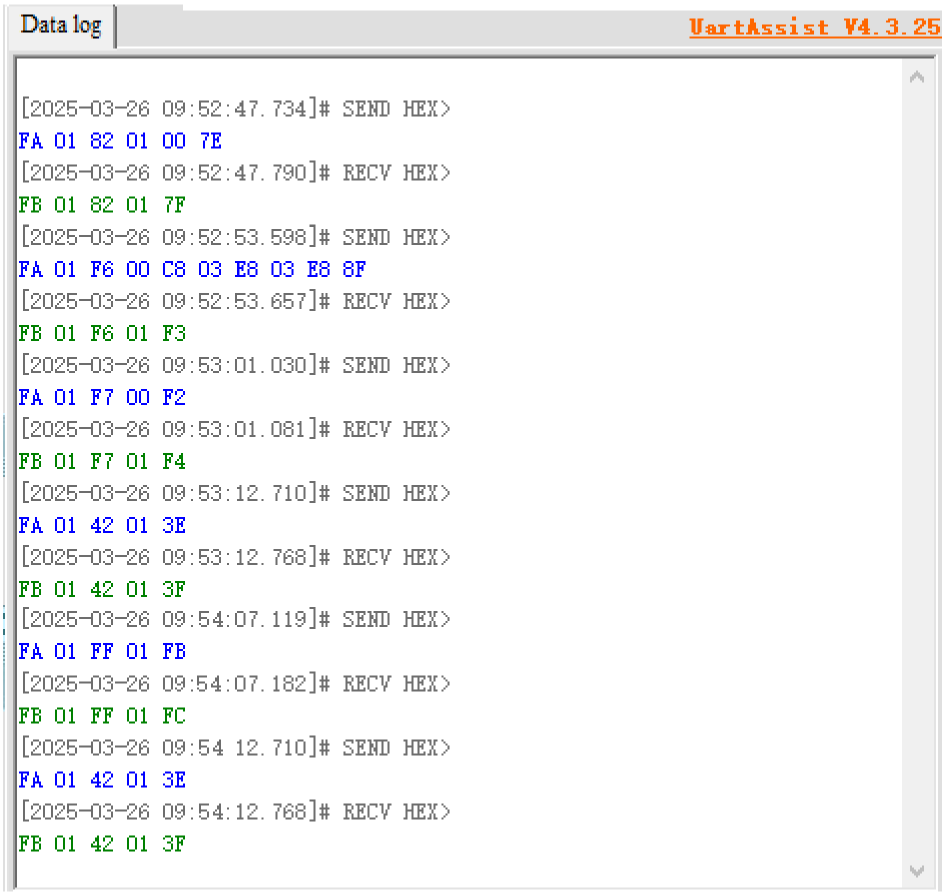
If you need to run automatically after power on, add the following configuration:

**7. Set automatic operation on power on**

FA 01 FF 01 CRC

**8. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-26 09:52:47.734]# SEND HEX>
FA 01 82 01 00 7E
[2025-03-26 09:52:47.790]# RECV HEX>
FB 01 82 01 7F
[2025-03-26 09:52:53.598]# SEND HEX>
FA 01 F6 00 C8 03 E8 03 E8 8F
[2025-03-26 09:52:53.657]# RECV HEX>
FB 01 F6 01 F3
[2025-03-26 09:53:01.030]# SEND HEX>
FA 01 F7 00 F2
[2025-03-26 09:53:01.081]# RECV HEX>
FB 01 F7 01 F4
[2025-03-26 09:53:12.710]# SEND HEX>
FA 01 42 01 3E
[2025-03-26 09:53:12.768]# RECV HEX>
FB 01 42 01 3F
[2025-03-26 09:54:07.119]# SEND HEX>
FA 01 FF 01 FB
[2025-03-26 09:54:07.182]# RECV HEX>
FB 01 FF 01 FC
[2025-03-26 09:54 12.710]# SEND HEX>
FA 01 42 01 3E
[2025-03-26 09:54:12.768]# RECV HEX>
FB 01 42 01 3F
```

# 7.4 Multi- speed mode operation

## 7.4.1 Speed segment parameter configuration instructions

### 1. Common configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Bytes 6-7 | Bytes 8-9 | Byte 10 |
| Frame Header | Slave Address | Function code | model | Number of segments | Acceleration time | Deceleration time | Checksum |
| FA | addr | A0H | mode | number | accTime | decTime | CRC |

mode: 00: Single run 01: Cycle run (default value 00)
number: speed segment number (range 1~4) (default value 4)
accTime : acceleration time (range 0~65535, unit: ms ) (default value 100)
decTime : deceleration time (range 0~65535, unit: ms ) (default value 100)

Note: The acceleration/deceleration time is the acceleration/deceleration time of 0-1000rpm. When switching between segments, the acceleration/deceleration will be automatically and smoothly transitioned.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | A0H | status (uint8_t) | CRC |

status = 0 Setting failed
status = 1 Setting successful

## 2. Section 1 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Running speed | Run time | Checksum |
| FA | addr | A1H | speed1 | Time1 | CRC |

speed1: running speed (range -3000 ~ 3000, unit: RPM, positive and negative numbers determine the running direction)

time1: running time (range 0 ~ 65535, unit: 100ms)

Note: The above parameters can be modified during operation. The parameters will take effect the next time this section is run. The same applies to other sections.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | A1H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 3. Section 2 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Running speed | Run time | Checksum |
| FA | addr | A2H | speed2 | time2 | CRC |

speed2: running speed (range -3000 ~ 3000, unit: RPM, positive and negative numbers determine the running direction)

time2: running time (range 0 ~ 65535, unit: 100ms)

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | A2H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 4. Section 3 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Running speed | Run time | Checksum |
| FA | addr | A3H | speed3 | time3 | CRC |

speed3: running speed(range -3000 ~ 3000, unit: RPM, positive and negative numbers determine the running direction)

time3: running time (range 0 ~ 65535, unit: 100ms)

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | A3H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 5. Section 4 parameter configuration instructions

| Downlink frame (PC → SmartMotor ) | | | | | |
|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Byte 8 |
| Frame Header | Slave Address | Function code | Running speed | Run time | Checksum |
| FA | addr | A4H | speed4 | time4 | CRC |

speed4: running speed (range -3000 ~ 3000, unit: RPM, positive and negative numbers determine the running direction)

time4: running time (range 0 ~ 65535, unit: 100ms)

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | A4H | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

### 7.4.2 Multi- speed mode operation instructions

multi- speed mode operation. Use the F3H instruction to enable the motor to start running.

The F3H or F7H command can stop the motor.

When stopping, the motor stops running in the stop mode set by the 87H instruction.

### 7.4.3 Multi- speed mode operation configuration example

**1. Set 82H parameters: multi-speed mode, bus control**

FA 01 82 01 01 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 0C 80 CRC

**3. Set A0H parameters: single run, 4 speeds, acceleration time 100, deceleration time 100**

FA 01 A0 **00 04 00 64** 00 64 CRC

**4. Set A1H parameters: Parameters for the first section ( 50 RPM, 10000 ms)**

FA 01 A1 **00 32 00 64** CRC

**5. Set A2H parameters: Parameters for the second stage (- 50 RPM, 10000 ms)**

FA 01 A2 **FF CE 00 64** CRC

**6. Set A3H parameters: 3rd section parameters ( 200 RPM, 10000 ms)**

FA 01 A3 **00 C8 00 64** CRC

**7. Set A4H parameters: 4th segment parameters (- 200 RPM, 10000 ms)**

FA 01 A4 **FF 38 00 64** CRC

**8. Save Parameters**

FA 01 42 01 CRC

**9. Enable motor operation**

FA 01 F3 01 CRC

**10. Observe the motor operation**

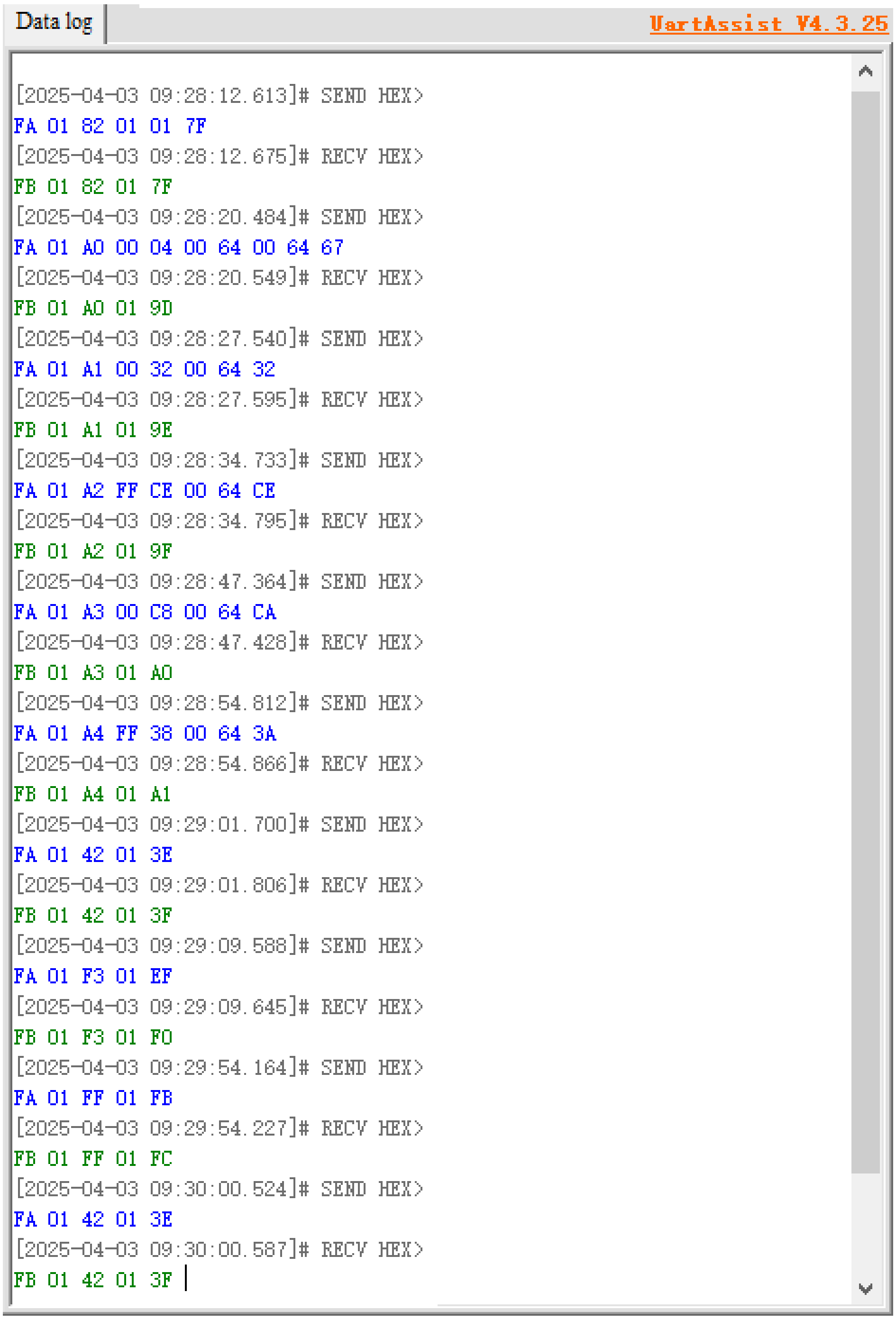
If you need to run automatically after power on, add the following configuration:

**11. Set automatic operation on power on**

FA 01 FF 01 CRC

**12. Save Parameters**

FA 01 42 01 CRC



```
[2025-04-03 09:28:12.613]# SEND HEX>
FA 01 82 01 01 7F
[2025-04-03 09:28:12.675]# RECV HEX>
FB 01 82 01 7F
[2025-04-03 09:28:20.484]# SEND HEX>
FA 01 A0 00 04 00 64 00 64 67
[2025-04-03 09:28:20.549]# RECV HEX>
FB 01 A0 01 9D
[2025-04-03 09:28:27.540]# SEND HEX>
FA 01 A1 00 32 00 64 32
[2025-04-03 09:28:27.595]# RECV HEX>
FB 01 A1 01 9E
[2025-04-03 09:28:34.733]# SEND HEX>
FA 01 A2 FF CE 00 64 CE
[2025-04-03 09:28:34.795]# RECV HEX>
FB 01 A2 01 9F
[2025-04-03 09:28:47.364]# SEND HEX>
FA 01 A3 00 C8 00 64 CA
[2025-04-03 09:28:47.428]# RECV HEX>
FB 01 A3 01 A0
[2025-04-03 09:28:54.812]# SEND HEX>
FA 01 A4 FF 38 00 64 3A
[2025-04-03 09:28:54.866]# RECV HEX>
FB 01 A4 01 A1
[2025-04-03 09:29:01.700]# SEND HEX>
FA 01 42 01 3E
[2025-04-03 09:29:01.806]# RECV HEX>
FB 01 42 01 3F
[2025-04-03 09:29:09.588]# SEND HEX>
FA 01 F3 01 EF
[2025-04-03 09:29:09.645]# RECV HEX>
FB 01 F3 01 F0
[2025-04-03 09:29:54.164]# SEND HEX>
FA 01 FF 01 FB
[2025-04-03 09:29:54.227]# RECV HEX>
FB 01 FF 01 FC
[2025-04-03 09:30:00.524]# SEND HEX>
FA 01 42 01 3E
[2025-04-03 09:30:00.587]# RECV HEX>
FB 01 42 01 3F
```

# Part8. Torque Control Mode Description

In torque control mode, the motor accelerates in the direction of the set output torque (torque). When it accelerates to the maximum speed limit ( MaxSpeed ) or the output torque is insufficient to support continued acceleration, the motor will stop accelerating and the speed fluctuates according to the load.

When the load is equal to the output torque, the motor will stop.

When the load is greater than the output torque, the motor will be dragged to rotate in the opposite direction, which becomes a damping torque.

Setting an appropriate filter constant ( filterTime ) can make the operation smoother and more stable. Too large a filter constant will also reduce the motor response speed.

## 8.1 Torque mode configuration instructions

Configuration directives:

| Downlink frame (PC → SmartMotor ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Bytes 4-5 | Bytes 6-7 | Bytes 8-9 | Bytes 10-11 | Byte 12 |
| Frame Header | Slave Address | Function code | Output torque | Filter constant | Forward maximum speed limit | Negative maximum speed limit | Checksum |
| FA | addr | 9F | torque | filterTime | forward MaxSpeed | backward MaxSpeed | CRC |

Torque: output torque (range -3000~3000, unit: 0.1% rated torque)
filterTime : filter constant (range 0~3000, unit: 0.01ms)
forwardMaxSpeed : Forward maximum speed limit (range 0~3000, unit: RPM)
backwardMaxSpeed : Negative maximum speed limit (range 0~3000, unit: RPM)

Note: The positive/negative value of the output torque affects the running direction of the motor.

Return data:

| Uplink frame (PC ← SmartMotor ) | | | | |
|---|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 |
| Frame Header | Slave Address | Function code | Set Status | Checksum |
| FB | addr | 9F | status (uint8_t) | CRC |

| | |
|---|---|
| status = 0 | Setting failed |
| status = 1 | Setting successful |

## 8.2 Torque mode operation command

Torque control operation does not require additional instructions. Use the F3H instruction to enable the motor to start running.

The F3H or F7H command can stop the motor running in torque mode.

When the torque mode is stopped, the motor stops in the stop mode set by the 87H command.

## 8.3 Torque Mode Operation Configuration Example

**1. Set 82H parameters: torque mode, bus control**

FA 01 82 02 00 CRC

**2. Set 84H parameters: 3200 pulses/circle**

FA 01 84 00 00 0C 80 CRC

**3. Set 9FH parameters: torque 100, filter constant 80, positive maximum speed 60, negative maximum speed 30**

FA 01 9F 00 64 00 50 00 3C 00 1E CRC

**4. Save Parameters**

FA 01 42 01 CRC

**5. Enable motor operation**

FA 01 F3 01 CRC

**6. Stop the motor**

FA 01 F7 00 CRC

The motor stops running in the stop mode set by 87H instruction.

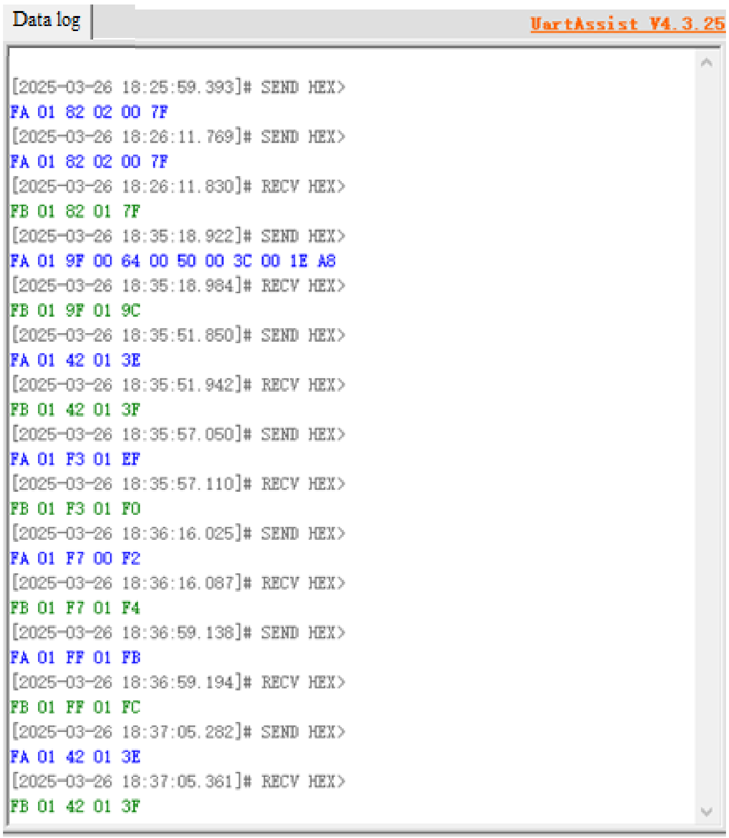
If you need to run automatically after power on, add the following configuration:

**7. Configure automatic operation at power on**

FA 01 FF 01 CRC

**8. Save Parameters**

FA 01 42 01 CRC



```
[2025-03-26 18:25:59.393]# SEND HEX>
FA 01 82 02 00 7F
[2025-03-26 18:26:11.769]# SEND HEX>
FA 01 82 02 00 7F
[2025-03-26 18:26:11.830]# RECV HEX>
FB 01 82 01 7F
[2025-03-26 18:35:18.922]# SEND HEX>
FA 01 9F 00 64 00 50 00 3C 00 1E A8
[2025-03-26 18:35:18.984]# RECV HEX>
FB 01 9F 01 9C
[2025-03-26 18:35:51.850]# SEND HEX>
FA 01 42 01 3E
[2025-03-26 18:35:51.942]# RECV HEX>
FB 01 42 01 3F
[2025-03-26 18:35:57.050]# SEND HEX>
FA 01 F3 01 EF
[2025-03-26 18:35:57.110]# RECV HEX>
FB 01 F3 01 F0
[2025-03-26 18:36:16.025]# SEND HEX>
FA 01 F7 00 F2
[2025-03-26 18:36:16.087]# RECV HEX>
FB 01 F7 01 F4
[2025-03-26 18:36:59.138]# SEND HEX>
FA 01 FF 01 FB
[2025-03-26 18:36:59.194]# RECV HEX>
FB 01 FF 01 FC
[2025-03-26 18:37:05.282]# SEND HEX>
FA 01 42 01 3E
[2025-03-26 18:37:05.361]# RECV HEX>
FB 01 42 01 3F
```

# Part9. Common problems and precautions

## 9.1 Precautions

1. Power input voltage 20 V- 60 V;
2. Do not unplug the power cord or signal cable while it is powered on to avoid damaging the driver board ;

## 9.2 Frequently asked questions

| NO | Question | Solution |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |
| 11 | | |
| 12 | | |

# Part10. Contact us

https://makerbase.aliexpress.com/

# Part11. Fault code correspondence table

Note: The fault code can be read out through 3B command.

| Flashing red light | Fault Codes | Fault Information | Cause | Cause | Reference solution |
|---|---|---|---|---|---|
| 2 times | 0200H | Location out of tolerance | The position deviation is greater than the threshold set by the 75H instruction | Motor stall | Check the mechanical structure |
| | | | | Input pulse frequency is too high | Reduce the pulse frequency |
| | | | | The motor accelerates or decelerates too quickly when starting, stopping or reversing. | Increase acceleration and deceleration time |
| | | | | Insufficient motor torque | Choose a high torque motor |
| | | | | The gain is low and the motor responds slowly | Increase gain |
| | | | | The load inertia is large and the motor responds slowly | Increase the 75H instruction threshold |
| 3 times | 0300H | Driver overvoltage | Input voltage is too high | Input voltage is unstable | Choose high-quality power supply |
| | | | | Large load inertia and large start-stop regeneration energy | Increase acceleration and deceleration time or configure a brake module to absorb regenerative energy |
| | 0301H | Driver undervoltage | Input voltage is too low | Input voltage is unstable | Choose high quality or high power supply |
| 4 times | 0400H | Motor overload | Overload energy is too large | Too much load | Larger motor option |
| | | | | Start, stop or change direction too quickly | Increase acceleration and deceleration time |
| | | | | Mechanical factors cause motor stalling | Check the mechanical structure |
| | | | | Inappropriate gain or too strong rigidity | Adjusting Gain |

| Flashing red light | Fault Codes | Fault Information | Cause | Cause | Reference solution |
|---|---|---|---|---|---|
| 4 times | 0401H | Motor stall | The motor does not run as instructed | Mechanical or other factors cause the motor to stall | Check the mechanical structure |
|  | 0402H | Hardware overcurrent | Current exceeds hardware range | Inappropriate gain or too strong rigidity | Adjusting Gain |
|  | 0403H | Software overcurrent | Current exceeds software range | Load sudden change | Check the mechanical structure |
|  |  |  |  | The driver is damaged or the motor coil is short-circuited | Check the motor phase line |
| 5 times | 0500H | Motor overspeed | The motor speed is too high | The speed exceeds the maximum allowed value | Reduce motor speed |
| 6 times | 0600H | Temperature is too high | The motor detects that the temperature is too high | Ambient temperature is too high | Lower the ambient temperature |
|  |  |  |  | Long-term full-load operation causes the motor to overheat | Larger motor option |
| 7 times | 0700H | The motor needs to be restarted | Remind to restart after power failure | The user has configured some parameters, which need to be restarted to take effect | Power off and restart |
| 8 times | 0800H | Return to origin does not match | The return to origin method does not match | The positive and negative directions of the mechanical origin offset are incorrect (positive value = positive direction, negative value = negative direction) | 94H command modifies the direction of the parameter origin offset value |

| Flashing red light | Fault Codes | Fault Information | Cause | Cause | Reference solution |
|---|---|---|---|---|---|
| 8 times | 0800H | Return to origin does not match | The return to origin method does not match | The switch origin return is selected, and the DI port is not configured with the corresponding switch | 21H instruction configuration corresponding switch function |
| | | Return to origin does not match | The return to origin method does not match | The unreasonable phenomenon that the origin switch and overtravel are turned on at the same time during the return to zero process | Check if the switches are turned on at the same time |
| | 0801H | Return to origin timeout | The return to zero time exceeds the setting value of 93H | The origin is far away and cannot be found within the specified time. | The 95H instruction increases the speed or the 93H instruction increases the timeout value |
| | | | | Switch has no trigger signal | Check whether the switch signal is normal |
| 9 times | 0900H | Encoder failure | Encoder data is incorrect | The encoder chip is damaged or the magnet is not installed properly | Check if the magnet is loose |
| 10 times | 0A00H | EEPROM Error | EEPROM parameter abnormality | The user configured abnormal parameters | 3FH command restores EEPROM parameters |
| | | | EEPROM chip is damaged | Frequent writing to EEPROM | Replace the EEPROM chip |
| 11 times | 0B00H | Code exception | FLASH data abnormality | | IAP firmware upgrade or repair |
| 12 times | 0C03H | Forward overtravel connection | | Remind that the forward overtravel switch signal is turned on, limiting forward movement | Disconnect the forward overtravel switch |

| Flashing red light | Fault Codes | Fault Information | Cause | Cause | Reference solution |
|---|---|---|---|---|---|
| 12 times | 0C04H | Reverse overtravel connection | | Remind that the reverse overtravel switch signal is turned on, limiting reverse movement | Disconnect the reverse overtravel switch |
| | 0C05H | DI function duplicate assignment | | Assign the same function to multiple DI ports | 21H instruction reconfigures DI function |
| | 0C06H | Emergency Stop | | Emergency shutdown | Use F7H command to cancel the emergency stop. |
| | 0C07H | Motor Speed | | Rapid motor shaft movement detected at power-up | Ensure that there is no external force interference on the motor shaft when power is turned on |
| | 0C08H | User parameter abnormality | | The user configured abnormal parameters | 3FH command restores user parameters |
| | 0C09H | System parameter abnormality | | The user configured abnormal parameters | 3FH command restores system parameters |