MKS SmartMotor

CAN User Manual V1.0.0

Note: This manual corresponds to the firmware version V 1.0.0

MKS SmartMotor_CAN 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

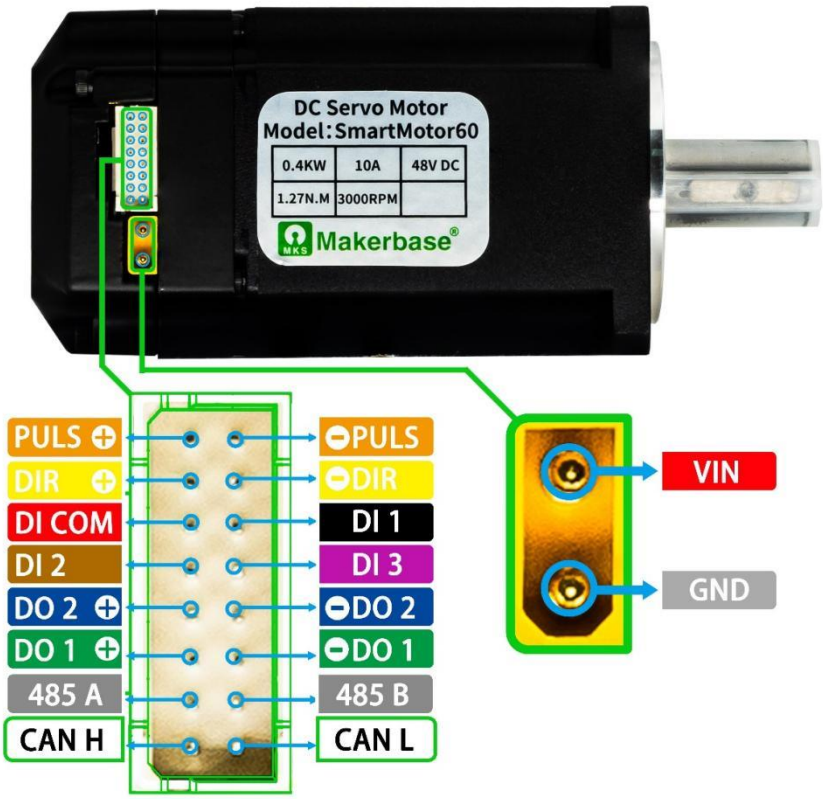
Motor performance	Maximum power 400W / 750W, rated torque 1.27NM / 2.40NM
	Rated speed 3000RPM, response speed less than 0.5ms
	Speed fluctuation rate: <±0.03 (load 0~100%): <±0.02×(0.9~1.1) power supply voltage
Position control mode	Pulse mode: Maximum receiving frequency 100KHZ (duty ratio 1:1)
	Pulse 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".

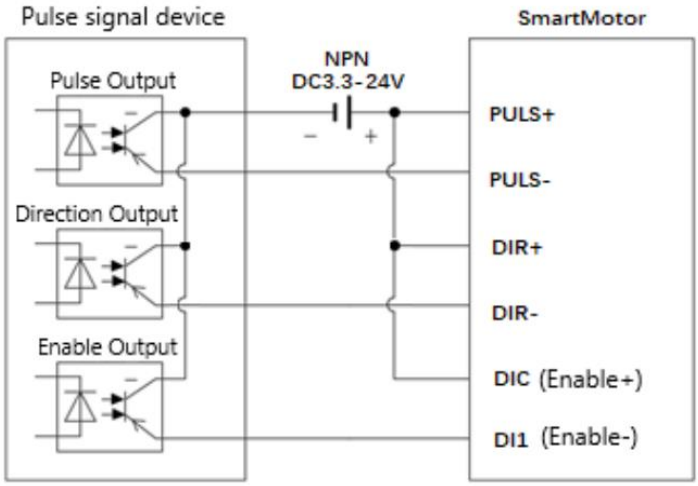
Part2. Wiring method

2.1 Pulse control wiring method

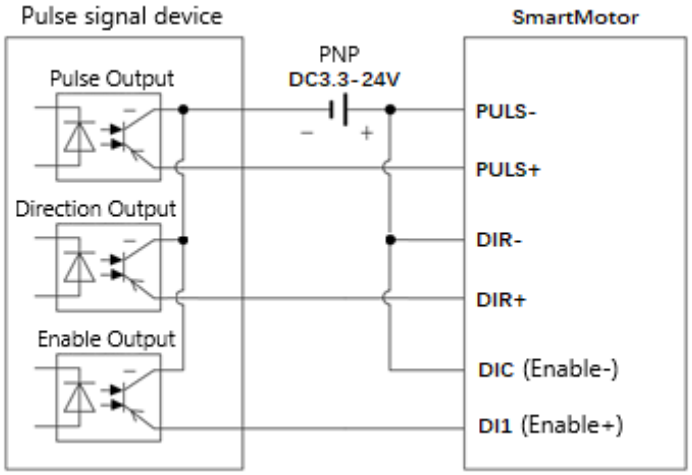
The pulse interface supports 3.3V-24V input, the maximum receiving pulse frequency is 100KHZ, and the duty cycle is as close to 1:1 as possible, otherwise the pulse may be lost, resulting in abnormal positioning.

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.

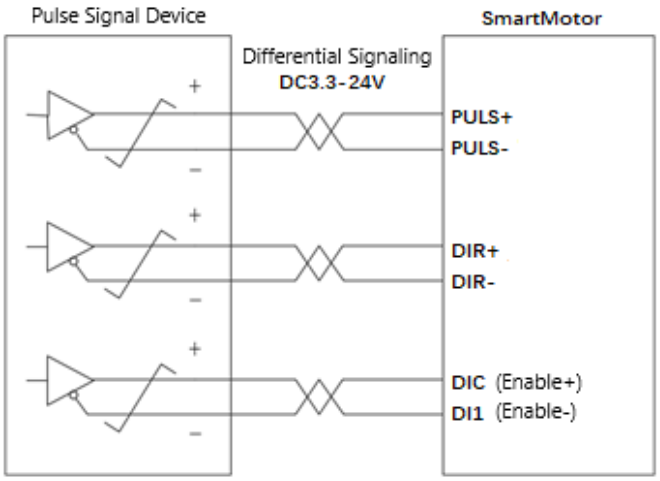
2.1.1 Common anode wiring method



2.1.2 Common cathode wiring method



2.1.3 Differential Wiring Method



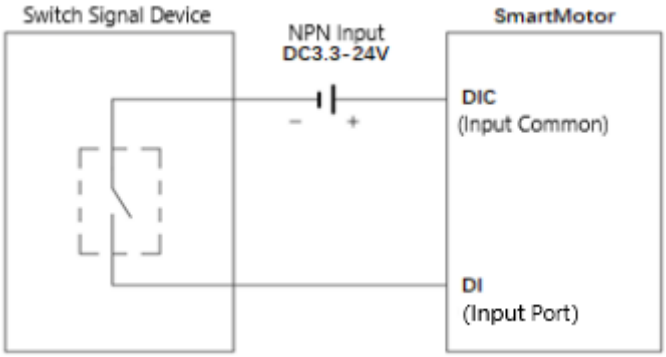
2.2 Input port wiring method

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.

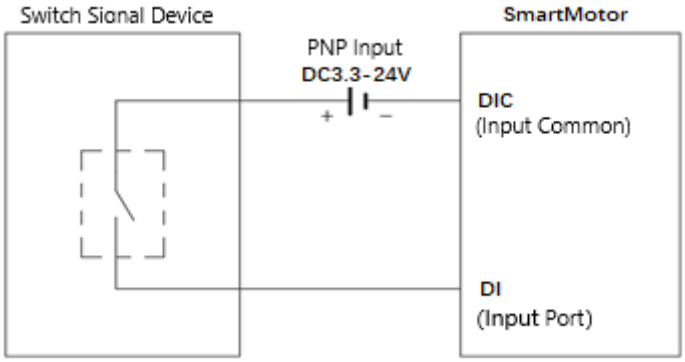
The input voltage range is DC3.3-24V, and the maximum input signal frequency is 100Hz.

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

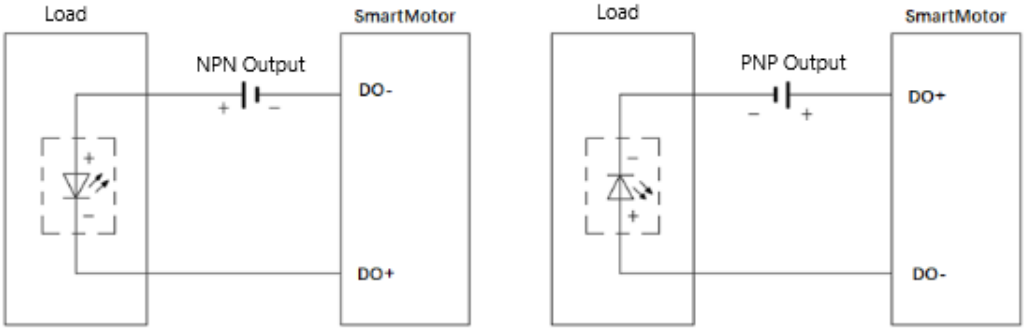


2.2.2 Common cathode wiring method



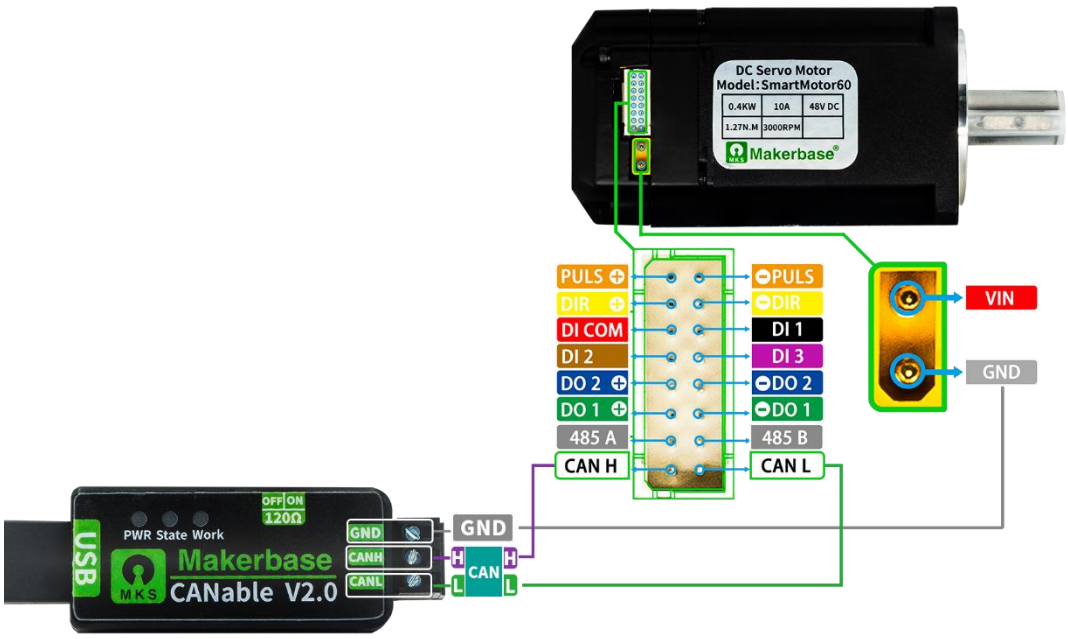
2.3 Output port wiring method

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.

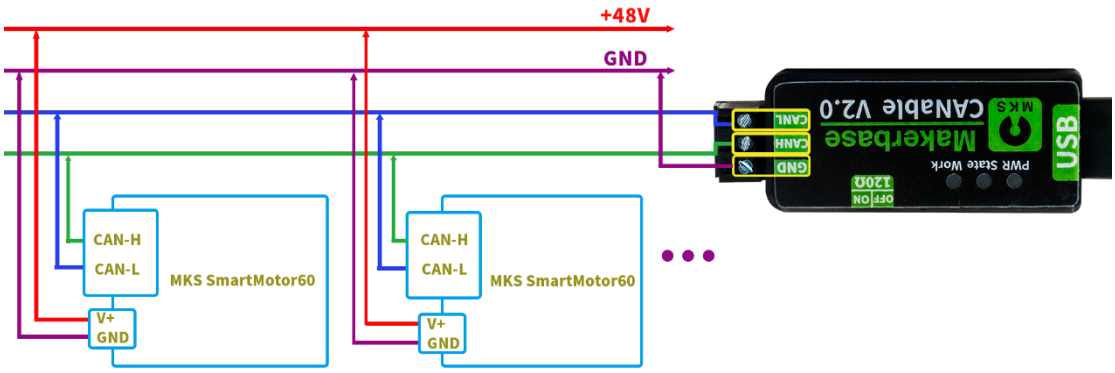


2.4 CAN wiring method

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. CAN message format description

The message uses a standard frame, and the maximum length of the data field is 8 bytes.

Downlink frame (PC → SmartMotor)					
address	...	Data length	Byte 1	Byte 2···Byte n-1	Byte n (n≤8)
address(ID)		DLC(n)	Function code (code)	Instruction data	Checksum (CRC)

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2···Byte n-1	Byte n (n≤8)
address(ID)		DLC(n)	Function code (code)	Return data	Checksum (CRC)

1. The command data and return data are in big-endian mode.
2. The address (ID) range is 00~2047 , the default address is 01, and the broadcast address is 00.
3. the data length (DLC) is 8, indicating the number of bytes in the data field.
4. The function code (code) executes the corresponding instruction , for example, 0x82 sets the working mode .
5. command data or return data , see "CAN Command Description ".
6. The checksum CRC is the sum of the address and data fields, taking the lower 8 bits .

For example, read the encoder value command:

Downlink frame (PC → SmartMotor)				
address	...	Data length	Byte 1	Byte 2
01		2	31	32

Checksum CRC = (0x01 + 0x3 1) & 0xFF = 0x3 2 & 0xFF = 0x3 2

7. 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 $=\frac{1}{3200}\times 360^\circ = 0.1125^\circ$

Encoder unit: the physical unit of the motor encoder.

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

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

Uplink frame (PC ← SmartMotor)									
address	...	DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
01		7	code	FB	FF	FF	FF	FF	CRC

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

Part4. CAN Command Description

Note1: When a command is sent using a broadcast address or a group address, the slave will not respond.

Note2: 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)				
address	...	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			31H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	Command absolute position	Checksum
			31 H	value (int 32 _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 the lower 32 bits of instruction:

Downlink frame (PC → SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Sub-function code	Checksum
			35H	00H	CRC

Return the lower 32 bits of data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Bytes 2-5	Byte 6
01	...	6	Function code	Lower 32 bits	Checksum
			35H	value L (int 32 _t)	CRC

Read the high 32 bits of instruction:

Downlink frame (PC → SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Sub-function code	Checksum
			35H	01H	CRC

Return the high 32 bits of data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Bytes 2-5	Byte 6
01	...	6	Function code	Lower 32 bits	Checksum
			35H	value H (int 32 _t)	CRC

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

Using 17-bit encoder, single-turn value range 0~0x 20 000

Calculation rule: The motor shaft rotates counterclockwise for one circle, and the encoder value is +0x 20 000;

The motor shaft rotates one circle clockwise, and the encoder value is -0x 20 000;

For example :

current encoder value is 0x 1492A. After one clockwise rotation (- 0x 20 000), the encoder value is 0xFFFFFFFFFFFF492A.

current encoder value is 0x 1492A. After one counterclockwise rotation (+ 0x 20 000), 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)				
address	...	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			32H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Real-time speed	Checksum
			32H	rpm(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)				
address	...	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			33H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	Number of pulses	Checksum
			33H	(int32_t)	CRC

5. Reading fault codes

Read instruction:

Downlink frame (PC → SmartMotor)				
address	...	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			3BH	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Fault Codes	Checksum
			3BH	(uint16_t)	CRC

Note1: For the definition of fault codes, see Part 11 "Fault Code Correspondence Table"

Note2: The fault alarm can be reset by command 41H

6. Read real-time torque output value

Read instruction:

Downlink frame (PC → SmartMotor)				
address	...	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			36H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Torque output value	Checksum
			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)				
address	…	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			37H	CRC

Return data:

Uplink frame (PC ← SmartMotor)						
address	…	Data length	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01		6	Function code	Bus voltage	Phase current	Checksum
			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)				
address	…	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			38H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	…	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Temperature value	Checksum
			38H	(int16_t)	CRC

Note: Unit: ° C

9. coder position deviation

Read instruction:

Downlink frame (PC → SmartMotor)				
address	…	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			39H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	…	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	Position deviation	Checksum
			39H	(int32_t)	CRC

10. Read servo enable status

Read instruction:

Downlink frame (PC → SmartMotor)				
address	…	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			3AH	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	…	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Enable status	Checksum
			3AH	(uint8_t)	CRC

Note1: Enabled 0: Not enabled

11. Read version information

Read instruction:

Downlink frame (PC → SmartMotor)				
address	…	Data length	Byte 1	Byte 2
01		2	Function code	Checksum
			40H	CRC

Return data:

Uplink frame (PC ← SmartMotor)						
address	…	Data length	Byte 1	Byte 2	Bytes 3-5	Byte 6
01		6	Function code	Hardware version	Firmware version	Checksum
			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)						
address	...	Data length	Byte 1	Byte 2	Byte 3	Byte 4
01		4	Function code	Main Mode	Secondary Mode	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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 one circle .

Set the command:

Downlink frame (PC → SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	Number of pulses per revolution	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Enable level	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Direction of rotation	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Shutdown mode	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Output pulse number	Checksum
			88H	outPulses_t	CRC

outPulses : Output pulse number (range 1~50)

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			88H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

7. Setting the CAN bit rate

Set the command:

Downlink frame (PC → SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Output pulse number	Checksum
			8A	baudrate	CRC

00 125K

01 250K

02 500K (default)

03 1000K

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			8A	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.

8. Set the slave address

Set the command:

Downlink frame (PC → SmartMotor)					
address	…	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Slave Address	Checksum
			8BH	address(uint16_t)	CRC

Note : 00 is the broadcast address, 01 is the default address.

Return data:

Uplink frame (PC ← SmartMotor)					
address	…	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			8BH	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.

9. Set group address

Set the command:

Downlink frame (PC → SmartMotor)					
address		Data length	Byte 1	Bytes 2-3	Byte 4
01	...	4	Function code	Group Address	Checksum
			8DH	address(uint16_t)	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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 01 FD 01 2C 64 00 0C 80 1B Motor 1 run

Send 00 FD 01 2C 64 00 0C 80 1A Motor 1 -6 Running

Send 50 FD 01 2C 64 00 0C 80 6A Motor 1 -3 Running

Send 51 FD 01 2C 64 00 0C 80 6B 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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Answer Mode	Checksum
			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 01 or fails 00

c. Default mode (ackMode = 2)

slave immediately returns the position control start 01 or fails 00

After the motor is finished running, it returns to 02

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			8CH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

11. Set user-defined parameters

Set the command:

Downlink frame (PC → SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	User Parameters	Checksum
			43H	USER ID	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			43H	status (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)							
address		Data length	Byte 1	Bytes 2-3	Bytes 4-5	Bytes 6-7	Byte 8
01	...	8	Function code	Kp	Kv	Tvi	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)						
address		Data length	Byte 1	Byte 2	Byte 3	Byte 4
01	...	4	Function code	Control Word	Rigidity level	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)						
address	...	Data length	Byte 1	Byte 2	Bytes 3-4	Byte 5
01	...	5	Function code	Control Word	Protection gain	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			72H	status (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 $0.52Kg \cdot cm^2$.

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

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)					
address	...	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Moment of inertia ratio	Checksum
			73H	Kj	CRC

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

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Bytes 2-3	Byte 4
01		4	Function code	Time constant	Checksum
			74H	delayTime	CRC

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

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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

Set the command:

Downlink frame (PC → SmartMotor)					
address	...	Data length	Byte 1	Bytes 2-5	Byte 6
01		6	Function code	Position deviation threshold	Checksum
			75H	pDeviation	CRC

pDeviation range: 1~ 1073741824Default value: 1310720

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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

Note1: One function option can only be associated with one DI port.

Note2: 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)										
address		length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
01	…	8	Function code	DI1 Function	DI1 Logic	DI2 Function	DI2 Logic	DI3 Function	DI3 Logic	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	…	3	Function code	Set Status	Checksum
			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)								
address		length	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
01	…	6	Function code	DO1 Function	DO1 Logic	DO2 function	DO2 Logic	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	…	3	Function code	Set Status	Checksum
			22H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

4.4.3 Read I O port status

Read instruction:

Downlink frame (PC → SmartMotor)				
address		Data length	Byte 1	Byte 2
01	…	2	Function code	Checksum
			23H	CRC

Return data:

Uplink frame (PC ← SmartMotor)						
address		Data length	Byte 1	Byte 2	Byte 3	Byte 4
01	…	4	Function code	DO port status	DI port status	Checksum
			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)						
address		Data length	Byte 1	Byte 2	Byte 3	Byte 4
01	...	4	Function code	DO1	DO2	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Control Word	Checksum
			42H	01H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Save state	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Control Word	Checksum
			3FH	mode	CRC

mode = 0 restores only user parameters (recommended)

mode = 1 restores user parameters and system parameters

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Restore Status	Checksum
			3FH	status	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)					
address	…	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Reset Mode	Checksum
			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)					
address	…	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Reset state	Checksum
			41H	status	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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Control Word	Function code	Checksum
			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:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2-n	Byte n+1
01		n+1	Function code	Parameter Value	Checksum
			code	param	CRC

param : corresponding parameter value

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

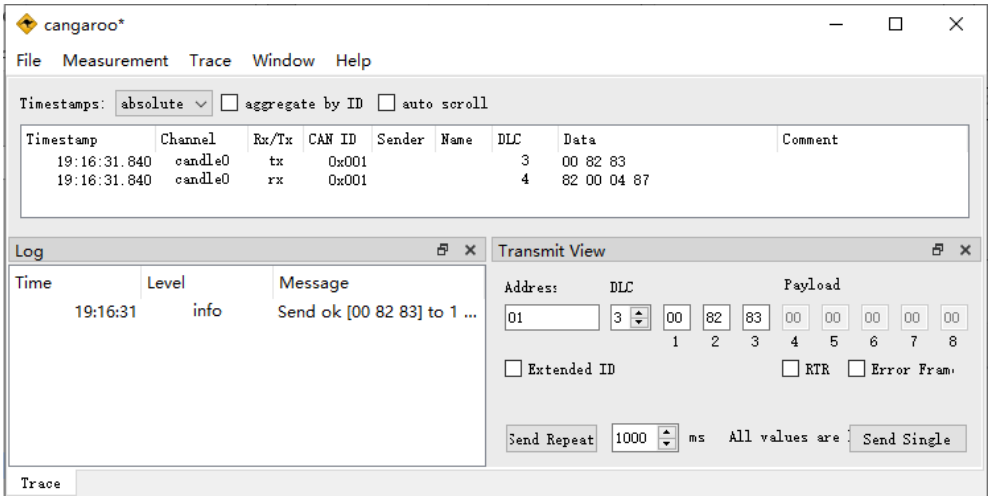
If the parameter does not support reading, the returned data is as follows:

Uplink frame (PC ← SmartMotor)						
address	...	Data length	Byte 1	Byte 2	Byte 3	Byte 4
01		4	Function code			Checksum
			code	FFH	FFH	CRC

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

Read instruction: 01 00 82 83

Return data:01 82 00 04 87



Part5. Motor zero return instructions

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

Note1: The motor return to zero function is only valid in "position mode".

Note2: 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)								
address		DLC	Byte 1	Byte 2	Byte 3	Byte 4	Bytes 5-6	Byte 7
01	...	7	Function code	Zero return mode	Return to zero direction	Zero return trigger mode	Return to zero timeout (ms)	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)						
address	...	DLC	Byte 1	Bytes 2-3	Bytes 4-7	Byte 8
01		8	Function code	Zero return torque	Origin offset	Checksum
			94H	HmTorque (uint16_t)	Orgoffset (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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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)							
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Bytes 6-7	Byte 8
01	...	8	Function code	"High" speed	"Low" speed	Acceleration and deceleration time	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)					
address		DLC	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Control Word	Checksum
			92H	00H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			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)					
address	…	DLC	Byte 1	Byte 2	Byte 3
01		3	Function code	Execution	Checksum
			91H	goZeroMode	CRC

goZeroMode 00: Execute the “origin return” function

01: Execute the "coordinate return to zero" function

Return data:

Uplink frame (PC ← SmartMotor)					
address	…	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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"

01 21 FF FF 06 00 FF FF CRC

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

01 93 00 00 00 EA 60 CRC

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

01 94 01 90 00 00 00 00 CRC

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

01 95 00 32 00 0A 00 C8 CRC

5. Save Parameters

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

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:

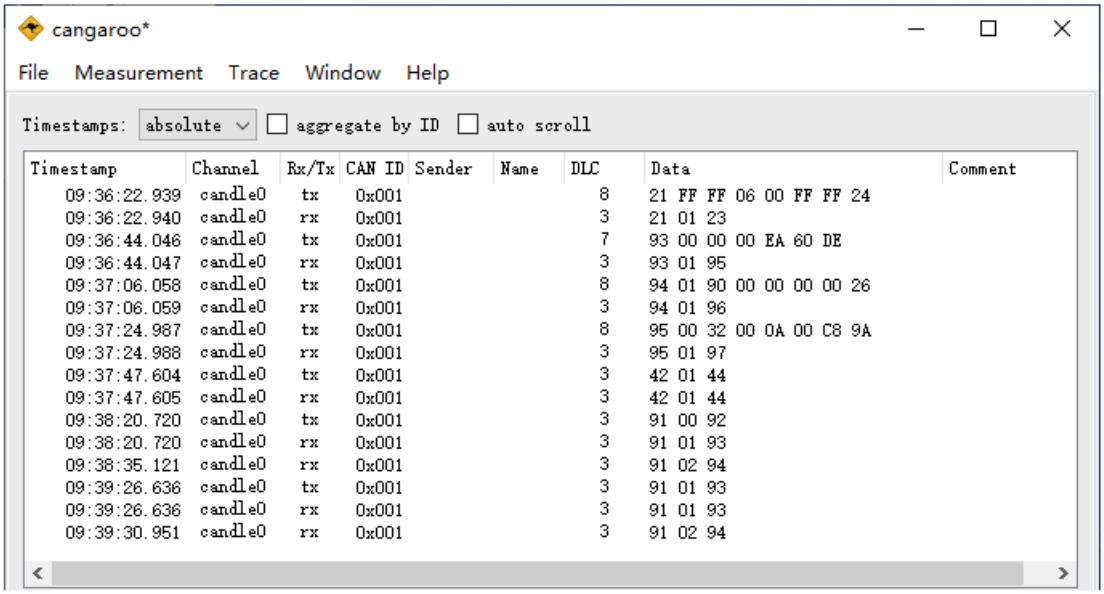
Reverse (clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

7. Execute coordinate return to zero

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.



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
09:36:22.939	candle0	tx	0x001			8	21 FF FF 06 00 FF FF 24	
09:36:22.940	candle0	rx	0x001			3	21 01 23	
09:36:44.046	candle0	tx	0x001			7	93 00 00 00 EA 60 DE	
09:36:44.047	candle0	rx	0x001			3	93 01 95	
09:37:06.058	candle0	tx	0x001			8	94 01 90 00 00 00 00 26	
09:37:06.059	candle0	rx	0x001			3	94 01 96	
09:37:24.987	candle0	tx	0x001			8	95 00 32 00 0A 00 C8 9A	
09:37:24.988	candle0	rx	0x001			3	95 01 97	
09:37:47.604	candle0	tx	0x001			3	42 01 44	
09:37:47.605	candle0	rx	0x001			3	42 01 44	
09:38:20.720	candle0	tx	0x001			3	91 00 92	
09:38:20.720	candle0	rx	0x001			3	91 01 93	
09:38:35.121	candle0	rx	0x001			3	91 02 94	
09:39:26.636	candle0	tx	0x001			3	91 01 93	
09:39:26.636	candle0	rx	0x001			3	91 01 93	
09:39:30.951	candle0	rx	0x001			3	91 02 94	

5.4.2 IO signal triggers return to zero

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

01 21 FF FF 06 00 07 00 CRC

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

01 93 00 00 01 EA 60 CRC

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

01 94 01 90 00 00 00 00 CRC

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

01 95 00 32 00 0A 00 C8 CRC

5. Save Parameters

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:

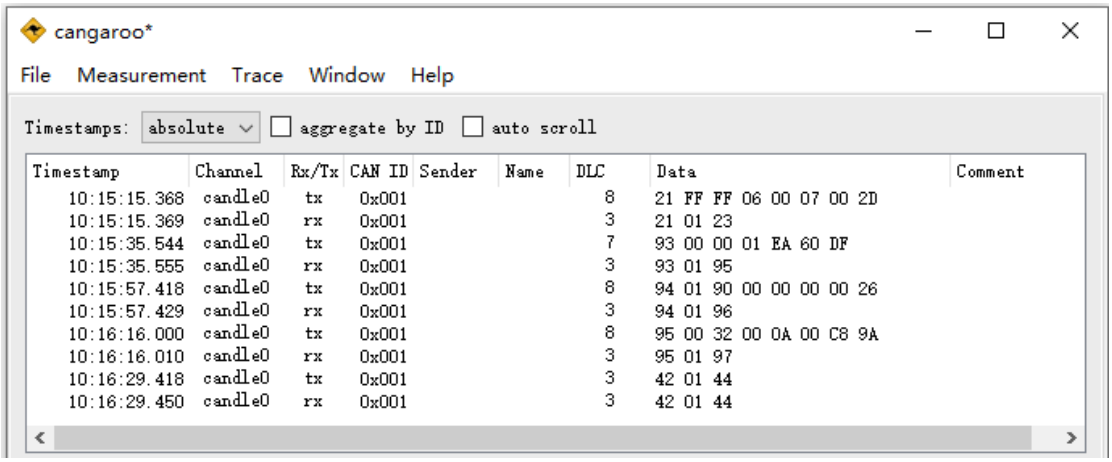
Reverse (clockwise) low speed operation → switch open → forward (counterclockwise) low speed operation → switch closed → stop.

8. Execute coordinate return to zero

Make DI3 signal effective again

Or execute instruction 01 91 01 CRC

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



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
10:15:15.368	candle0	tx	0x001			8	21 FF FF 06 00 07 00 2D	
10:15:15.369	candle0	rx	0x001			3	21 01 23	
10:15:35.544	candle0	tx	0x001			7	93 00 00 01 EA 60 DF	
10:15:35.555	candle0	rx	0x001			3	93 01 95	
10:15:57.418	candle0	tx	0x001			8	94 01 90 00 00 00 00 26	
10:15:57.429	candle0	rx	0x001			3	94 01 96	
10:16:16.000	candle0	tx	0x001			8	95 00 32 00 0A 00 C8 9A	
10:16:16.010	candle0	rx	0x001			3	95 01 97	
10:16:29.418	candle0	tx	0x001			3	42 01 44	
10:16:29.450	candle0	rx	0x001			3	42 01 44	

5.4.3 Automatically return to zero after power on

1. Configure DI2 function as "external origin switch"

01 21 FF FF 06 00 FF FF CRC

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

01 94 01 90 00 00 00 00 CRC

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

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

01 93 00 00 02 EA 60 CRC

5. Save Parameters

01 42 01 CRC

Note1: The above parameters only need to be set once and do not need to be set repeatedly.

Note2: 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.

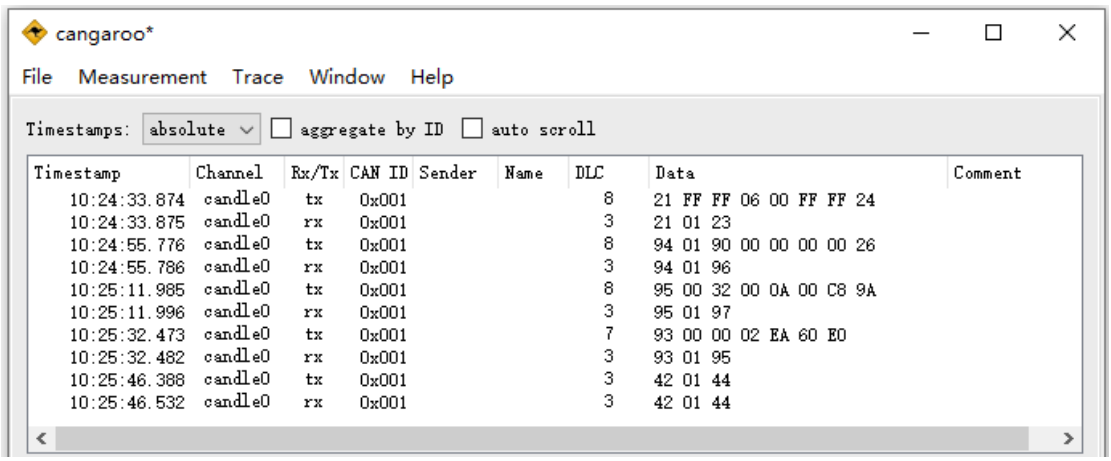
If the origin switch DI2 signal is triggered (switch closed), the motor runs as follows:

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

7. Execute coordinate return to zero

01 91 01 CRC

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



5.5 Mechanical limit origin return configuration example

5.5.1 Command triggers return to zero

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

01 93 03 00 00 EA 60 CRC

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

01 94 00 64 00 00 00 00 CRC

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

01 95 00 32 00 0A 00 C8 CRC

4. Save Parameters

01 42 01 CRC

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

5. Execute origin return

01 91 00 CRC

The motor operates as follows:

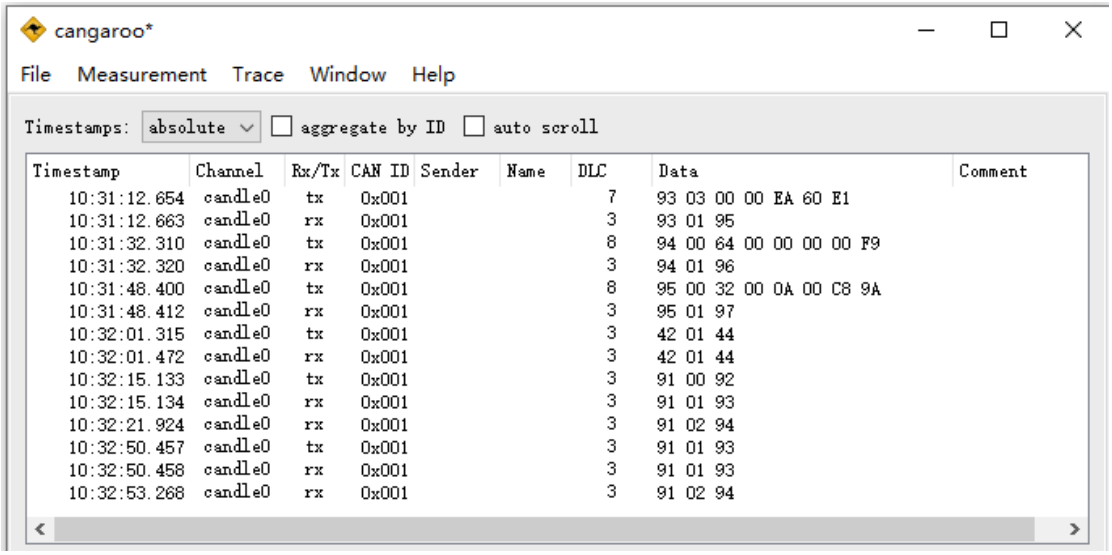
Run forward (counterclockwise) at low speed → reach mechanical limit → stop.

6. Execute coordinate return to zero

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.



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
10:31:12.654	candle0	tx	0x001			7	93 03 00 00 EA 60 E1	
10:31:12.663	candle0	rx	0x001			3	93 01 95	
10:31:32.310	candle0	tx	0x001			8	94 00 64 00 00 00 00 F9	
10:31:32.320	candle0	rx	0x001			3	94 01 96	
10:31:48.400	candle0	tx	0x001			8	95 00 32 00 0A 00 C8 9A	
10:31:48.412	candle0	rx	0x001			3	95 01 97	
10:32:01.315	candle0	tx	0x001			3	42 01 44	
10:32:01.472	candle0	rx	0x001			3	42 01 44	
10:32:15.133	candle0	tx	0x001			3	91 00 92	
10:32:15.134	candle0	rx	0x001			3	91 01 93	
10:32:21.924	candle0	rx	0x001			3	91 02 94	
10:32:50.457	candle0	tx	0x001			3	91 01 93	
10:32:50.458	candle0	rx	0x001			3	91 01 93	
10:32:53.268	candle0	rx	0x001			3	91 02 94	

5.5.2 IO signal triggers return to zero

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

01 21 FF FF FF FF 07 00 CRC

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

01 93 03 00 01 EA 60 CRC

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

01 94 00 64 00 00 00 00 CRC

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

01 95 00 32 00 0A 00 C8 CRC

5. Save Parameters

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:

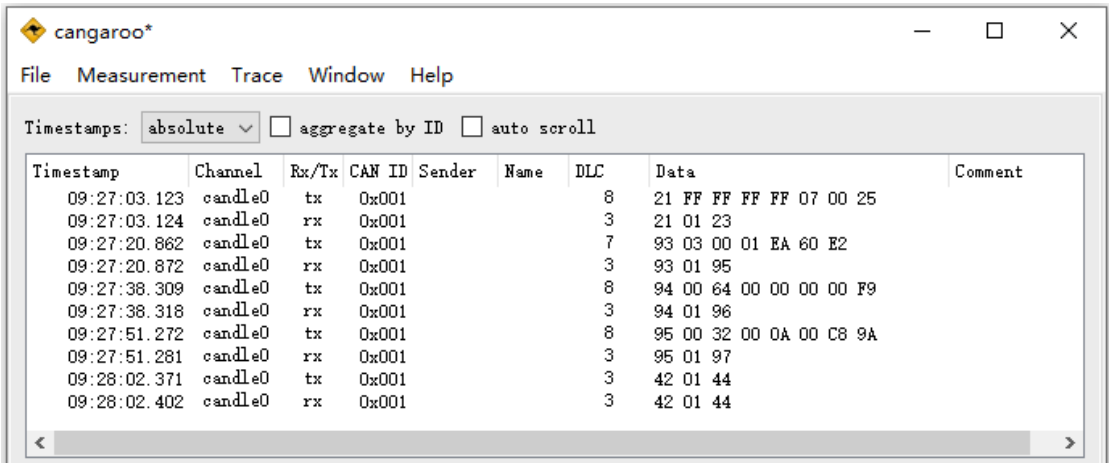
Run forward (counterclockwise) at low speed → reach mechanical limit → stop.

8. Execute coordinate return to zero

Make DI3 signal effective again

Or execute instruction :01 91 01 CRC

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



5.5.3 Automatically return to zero after power on

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

01 94 00 64 00 00 00 00 CRC

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

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

01 93 03 00 02 EA 60 CRC

4. Save Parameters

01 42 01 CRC

Note1: The above parameters only need to be set once and do not need to be set repeatedly.

Note2: 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

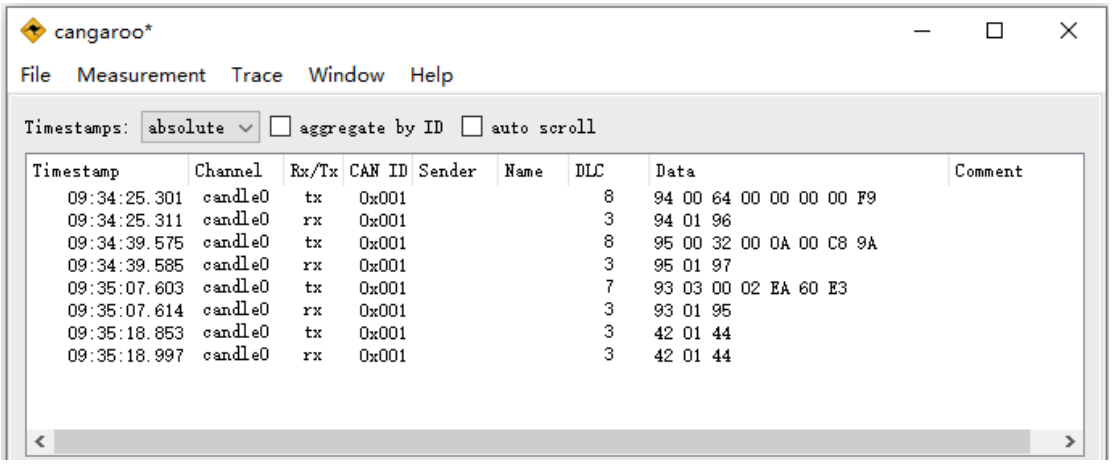
The motor operates as follows:

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

6. Execute coordinate return to zero

01 91 01 CRC

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



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

01 92 00 CRC

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

01 93 04 02 00 EA 60 CRC

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

01 95 00 32 00 0A 00 C8 CRC

4. Save Parameters

01 42 01 CRC

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

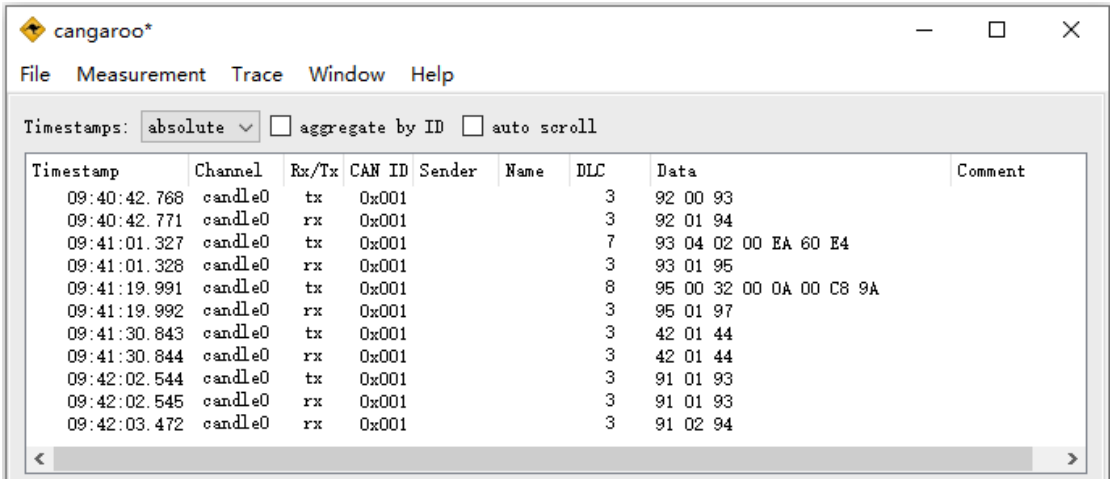
5. If the motor is not enabled, enable the motor

6. Execute single-turn coordinate return to zero

01 91 01 CRC

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
09:40:42.768	candle0	tx	0x001			3	92 00 93	
09:40:42.771	candle0	rx	0x001			3	92 01 94	
09:41:01.327	candle0	tx	0x001			7	93 04 02 00 EA 60 E4	
09:41:01.328	candle0	rx	0x001			3	93 01 95	
09:41:19.991	candle0	tx	0x001			8	95 00 32 00 0A 00 C8 9A	
09:41:19.992	candle0	rx	0x001			3	95 01 97	
09:41:30.843	candle0	tx	0x001			3	42 01 44	
09:41:30.844	candle0	rx	0x001			3	42 01 44	
09:42:02.544	candle0	tx	0x001			3	91 01 93	
09:42:02.545	candle0	rx	0x001			3	91 01 93	
09:42:03.472	candle0	rx	0x001			3	91 02 94	

5.6.2 IO signal triggers return to zero

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

01 92 00 CRC

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

01 21 FF FF FF FF 07 00 CRC

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

01 93 04 02 01 EA 60 CRC

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

01 95 00 32 00 0A 00 C8 CRC

5. Save Parameters

01 42 01 CRC

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

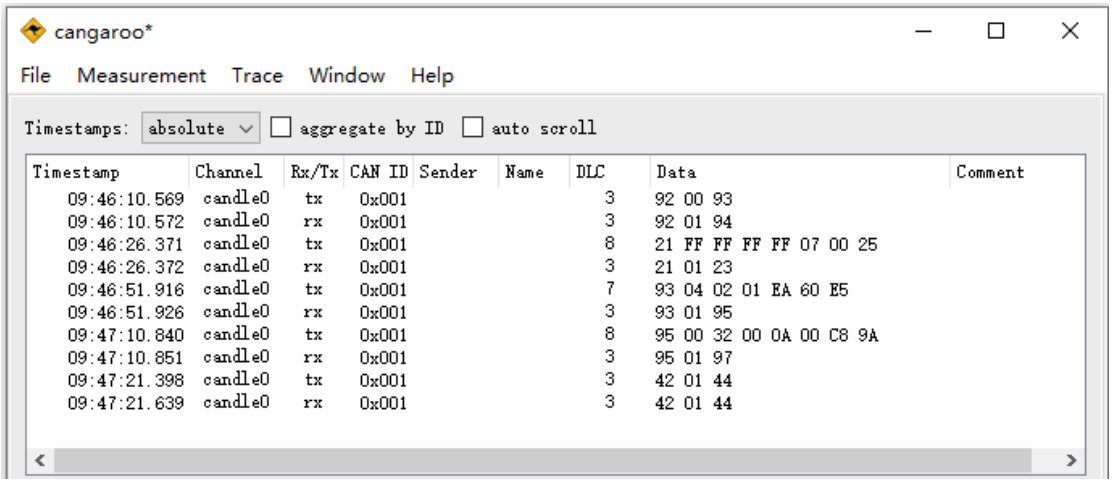
6. If the motor is not enabled, enable the motor

7. Execute single-turn return to zero

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

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



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

01 92 00 CRC

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

01 93 04 02 02 EA 60 CRC

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

01 95 00 32 00 0A 00 C8 CRC

4. Save Parameters

01 42 01 CRC

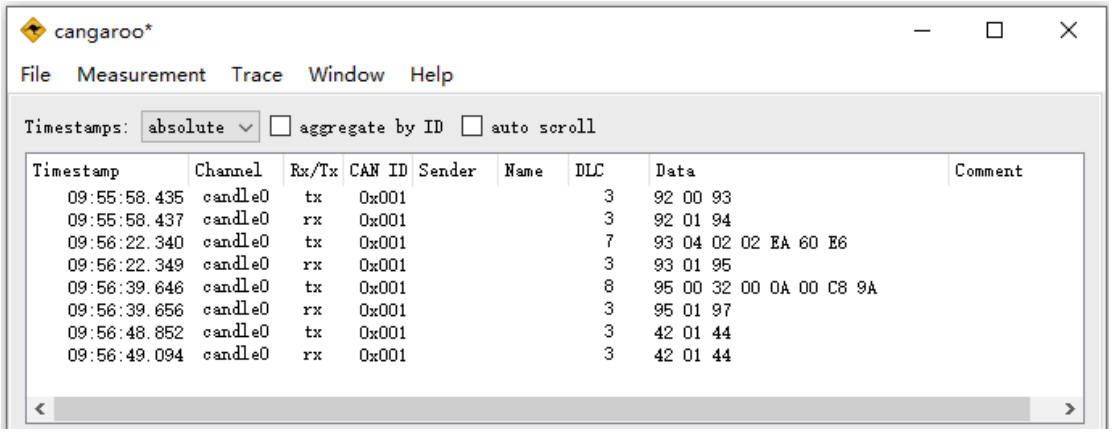
Note: The above parameters only need to be set once and there is no need to set them repeatedly.

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

Or execute instruction 01 91 01 CRC

The motor operates as follows:

High-speed operation → reach zero coordinate → stop.



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

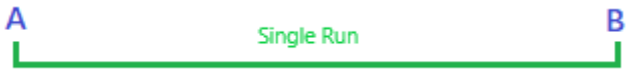
After running 200 at the relative position, the motor reaches point E.

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

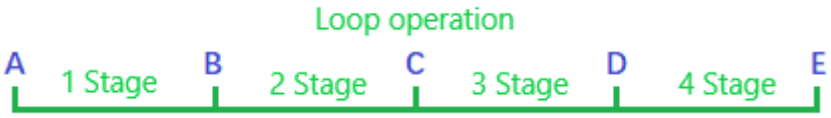
After running 100 at the absolute position, the motor reaches point C.

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)					
address		DLC	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Control Word	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	state	Checksum
			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)				
address		DLC	Byte 1	Byte 2
01	…	2	Function code	Checksum
			F1H	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	…	3	Function code	state	Checksum
			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)					
address		DLC	Byte 1	Byte 2	Byte 3
01	…	3	Function code	Control Word	Checksum
			F3H	enable	CRC

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

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	…	3	Function code	state	Checksum
			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)					
address	...	DLC	Byte 1	Byte 2	Byte 3
01		3	Function code	Control Word	Checksum
			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)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	state	Checksum
			FFH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Note1: 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.

Note2: 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)					
address		DLC	Byte 1	Bytes 2-3	Byte 4
01	...	4	Function code	Threshold	Checksum
			F2H	thereshold	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	state	Checksum
			F2H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

6.4 Position control configuration instructions

Downlink frame (PC → SmartMotor)								
address		DLC	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
01	…	6	Function code	How it works	Displacement method	Displacement Segment Number	Residue processing method	Checksum
			96H	runMode	shiftMode	shiftNum	remainMode	CRC

runMode : 00 single position run (default value)

Single multi-stage position operation

Cyclic multi-segment position operation

shiftMode : 00 relative shift (default)

01Absolute 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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	…	3	Function code	state	Checksum
			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)									
address		DLC	Byte 1	Byte 2	Byte 3		Byte 4	Bytes 5-7	Byte 8
					b7-b4	b3-b0			
1	...	8	Function code	Maximum speed	Acceleration and deceleration time			Displacement	Checksum
			FD	Speed(uint12_t)	accTime (uint12_t)			pulses(int24_t)	CRC

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

accTime : acceleration and deceleration time (range 1~4095, unit: 10ms)

pulses: displacement (range -8388608~8388607, unit: pulse unit)

Note1: accTime is the acceleration time from 0 to 1000RPM and the deceleration time from 1000 to 0RPM, the same below.

Note2: pulses are positive/negative numbers that determine the direction of motor operation.

For example, to set the maximum speed to 300 RPM, the deceleration time to 10000 ms, and the displacement to 80000 , the instructions are as follows:

01 FD 12 C3 E8 00 1F 40 CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Running status	Checksum
			FDH	status (uint8_t)	CRC

status = 0 Run failed

status = 1 Run started

status = 2 Run completed

Note: 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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

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

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

01 96 00 00 01 01 CRC

4. Save Parameters

01 42 01 CRC

5. Coordinates cleared for easy observation (optional)

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 **300** RPM, acceleration/deceleration time **1000** ms, displacement **32000** (10 turns)

01 FD 12 C0 64 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

01 31 CRC

9. During operation

01 F7 00 CRC command can terminate the motor operation

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

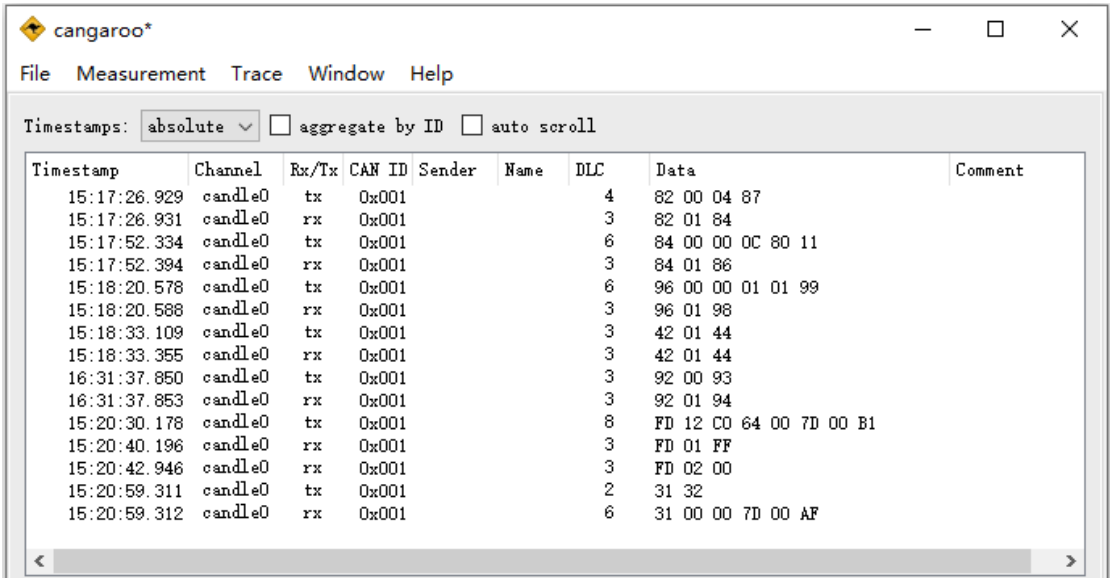
If you need to run automatically after power on, add the following configuration:

10. Set automatic operation on power on

01 FF 01 CRC

11. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
15:17:26.929	candle0	tx	0x001			4	82 00 04 87	
15:17:26.931	candle0	rx	0x001			3	82 01 84	
15:17:52.334	candle0	tx	0x001			6	84 00 00 0C 80 11	
15:17:52.394	candle0	rx	0x001			3	84 01 86	
15:18:20.578	candle0	tx	0x001			6	96 00 00 01 01 99	
15:18:20.588	candle0	rx	0x001			3	96 01 98	
15:18:33.109	candle0	tx	0x001			3	42 01 44	
15:18:33.355	candle0	rx	0x001			3	42 01 44	
16:31:37.850	candle0	tx	0x001			3	92 00 93	
16:31:37.853	candle0	rx	0x001			3	92 01 94	
15:20:30.178	candle0	tx	0x001			8	FD 12 C0 64 00 7D 00 B1	
15:20:40.196	candle0	rx	0x001			3	FD 01 FF	
15:20:42.946	candle0	rx	0x001			3	FD 02 00	
15:20:59.311	candle0	tx	0x001			2	31 32	
15:20:59.312	candle0	rx	0x001			6	31 00 00 7D 00 AF	

6.5.3 Single absolute position run configuration example

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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

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

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

01 96 00 01 01 01 CRC

4. Save Parameters

01 42 01 CRC

5. Coordinates cleared for easy observation

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 **300** RPM, acceleration/deceleration time **1000** ms, displacement **32000** (10 turns)

01 FD **12 C0 64 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

01 31 CRC

9. During operation

01 F7 00 CRC command can terminate the motor operation

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

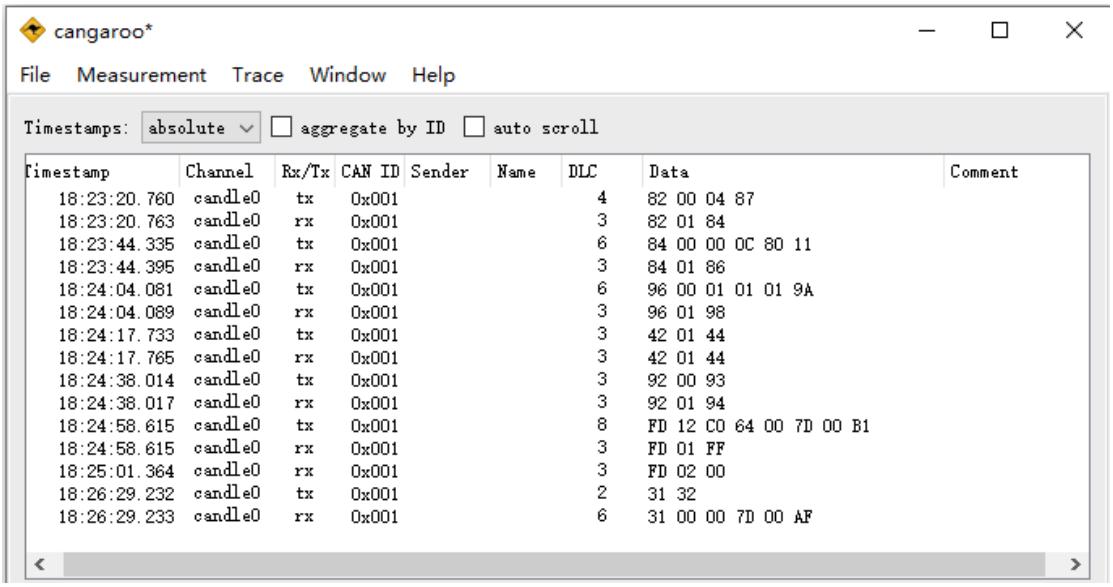
If you need to run automatically after power on, add the following configuration:

10. Set automatic operation on power on

01 FF 01 CRC

11. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
18:23:20.760	candle0	tx	0x001			4	82 00 04 87	
18:23:20.763	candle0	rx	0x001			3	82 01 84	
18:23:44.335	candle0	tx	0x001			6	84 00 00 0C 80 11	
18:23:44.395	candle0	rx	0x001			3	84 01 86	
18:24:04.081	candle0	tx	0x001			6	96 00 01 01 01 9A	
18:24:04.089	candle0	rx	0x001			3	96 01 98	
18:24:17.733	candle0	tx	0x001			3	42 01 44	
18:24:17.765	candle0	rx	0x001			3	42 01 44	
18:24:38.014	candle0	tx	0x001			3	92 00 93	
18:24:38.017	candle0	rx	0x001			3	92 01 94	
18:24:58.615	candle0	tx	0x001			8	FD 12 C0 64 00 7D 00 B1	
18:24:58.615	candle0	rx	0x001			3	FD 01 FF	
18:25:01.364	candle0	rx	0x001			3	FD 02 00	
18:26:29.232	candle0	tx	0x001			2	31 32	
18:26:29.233	candle0	rx	0x001			6	31 00 00 7D 00 AF	

6.6 Multi- position operation

6.6.1 Bit field parameter configuration instructions

1. Section 1 parameter configuration instructions

Configuration directive 1:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Maximum speed	Acceleration and deceleration time	Checksum
			97H	speed1	accTime1	CRC

speed1: maximum speed (range 1~3000, unit: RPM)

accTime1: acceleration and deceleration time (range 1~65535, unit: ms)

Note1: accTime is the acceleration time from 0 to 1000RPM and the deceleration time from 1000 to 0RPM, and the other segments are the same.

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			97H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Configuration directive 2:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-7	Byte 8
01	...	8	Function code	Waiting time after the run is completed	Displacement	Checksum
			98H	waitTime1	pulses1	CRC

waitTime1: Waiting time after the operation is completed (range 0~10000, unit: ms)

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

Note2: pulses is a positive/negative number that determines the direction of motor operation. The other segments are the same.

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			98H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Note3: 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.

2. Section 2 parameter configuration instructions

Configuration directive 1:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Maximum speed	Acceleration and deceleration time	Checksum
			99H	speed2	accTime2	CRC

speed2: maximum speed (range 1~3000, unit: RPM)

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

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			99H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Configuration directive 2:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-7	Byte 8
01	...	8	Function code	Waiting time after the run is completed	Displacement	Checksum
			9AH	waitTime2	pulses2	CRC

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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			9AH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

3. Section 3 parameter configuration instructions

Configuration directive 1:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Maximum speed	Acceleration and deceleration time	Checksum
			9BH	speed3	accTime3	CRC

speed3: maximum speed (range 1~3000, unit: RPM)

accTime3: acceleration and deceleration time (range 1~65535, unit: ms)

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			9BH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Configuration directive 2:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-7	Byte 8
01	...	8	Function code	Waiting time after the run is completed	Displacement	Checksum
			9CH	waitTime3	pulses3	CRC

waitTime3: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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			9CH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

4. Section 4 parameter configuration instructions

Configuration directive 1:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Maximum speed	Acceleration and deceleration time	Checksum
			9DH	speed4	accTime4	CRC

speed4:maximum speed (range 1~3000, unit: RPM)

accTime4: acceleration and deceleration time (range 1~65535, unit: ms)

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			9DH	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Configuration directive 2:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-7	Byte 8
01	...	8	Function code	Waiting time after the run is completed	Displacement	Checksum
			9E	waitTime4	pulses4	CRC

waitTime4: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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			9E	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)					
address	...	DLC	Byte 1	Byte 2	Byte 6
01		3	Function code	Control Word	Checksum
			FEH	01	CRC

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Running status	Checksum
			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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

01 84 00 00 0C 80 CRC

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

01 96 01 00 04 01 CRC

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

01 97 00 C8 03 E8 CRC

5. Set 98H parameters: Parameters for the first section (2000 ms, 32000 pulses)

01 98 07 D0 00 00 7D 00 CRC

6. Set 99H parameters: Parameters for the second section (300 RPM, 100 ms)

01 99 01 2C 00 64 CRC

7. Set 9AH parameters: Parameters for the second section (2000 ms, -32000 pulses)

01 9A 07 D0 FF FF 83 00 CRC

8. Set 9BH parameters: 3rd section parameters (600 RPM, 2000 ms)

01 9B 02 58 07 D0 CRC

9. Set 9CH parameters: 3rd section parameters (500 ms, 128000 pulses)

01 9C 01 F4 00 01 F4 00 CRC

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

01 9D 03 E8 00 64 CRC

11. Set 9EH parameters: 4th section parameters (10 ms, -128000 pulses)

01 9E 00 0A FF FE 0C 00 CRC

12. Save Parameters

01 42 01 CRC

13. Coordinates cleared for easy observation (optional)

01 92 00 CRC

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

14. Control motor operation

01 FE 01 CRC

15. The motor starts running.

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

01 31 CRC

17. During operation

01 F7 00 CRC command can terminate the motor operation

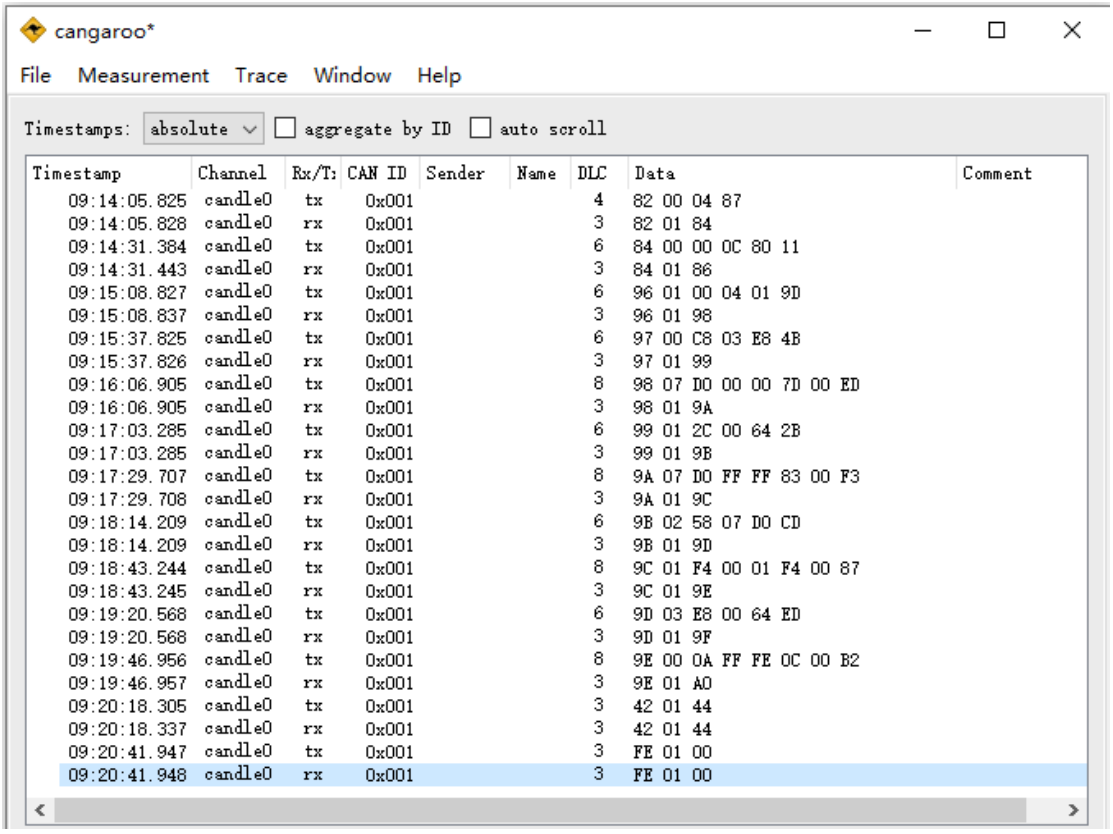
18. If you need to run automatically after power on, add the following configuration:

19. Set automatic operation on power on

01 FF 01 CRC

20. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
09:14:05.825	candle0	tx	0x001			4	82 00 04 87	
09:14:05.828	candle0	rx	0x001			3	82 01 84	
09:14:31.384	candle0	tx	0x001			6	84 00 00 0C 80 11	
09:14:31.443	candle0	rx	0x001			3	84 01 86	
09:15:08.827	candle0	tx	0x001			6	96 01 00 04 01 9D	
09:15:08.837	candle0	rx	0x001			3	96 01 98	
09:15:37.825	candle0	tx	0x001			6	97 00 C8 03 E8 4B	
09:15:37.826	candle0	rx	0x001			3	97 01 99	
09:16:06.905	candle0	tx	0x001			8	98 07 D0 00 00 7D 00 ED	
09:16:06.905	candle0	rx	0x001			3	98 01 9A	
09:17:03.285	candle0	tx	0x001			6	99 01 2C 00 64 2B	
09:17:03.285	candle0	rx	0x001			3	99 01 9B	
09:17:29.707	candle0	tx	0x001			8	9A 07 D0 FF FF 83 00 F3	
09:17:29.708	candle0	rx	0x001			3	9A 01 9C	
09:18:14.209	candle0	tx	0x001			6	9B 02 58 07 D0 CD	
09:18:14.209	candle0	rx	0x001			3	9B 01 9D	
09:18:43.244	candle0	tx	0x001			8	9C 01 F4 00 01 F4 00 87	
09:18:43.245	candle0	rx	0x001			3	9C 01 9E	
09:19:20.568	candle0	tx	0x001			6	9D 03 E8 00 64 ED	
09:19:20.568	candle0	rx	0x001			3	9D 01 9F	
09:19:46.956	candle0	tx	0x001			8	9E 00 0A FF FE 0C 00 B2	
09:19:46.957	candle0	rx	0x001			3	9E 01 A0	
09:20:18.305	candle0	tx	0x001			3	42 01 44	
09:20:18.337	candle0	rx	0x001			3	42 01 44	
09:20:41.947	candle0	tx	0x001			3	FE 01 00	
09:20:41.948	candle0	rx	0x001			3	FE 01 00	

6.6.4 Single multi-segment absolute position operation configuration example

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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

01 84 00 00 0C 80 CRC

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

01 96 01 01 04 01 CRC

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

01 97 00 C8 03 E8 CRC

5. Set 98H parameters: Parameters for the first section (2000 ms, 32000 pulses)

01 98 07 D0 00 00 7D 00 CRC

6. Set 99H parameters: Parameters for the second section (300 RPM, 100 ms)

01 99 01 2C 00 64 CRC

7. Set 9AH parameters: Parameters for the second section (2000 ms, -32000 pulses)

01 9A 07 D0 FF FF 83 00 CRC

8. Set 9BH parameters: 3rd section parameters (600 RPM, 2000 ms)

01 9B 02 58 07 D0 CRC

9. Set 9CH parameters: 3rd section parameters (500 ms, 128000 pulses)

01 9C 01 F4 00 01 F4 00 CRC

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

01 9D 03 E8 00 64 CRC

11. Set 9EH parameters: 4th section parameters (10 ms, 0 pulses)

01 9E 00 0A 00 00 00 00 CRC

12. Save Parameters

01 42 01 CRC

13. Coordinates cleared for easy observation (optional)

01 92 00 CRC

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

14. Control motor operation

01 FE 01 CRC

15. The motor starts running.

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

01 31 CRC

17. During operation

01 F7 00 CRC command can terminate the motor operation

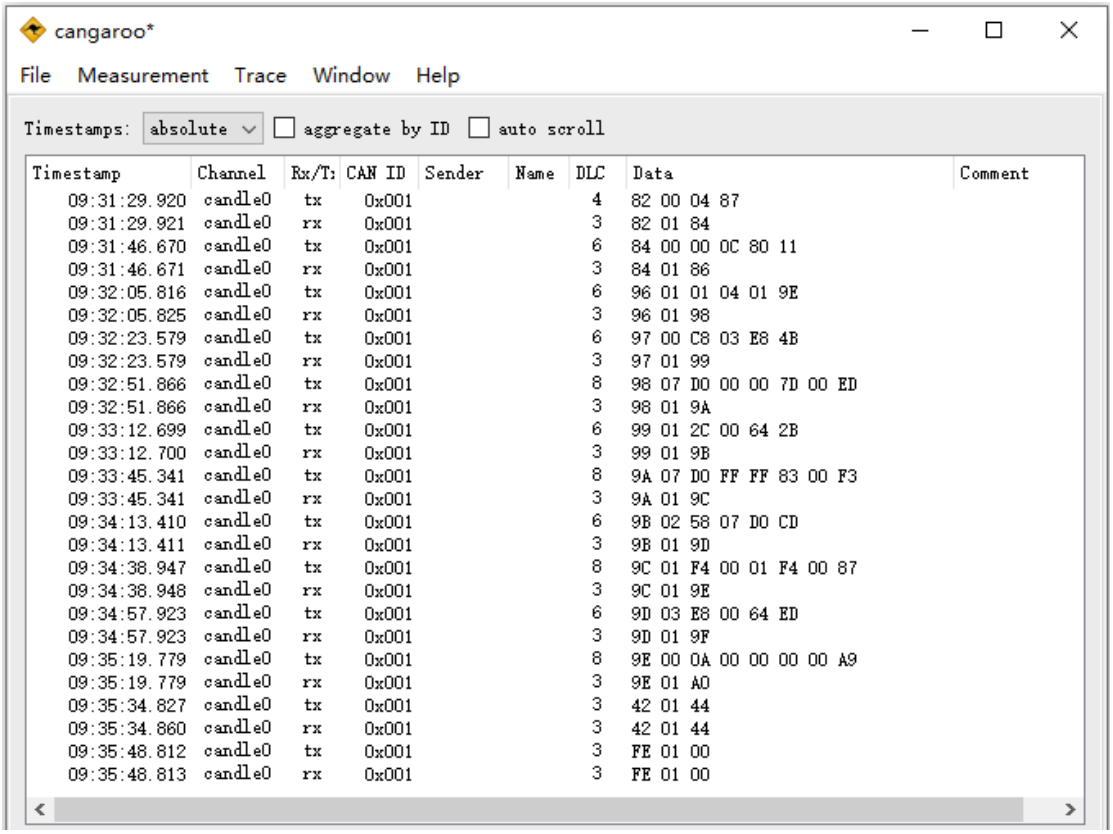
If you need to run automatically after power on, add the following configuration:

18. Set automatic operation on power on

01 FF 01 CRC

19. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
09:31:29.920	candle0	tx	0x001			4	82 00 04 87	
09:31:29.921	candle0	rx	0x001			3	82 01 84	
09:31:46.670	candle0	tx	0x001			6	84 00 00 0C 80 11	
09:31:46.671	candle0	rx	0x001			3	84 01 86	
09:32:05.816	candle0	tx	0x001			6	96 01 01 04 01 9E	
09:32:05.825	candle0	rx	0x001			3	96 01 98	
09:32:23.579	candle0	tx	0x001			6	97 00 C8 03 E8 4B	
09:32:23.579	candle0	rx	0x001			3	97 01 99	
09:32:51.866	candle0	tx	0x001			8	98 07 D0 00 00 7D 00 ED	
09:32:51.866	candle0	rx	0x001			3	98 01 9A	
09:33:12.699	candle0	tx	0x001			6	99 01 2C 00 64 2B	
09:33:12.700	candle0	rx	0x001			3	99 01 9B	
09:33:45.341	candle0	tx	0x001			8	9A 07 D0 FF FF 83 00 F3	
09:33:45.341	candle0	rx	0x001			3	9A 01 9C	
09:34:13.410	candle0	tx	0x001			6	9B 02 58 07 D0 CD	
09:34:13.411	candle0	rx	0x001			3	9B 01 9D	
09:34:38.947	candle0	tx	0x001			8	9C 01 F4 00 01 F4 00 87	
09:34:38.948	candle0	rx	0x001			3	9C 01 9E	
09:34:57.923	candle0	tx	0x001			6	9D 03 E8 00 64 ED	
09:34:57.923	candle0	rx	0x001			3	9D 01 9F	
09:35:19.779	candle0	tx	0x001			8	9E 00 0A 00 00 00 00 A9	
09:35:19.779	candle0	rx	0x001			3	9E 01 A0	
09:35:34.827	candle0	tx	0x001			3	42 01 44	
09:35:34.860	candle0	rx	0x001			3	42 01 44	
09:35:48.812	candle0	tx	0x001			3	FE 01 00	
09:35:48.813	candle0	rx	0x001			3	FE 01 00	

6.6.5 Example of loop multi-segment relative position operation configuration

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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

01 84 00 00 0C 80 CRC

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

01 96 02 00 04 01 CRC

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

01 97 **00 C8 03 E8** CRC

5. Set 98H parameters: Parameters for the first section (2000 ms, 32000 pulses)

01 98 **07 D0 00 00 7D 00** CRC

6. Set 99H parameters: Parameters for the second section (300 RPM, 100 ms)

01 99 **01 2C 00 64** CRC

7. Set 9AH parameters: Parameters for the second section (2000 ms, -32000 pulses)

01 9A **07 D0 FF FF 83 00** CRC

8. Set 9BH parameters: 3rd section parameters (600 RPM, 2000 ms)

01 9B **02 58 07 D0** CRC

9. Set 9CH parameters: 3rd section parameters (500 ms, 128000 pulses)

01 9C **01 F4 00 01 F4 00** CRC

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

01 9D **03 E8 00 64** CRC

11. Set 9EH parameters: 4th section parameters (10 ms, -128000 pulses)

01 9E **00 0A FF FE 0C 00** CRC

12. Save Parameters

01 42 01 CRC

13. Coordinates cleared for easy observation (optional)

01 92 00 CRC

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

14. Control motor operation

01 FE 01 CRC

15. The motor starts running.

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

01 31 CRC

17. During operation

01 F7 00 CRC command can terminate the motor operation

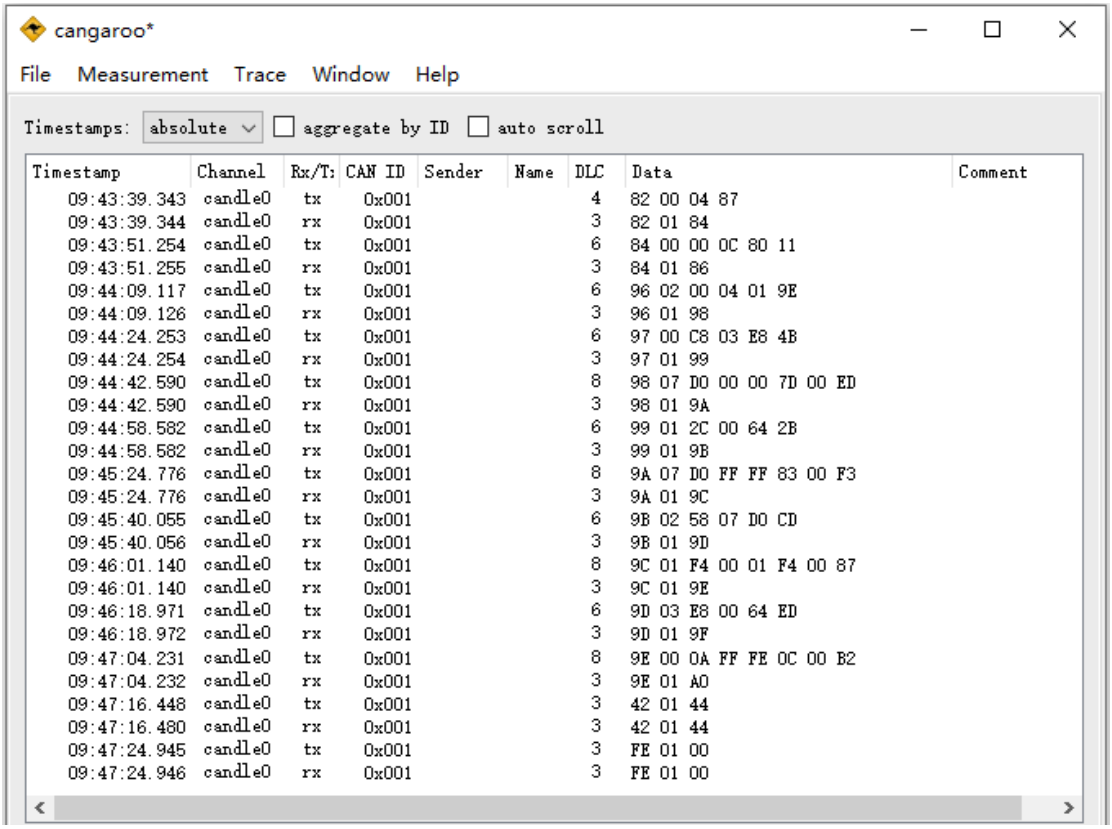
If you need to run automatically after power on, add the following configuration:

18. Set automatic operation on power on

01 FF 01 CRC

19. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
09:43:39.343	candle0	tx	0x001			4	82 00 04 87	
09:43:39.344	candle0	rx	0x001			3	82 01 84	
09:43:51.254	candle0	tx	0x001			6	84 00 00 0C 80 11	
09:43:51.255	candle0	rx	0x001			3	84 01 86	
09:44:09.117	candle0	tx	0x001			6	96 02 00 04 01 9E	
09:44:09.126	candle0	rx	0x001			3	96 01 98	
09:44:24.253	candle0	tx	0x001			6	97 00 C8 03 E8 4B	
09:44:24.254	candle0	rx	0x001			3	97 01 99	
09:44:42.590	candle0	tx	0x001			8	98 07 D0 00 00 7D 00 ED	
09:44:42.590	candle0	rx	0x001			3	98 01 9A	
09:44:58.582	candle0	tx	0x001			6	99 01 2C 00 64 2B	
09:44:58.582	candle0	rx	0x001			3	99 01 9B	
09:45:24.776	candle0	tx	0x001			8	9A 07 D0 FF FF 83 00 F3	
09:45:24.776	candle0	rx	0x001			3	9A 01 9C	
09:45:40.055	candle0	tx	0x001			6	9B 02 58 07 D0 CD	
09:45:40.056	candle0	rx	0x001			3	9B 01 9D	
09:46:01.140	candle0	tx	0x001			8	9C 01 F4 00 01 F4 00 87	
09:46:01.140	candle0	rx	0x001			3	9C 01 9E	
09:46:18.971	candle0	tx	0x001			6	9D 03 E8 00 64 ED	
09:46:18.972	candle0	rx	0x001			3	9D 01 9F	
09:47:04.231	candle0	tx	0x001			8	9E 00 0A FF FE 0C 00 B2	
09:47:04.232	candle0	rx	0x001			3	9E 01 A0	
09:47:16.448	candle0	tx	0x001			3	42 01 44	
09:47:16.480	candle0	rx	0x001			3	42 01 44	
09:47:24.945	candle0	tx	0x001			3	FE 01 00	
09:47:24.946	candle0	rx	0x001			3	FE 01 00	

6.6.6 Example of cyclic multi-segment absolute position operation configuration

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

01 82 00 04 CRC

2. Set 84H parameters: 3200 pulses/circle

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

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

01 96 02 01 04 01 CRC

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

01 97 **00 C8 03 E8** CRC

5. Set 98H parameters: Parameters for the first section (2000 ms, 32000 pulses)

01 98 **07 D0 00 00 7D 00** CRC

6. Set 99H parameters: Parameters for the second section (300 RPM, 100 ms)

01 99 **01 2C 00 64** CRC

7. Set 9AH parameters: Parameters for the second section (2000 ms, -32000 pulses)

01 9A **07 D0 FF FF 83 00** CRC

8. Set 9BH parameters: 3rd section parameters (600 RPM, 2000 ms)

01 9B **02 58 07 D0** CRC

9. Set 9CH parameters: 3rd section parameters (500 ms, 128000 pulses)

01 9C **01 F4 00 01 F4 00** CRC

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

01 9D **03 E8 00 64** CRC

11. Set 9EH parameters: 4th section parameters (10 ms, 0 pulses)

01 9E **00 0A 00 00 00 00** CRC

12. Save Parameters

01 42 01 CRC

13. Coordinates cleared for easy observation (optional)

01 92 00 CRC

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

14. Control motor operation

01 FE 01 CRC

15. The motor starts running.

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

01 31 CRC

17. During operation

01 F7 00 CRC command can terminate the motor operation

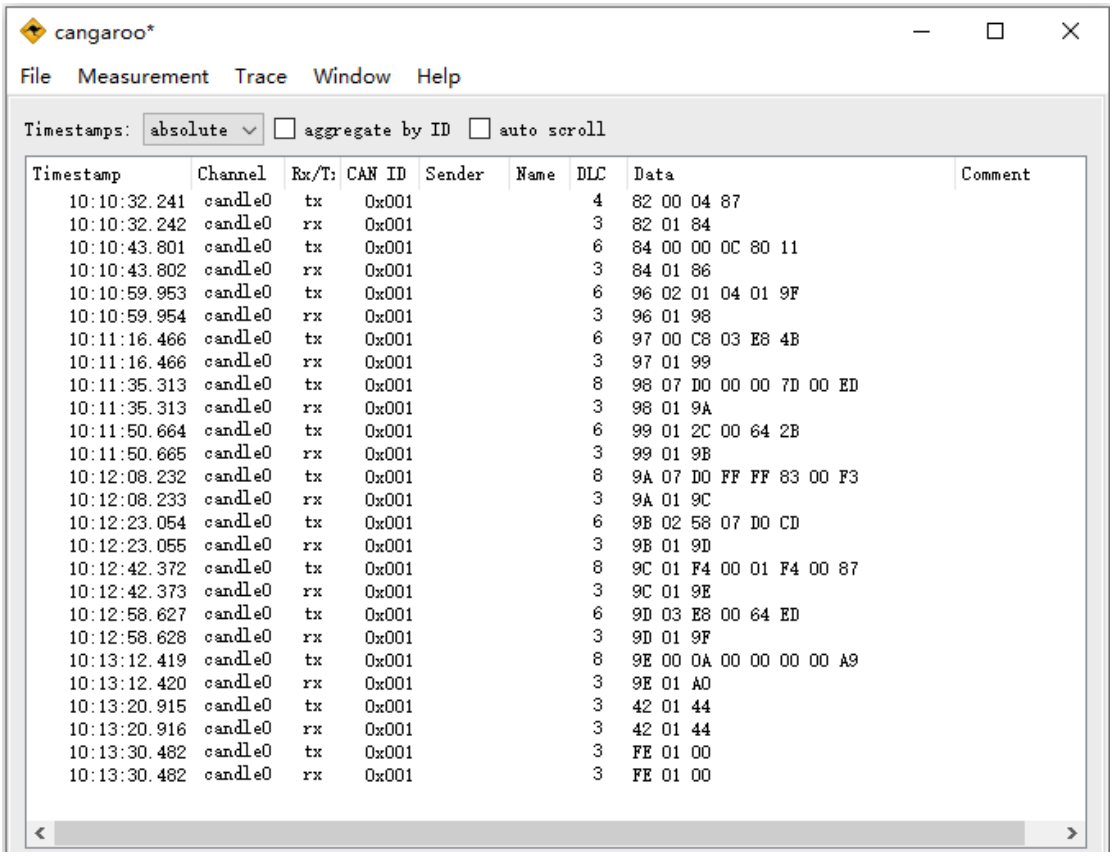
If you need to run automatically after power on, add the following configuration:

18. Set automatic operation on power on

01 FF 01 CRC

19. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
10:10:32.241	candle0	tx	0x001			4	82 00 04 87	
10:10:32.242	candle0	rx	0x001			3	82 01 84	
10:10:43.801	candle0	tx	0x001			6	84 00 00 0C 80 11	
10:10:43.802	candle0	rx	0x001			3	84 01 86	
10:10:59.953	candle0	tx	0x001			6	96 02 01 04 01 9F	
10:10:59.954	candle0	rx	0x001			3	96 01 98	
10:11:16.466	candle0	tx	0x001			6	97 00 C8 03 E8 4B	
10:11:16.466	candle0	rx	0x001			3	97 01 99	
10:11:35.313	candle0	tx	0x001			8	98 07 D0 00 00 7D 00 ED	
10:11:35.313	candle0	rx	0x001			3	98 01 9A	
10:11:50.664	candle0	tx	0x001			6	99 01 2C 00 64 2B	
10:11:50.665	candle0	rx	0x001			3	99 01 9B	
10:12:08.232	candle0	tx	0x001			8	9A 07 D0 FF FF 83 00 F3	
10:12:08.233	candle0	rx	0x001			3	9A 01 9C	
10:12:23.054	candle0	tx	0x001			6	9B 02 58 07 D0 CD	
10:12:23.055	candle0	rx	0x001			3	9B 01 9D	
10:12:42.372	candle0	tx	0x001			8	9C 01 F4 00 01 F4 00 87	
10:12:42.373	candle0	rx	0x001			3	9C 01 9E	
10:12:58.627	candle0	tx	0x001			6	9D 03 E8 00 64 ED	
10:12:58.628	candle0	rx	0x001			3	9D 01 9F	
10:13:12.419	candle0	tx	0x001			8	9E 00 0A 00 00 00 00 A9	
10:13:12.420	candle0	rx	0x001			3	9E 01 A0	
10:13:20.915	candle0	tx	0x001			3	42 01 44	
10:13:20.916	candle0	rx	0x001			3	42 01 44	
10:13:30.482	candle0	tx	0x001			3	FE 01 00	
10:13:30.482	candle0	rx	0x001			3	FE 01 00	

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

01 96 01 01 01 01 CRC

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

01 97 **00 C8 03 E8** CRC

3. Set the first motor 98H parameters: the first section parameters (2000 , 32000)

01 98 **07 D0 00 00 7D 00** CRC

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

02 96 01 01 01 01 CRC

5. Set the second motor 97H parameters: Parameters of the first section (200 , 1000)

02 97 **00 C8 03 E8** CRC

6. Set the second motor 98H parameters: 1st section parameters (2000 , 32000)

02 98 **07 D0 00 00 7D 00** CRC

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

0n 96 01 01 01 01 CRC

8. Set the nth motor 97H parameters: 1st section parameters (200 , 1000)

0n 97 **00 C8 03 E8** CRC

9. Set the nth motor 98H parameters: 1st section parameters (2000 , 32000)

0n 98 **07 D0 00 00 7D 00** CRC

10. Broadcast command controls motor operation

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

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)							
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Bytes 6-7	Byte 8
01	...	8	Function code	Speed	Acceleration time	Deceleration time	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Running status	Checksum
			F6H	status (uint8_t)	CRC

status = 0 Run failed

status = 1 Run started

Note: F7H command can stop speed mode operation.

7.3 Single -speed mode operation configuration example

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

01 82 01 00 CRC

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

01 F6 **00 C8** **03 E8** **03 E8** CRC

3. Observe the motor operation

4. Stop the motor

01 F7 00 CRC

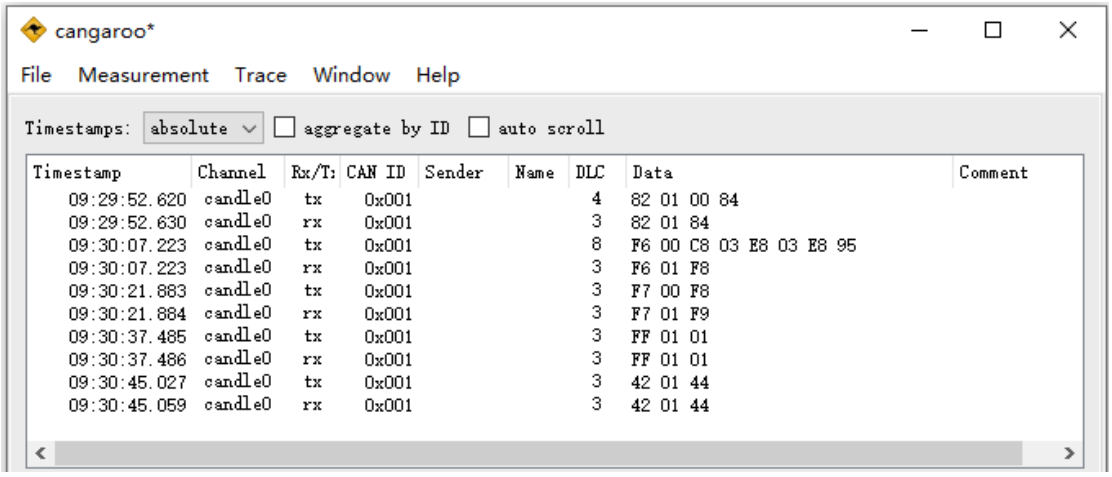
If you need to run automatically after power on, add the following configuration:

5. Set automatic operation on power on

01 FF 01 CRC

6. Save Parameters

01 42 01 CRC



7.4 Multi- speed mode operation

7.4.1 Speed segment parameter configuration instructions

1. Common configuration instructions

Downlink frame (PC → SmartMotor)								
address		DLC	Byte 1	Byte 2	Byte 3	Bytes 4-5	Bytes 6-7	Byte 8
01	...	8	Function code	model	Number of segments	Acceleration time	Deceleration time	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			A0H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

2. Section 1 parameter configuration instructions

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Running speed	Run time	Checksum
			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)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			A1H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

3. Section 2 parameter configuration instructions

Downlink frame (PC → SmartMotor)						
address	...	DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01		6	Function code	Running speed	Run time	Checksum
			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)

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			A2H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

4. Section 3 parameter configuration instructions

Downlink frame (PC → SmartMotor)						
address	...	DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01		6	Function code	Running speed	Run time	Checksum
			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)

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			A3H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

5. Section 4 parameter configuration instructions

Downlink frame (PC → SmartMotor)						
address	...	DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01		6	Function code	Running speed	Run time	Checksum
			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)

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			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

01 82 01 01 CRC

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

01 A0 00 04 00 64 00 64 CRC

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

01 A1 **00 32 00 64** CRC

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

01 A2 **FF CE 00 64** CRC

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

01 A3 **00 C8 00 64** CRC

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

01 A4 **FF 38 00 64** CRC

7. Enable motor operation

01 F3 01 CRC

8. Observe the motor operation

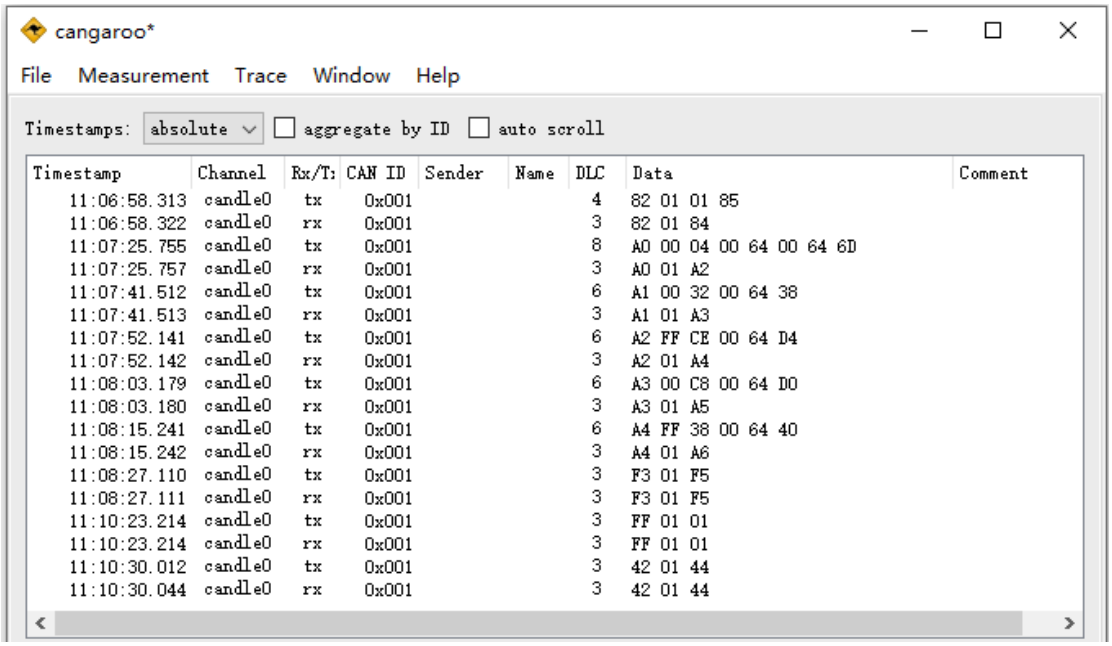
If you need to run automatically after power on, add the following configuration:

9. Set automatic operation on power on

01 FF 01 CRC

10. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
11:06:58.313	candle0	tx	0x001			4	82 01 01 85	
11:06:58.322	candle0	rx	0x001			3	82 01 84	
11:07:25.755	candle0	tx	0x001			8	A0 00 04 00 64 00 64 6D	
11:07:25.757	candle0	rx	0x001			3	A0 01 A2	
11:07:41.512	candle0	tx	0x001			6	A1 00 32 00 64 38	
11:07:41.513	candle0	rx	0x001			3	A1 01 A3	
11:07:52.141	candle0	tx	0x001			6	A2 FF CE 00 64 D4	
11:07:52.142	candle0	rx	0x001			3	A2 01 A4	
11:08:03.179	candle0	tx	0x001			6	A3 00 C8 00 64 D0	
11:08:03.180	candle0	rx	0x001			3	A3 01 A5	
11:08:15.241	candle0	tx	0x001			6	A4 FF 38 00 64 40	
11:08:15.242	candle0	rx	0x001			3	A4 01 A6	
11:08:27.110	candle0	tx	0x001			3	F3 01 F5	
11:08:27.111	candle0	rx	0x001			3	F3 01 F5	
11:10:23.214	candle0	tx	0x001			3	FF 01 01	
11:10:23.214	candle0	rx	0x001			3	FF 01 01	
11:10:30.012	candle0	tx	0x001			3	42 01 44	
11:10:30.044	candle0	rx	0x001			3	42 01 44	

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 directive 1:

Downlink frame (PC → SmartMotor)						
address	...	DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01		6	Function code	Output torque	Filter constant	Checksum
			A6H	torque	filterTime	CRC

Torque: output torque (range -3000~3000, unit: 0.1% rated torque)

filterTime : filter constant (range 0~3000, unit: 0.01ms)

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

Return data:

Uplink frame (PC ← SmartMotor)					
address	...	Data length	Byte 1	Byte 2	Byte 3
01		3	Function code	Set Status	Checksum
			A6H	status (uint8_t)	CRC

status = 0 Setting failed

status = 1 Setting successful

Configuration directive 2:

Downlink frame (PC → SmartMotor)						
address		DLC	Byte 1	Bytes 2-3	Bytes 4-5	Byte 6
01	...	6	Function code	Forward maximum speed limit	Negative maximum speed limit	Checksum
			7H	forward MaxSpeed	backward MaxSpeed	CRC

forwardMaxSpeed : Forward maximum speed limit (range 0~3000, unit: RPM)
backwardMaxSpeed : Negative maximum speed limit (range 0~3000, unit: RPM)

Return data:

Uplink frame (PC ← SmartMotor)					
address		Data length	Byte 1	Byte 2	Byte 3
01	...	3	Function code	Set Status	Checksum
			7H	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

01 82 02 00 CRC

2. Set A6H parameters: torque 100, filter constant 80

01 A6 00 64 00 50 CRC

3. Set A7H parameters: forward maximum speed 60, negative maximum speed 30

01 A7 00 3C 00 1E CRC

4. Enable motor operation

01 F3 01 CRC

5. Stop the motor

01 F7 00 CRC

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

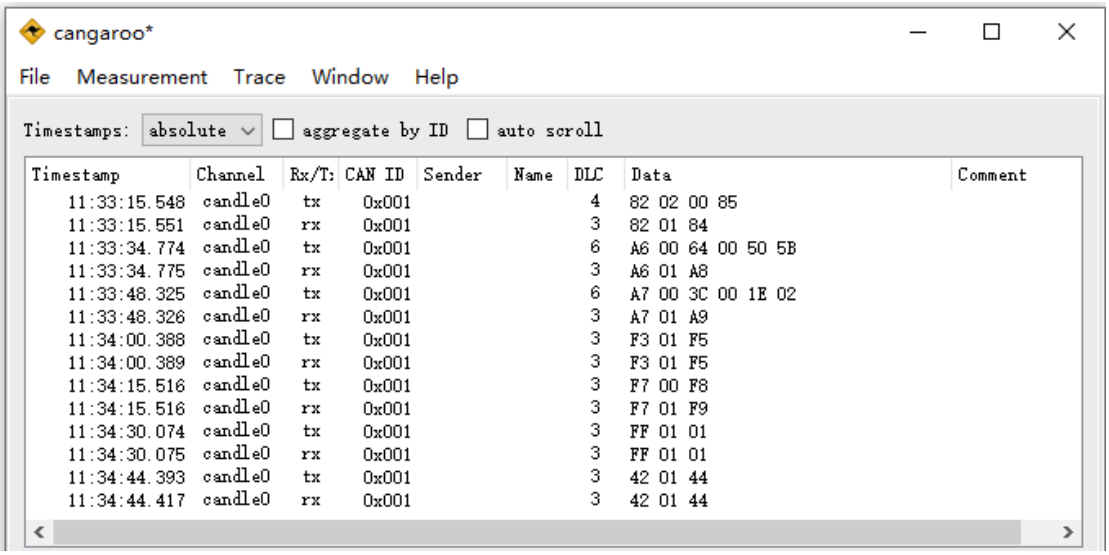
If you need to run automatically after power on, add the following configuration:

6. Configure automatic operation at power on

01 FF 01 CRC

7. Save Parameters

01 42 01 CRC



Timestamp	Channel	Rx/T:	CAN ID	Sender	Name	DLC	Data	Comment
11:33:15.548	candle0	tx	0x001			4	82 02 00 85	
11:33:15.551	candle0	rx	0x001			3	82 01 84	
11:33:34.774	candle0	tx	0x001			6	A6 00 64 00 50 5B	
11:33:34.775	candle0	rx	0x001			3	A6 01 A8	
11:33:48.325	candle0	tx	0x001			6	A7 00 3C 00 1E 02	
11:33:48.326	candle0	rx	0x001			3	A7 01 A9	
11:34:00.388	candle0	tx	0x001			3	F3 01 F5	
11:34:00.389	candle0	rx	0x001			3	F3 01 F5	
11:34:15.516	candle0	tx	0x001			3	F7 00 F8	
11:34:15.516	candle0	rx	0x001			3	F7 01 F9	
11:34:30.074	candle0	tx	0x001			3	FF 01 01	
11:34:30.075	candle0	rx	0x001			3	FF 01 01	
11:34:44.393	candle0	tx	0x001			3	42 01 44	
11:34:44.417	candle0	rx	0x001			3	42 01 44	

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

Serial number	question	Workaround
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

Part10. After-sales and technical support

https://makerbase.aliexpress.com/

Part11. Fault code 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