MKS SERVO42E/57E_CAN V1.0.1

USER MANUAL

MKS SERVO42E/57E_CAN Manual Release			
manual	discription	firmware	date
V1.0.0	First release	V1.0.0	Dec-2024
V1.0.1	Fix some bug	V1.0.1	Feb-2025

Part1. Product Overview

1.1 Introduction

MKS SERVO42E/57E_CAN closed-loop stepper motor is a product independently developed by the Maker Base to meet market demand and in accordance with industrial standards. It has a pulse interface and an RS485 interface, a built-in efficient FOC vector algorithm, and a high-precision encoder. Through position feedback, it effectively prevents the motor from losing steps. It is suitable for small robotic arms, 3D printers, engraving machines, writers, automation products, and e-sports applications.

1.2 Features

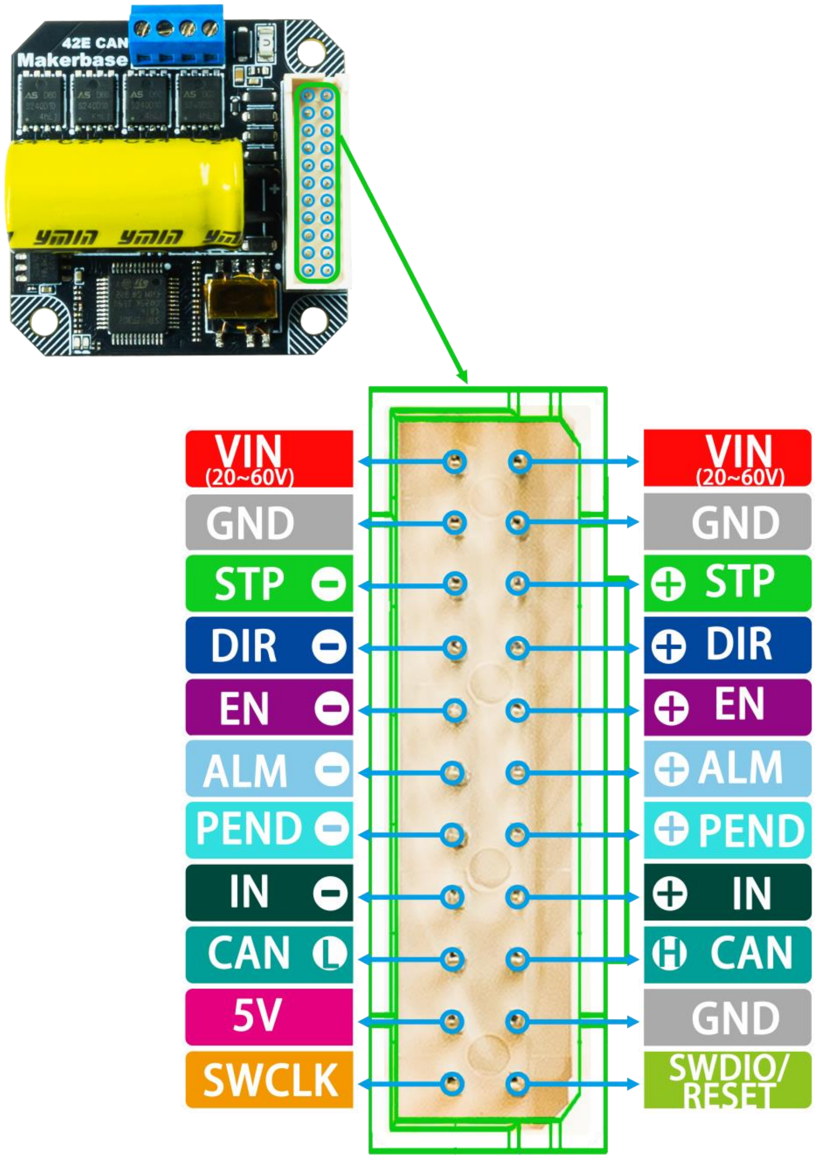
1. Support 6 working modes: pulse interface (pulse-pulse, pulse-direction mode), serial interface (open loop, closed loop mode);
2. High-performance FOC vector control algorithm, torque, speed, position three-loop control;
3. Support curve acceleration and deceleration, motor start and stop more smoothly;
4. Support single-turn unlimited position zeroing function;
5. Support multi-turn limited position zeroing function;
6. Support direct setting of zero point function;
7. Support relative position and absolute position control mode;
8. Built-in 256-step subdivision interpolation algorithm, the motor runs super quiet and ultra-low vibration;
9. Maximum input pulse frequency 300KHz , maximum speed 3000RPM;
10. Real-time update of motor angle information (motor enabled or disabled);
11. Built-in stall protection function;
12. Quickly restore factory configuration function;
13. Stable high-speed performance, smooth operation, no jitter, and emergency stop;
14. Integrated aluminum alloy shell, effective heat dissipation, and more stable continuous high-current operation of the motor;
15. Provide host computer (open source), STM32/Arduion usage routines
16. Support left and right limit functions.
17. Onboard industrial-grade high-precision magnetic encoder;
18. Onboard high-power MOSFET, 100V/25A;
19. Onboard CAN interface, 2048 slave addresses, support broadcast address and group address;
20. Industrial-grade selection design, stable and reliable;
21. Working voltage 20V~60V;
22.

1.3 Parameters

Parameters		
Motherboard model	MKS SERVO42E V1.0	MKS SERVO57E V1.0
MCU	STM32F302RBT6(Cortex-M4)	STM32F302RBT6(Cortex-M4)
MOSFET	ASS240D10Q1M	
Magnetic encoder	MT6816	
CAN transceiver	TJA1051T	
Operating Voltage	20-60V	
Working current	0-3000mA	0-5200mA
Closed loop feedback frequency	Torque loop 20KHz	
	Speedloop 10KHz	
	Position loop 10KHz	
Maximum speed	3000RPM+	
Segmentation support	2/4/8/16/32/64/128/5/10/20/25/40/50/100/200/256	
Silent/vibrate	Ultra-quiet	
Motortemperature	Motor does not heat up	
Pulse signal input	3.3V-24V (common anode)	
Pulse signal frequency	Up to 300KHz	
CAN interface rate	125K/250K/500K/1M	
can interface address	1 broadcast address, 2047 slave addresses	

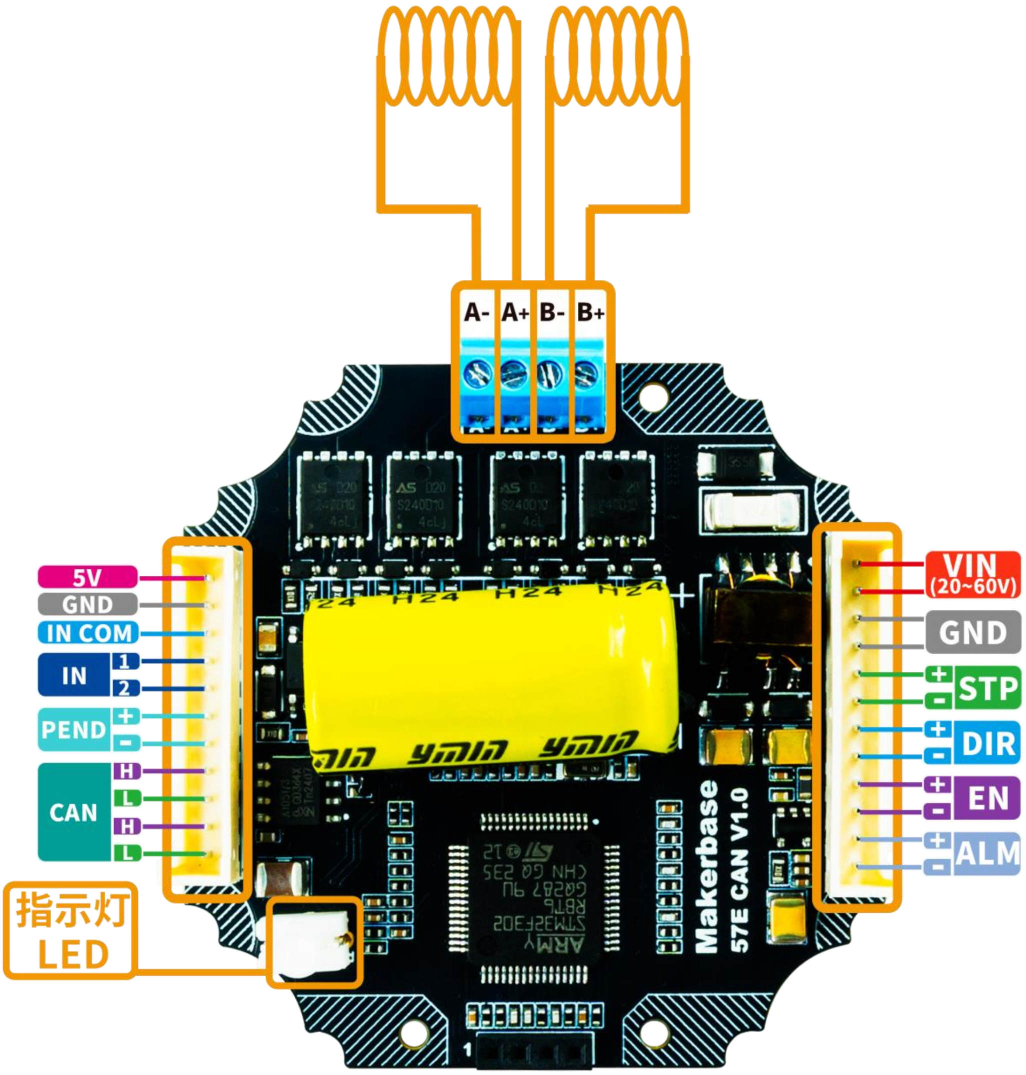
1.4 Interface Description

① Description of the SERVO42D_RS485 interface



1. Input power 20V~60V
2. 5V is the output power (5V, 100mA)
3. ALM is an alarm signal (when an alarm occurs, the optocoupler is closed)
4. PEND is the arrival signal (after arrival, the optocoupler is closed)
5. IN is the input signal, IN is the return to zero switch or the left limit, and EN can be remapped to the right limit. For details about the remapping content, please see the setting limit instruction in Chapter 4 (when EN is remapped to the right limit, the pulse control mode fails).

② Description of the SERVO57D_RS485 interface



1. Input power 20V~60V
2. 5V is the output power (5V, 100mA)
3. ALM is an alarm signal (when an alarm occurs, the optocoupler is closed)
4. PEND is the arrival signal (after arrival, the optocoupler is closed)
5. COM is the common port of IN1 and IN2 (can be connected in common negative or positive)
6. IN1, IN2 are input signals, IN2 is the zero return switch or left limit, IN1 is the right limit

1.5 Port funktionsbeskrivelse

Port funktionsbeskrivelse Table			
PIN	**I/O**	**Function**	**Description**
STP+ (CW+)	Input	Pulse signal (or CW pulse) positive	(1)Optocoupler input, falling edge trigger. The pulse goes one step when the pulse goes from high to low. (2)Positive voltage: 3.3 - 28V. (3) Negative voltage: High level 3.3 - 28V, Low level 0 - 0.5V. (4) Maximum input frequency 400kHz, pulse duration greater than 2.5us. (5) Single pulse (Pulse + Dir) or double pulse (CW + CCW) is set by the command.
STP- (CW-)		Pulse signal (or CW pulse) negative	
DIR+ (CCW+)		Direction signal (or CCW pulse) positive	
DIR- (CCW-)		Direction signal (or CCW pulse) negative	
EN+		Enable signal positive	(1) Optocoupler input, low level effective. When ineffective, the motor will be released while clearingthe alarm signal. (2) Positive voltage: 3.3 - 28V. (3) Negative voltage: High level 3.3 - 28V, Low level 0 - 0.5V.
EN-		Enable signal negative	
ALM+	Output	Alarm signal positive	(1) When an alarm occurs, the optocoupler output is on. (2) Maximum voltage +35V, maximum current 50mA.
ALM-		Alarm signal negative	
PEND+		In Position signal positive	(1) When the drive has finished a given pulse, the optocoupler output is on. (2) Maximum voltage +35V, maximum current 50mA.
PEND-		In Position signal negative	
A-	Output	Motor phase A-	Connected to motor phase line Note: If the phase sequence is not connected accordingly, the motor will alarm.
A+		Motor phase A+	
B-		Motor phase B-	
B+		Motor phase B+	
VDC	Input	Drive power supply positive	Operating Voltage +20 - 60V
GND		Drive power supply negative	

1.6 Status LED

Status LED Table	
Green stays on	**Motor Running**
Green flash	**Motor Stop**
1 Green + 1 Red	**Over Current**
1 Green + 2 Red	**Open-Phase**
1 Green + 3 Red	**Supply Voltage High**
1 Green + 4 Red	**Supply Voltage Low**
1 Green + 5 Red	**Position Error**
1 Green + 6 Red	**Encoder Error**

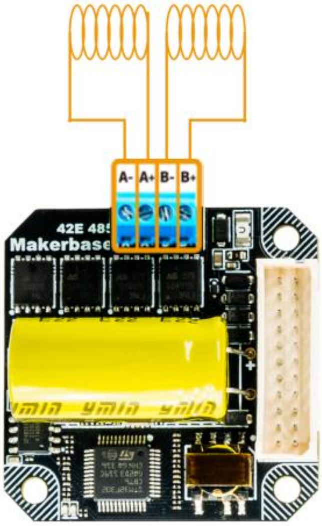
Tip: Please make sure the motor phase line is connected correctly, otherwise there will be a tracking error alarm after receiving the pulse (5 red and 1 green)

Part2. Motor wire

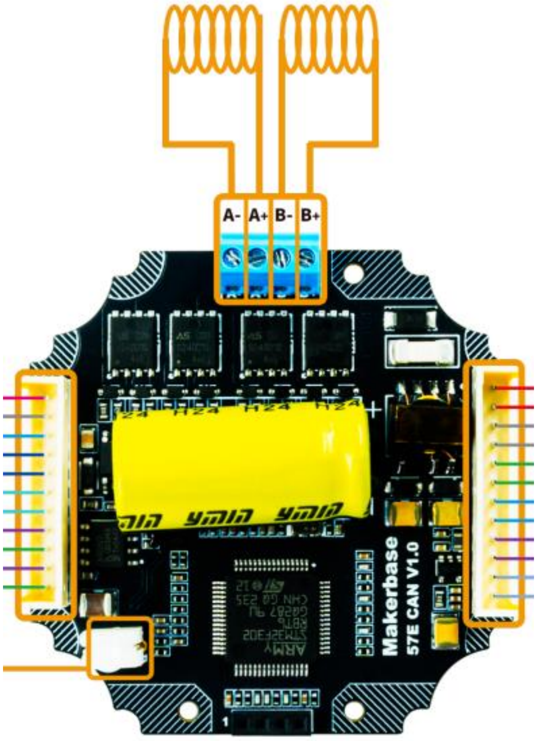
Note1: The motor internal resistance should be less than 10 ohms.

Note2:A+ A- is connected to one phase of the motor, B+ B- is connected to the other phase of the motor.

① SERVO42E_CAN motor wiring method



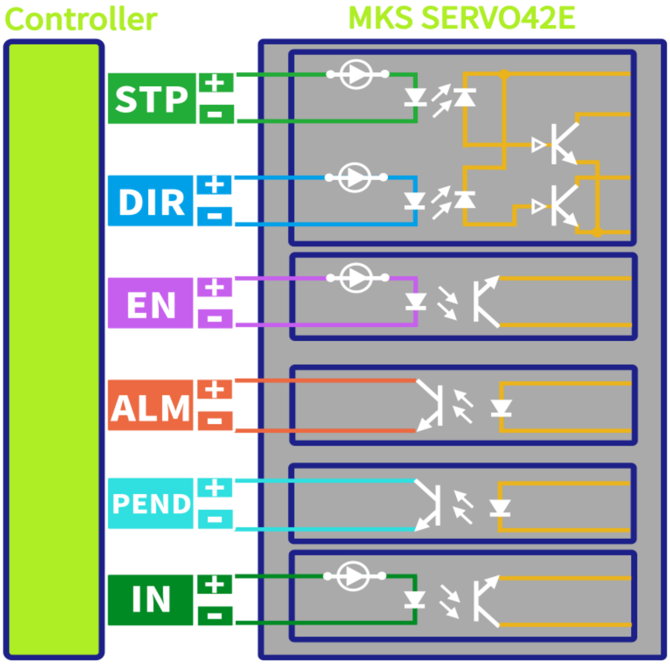
② SERVO57E_CAN motor wiring method



Tip: Please make sure the motor phase line is connected correctly, otherwise there will be a tracking error alarm after receiving the pulse (5 red and 1 green).

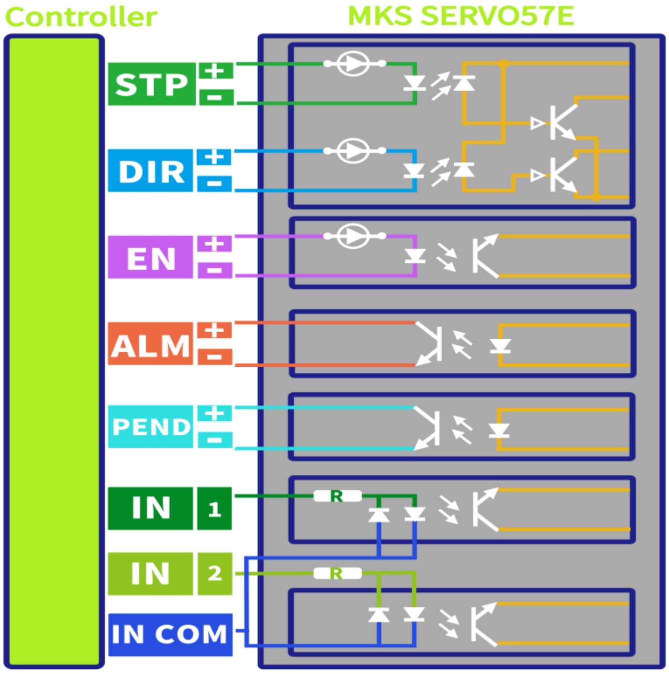
2.1 Pulse interface wire

①Connection method of SERVO42E_CAN



Note: All input ports have built-in 10mA current limiting, and can directly input 3.3V-24V signals without the need for external current limiting resistors

②Connection method of SERVO57E_CAN



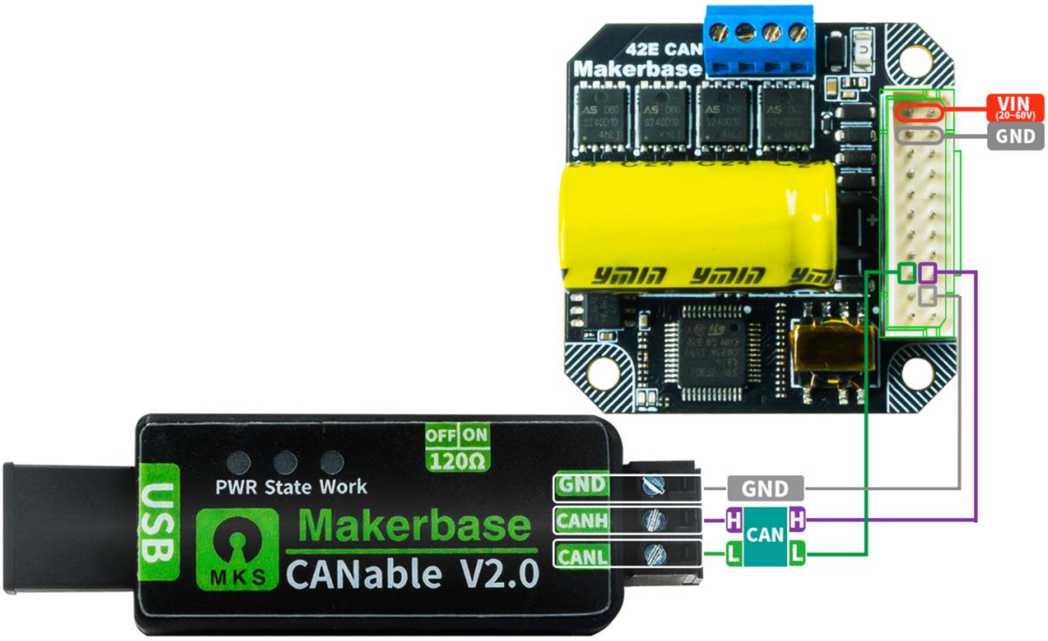
Note: All input ports have built-in 10mA current limiting, and can directly input 3.3V-24V signals without the need for external current limiting resistors

2.2 CAN wire

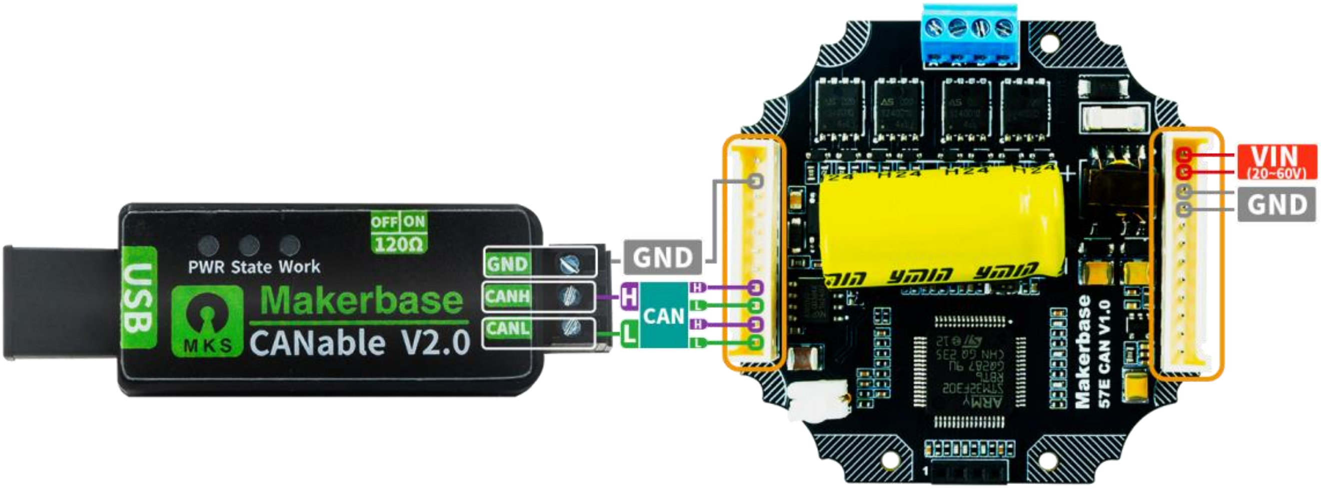
1. SERVO57E_CAN Single-slave

NOTE: To reduce bus interference, the host computer and the motor should share the same ground, and the CAN signal should be transmitted using shielded twisted pair cables.

① Connect cables to the MKS SERVO42E_CAN single machine

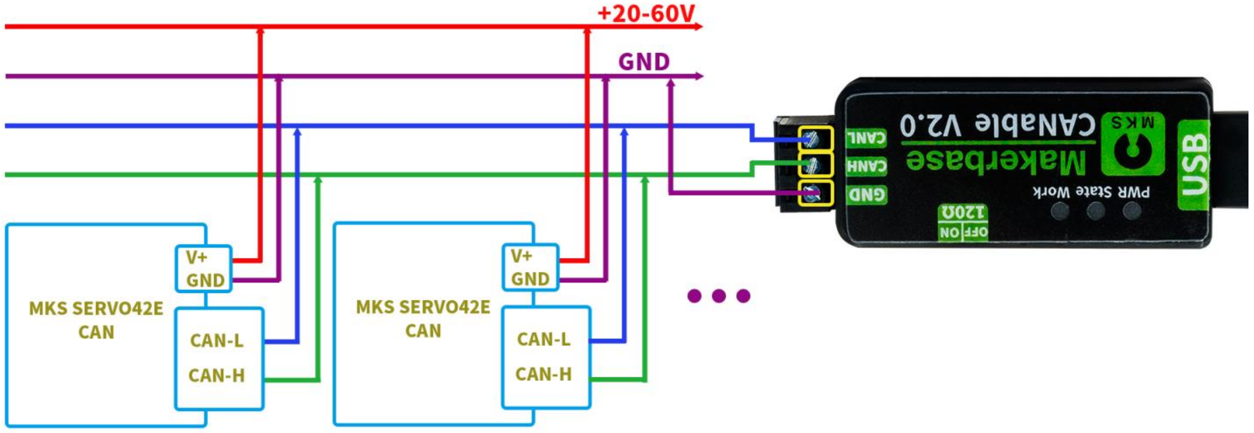


② Connect cables to the MKS SERVO57E_CAN single machine

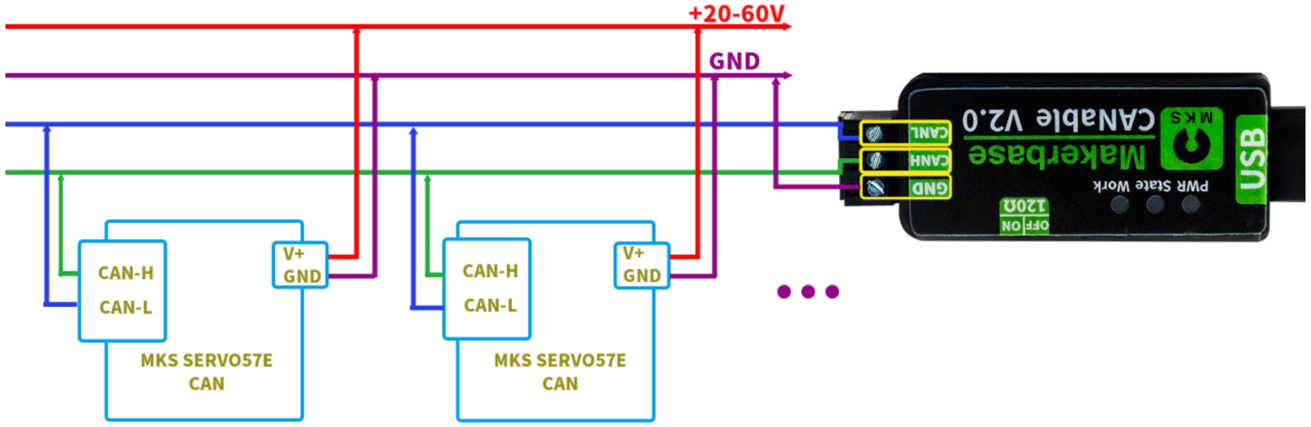


2. Multiple-slave

① Connection method of SERVO42E_CAN



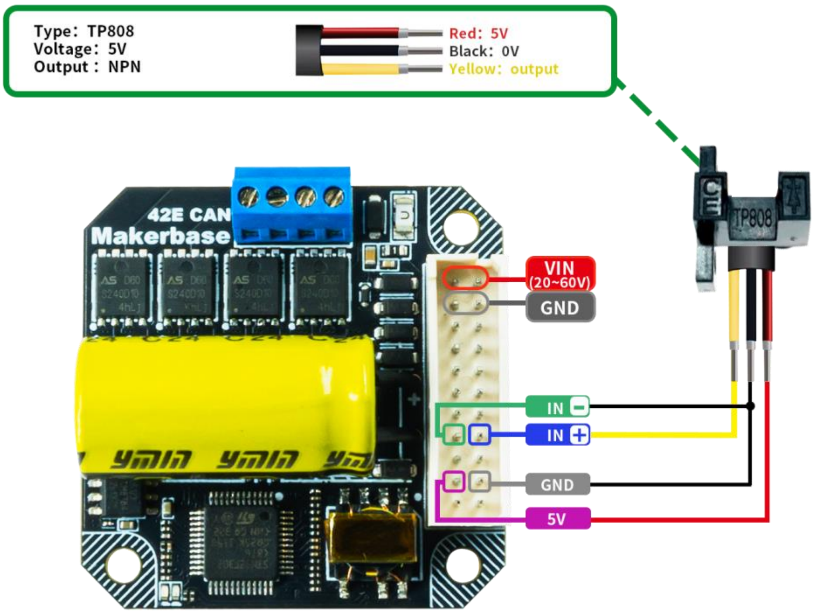
② Connection method of SERVO57E_CAN



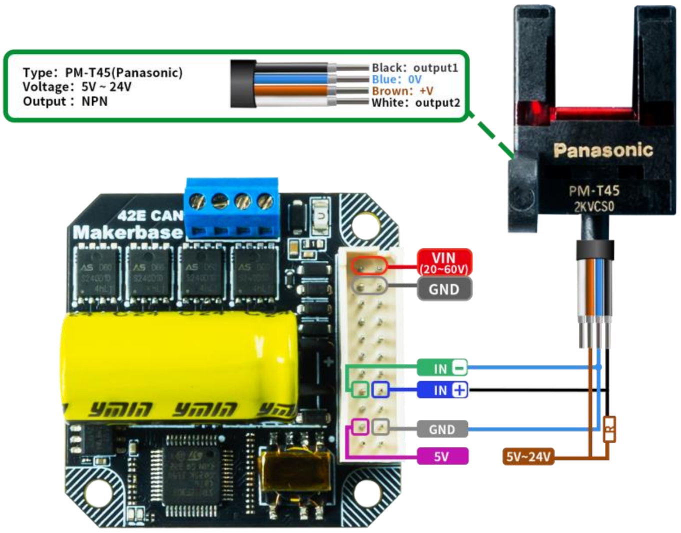
2.3 End stop wire

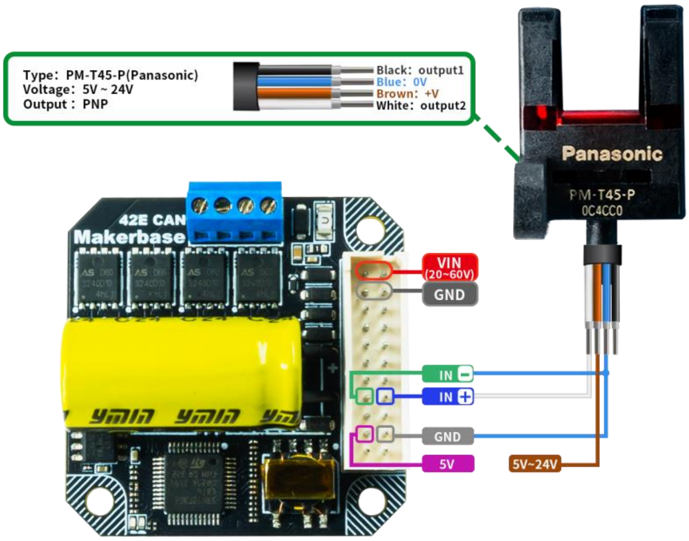
① Connection method of SERVO42E_CAN

(1) TP808 wiring



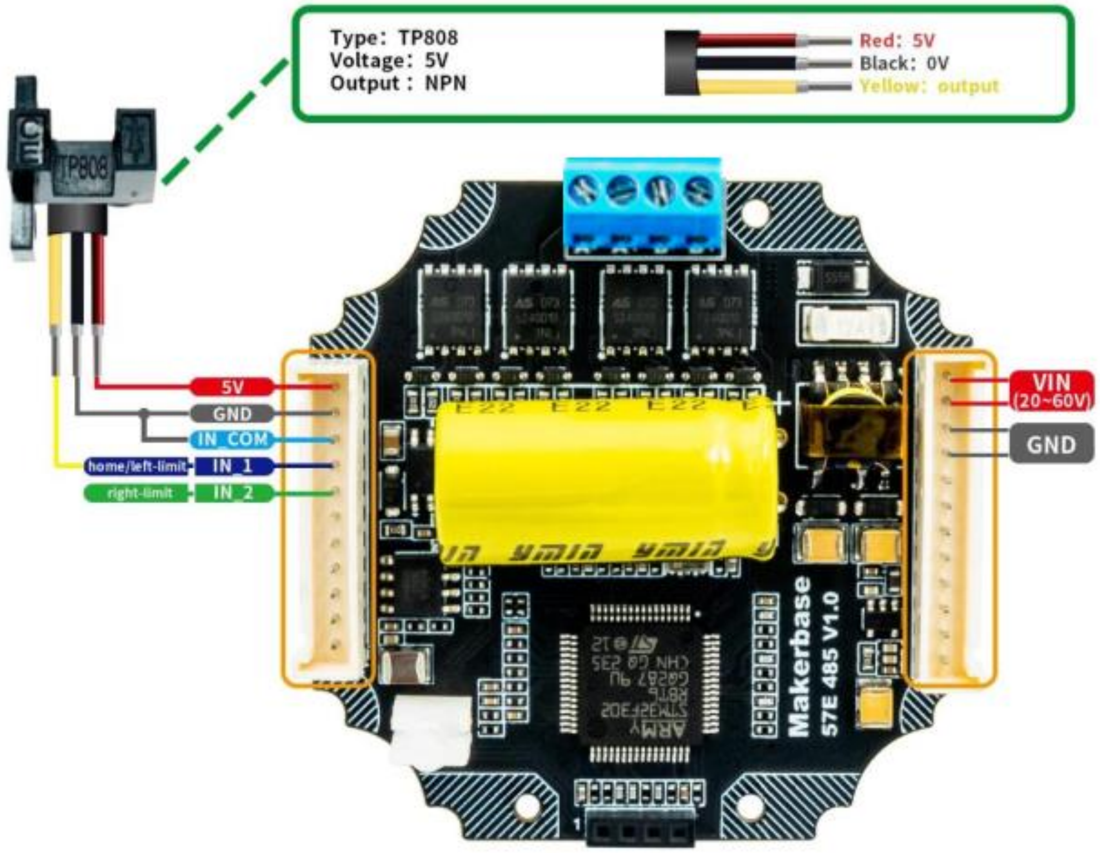
(2) PM-T45 and PM-T45-P wiring



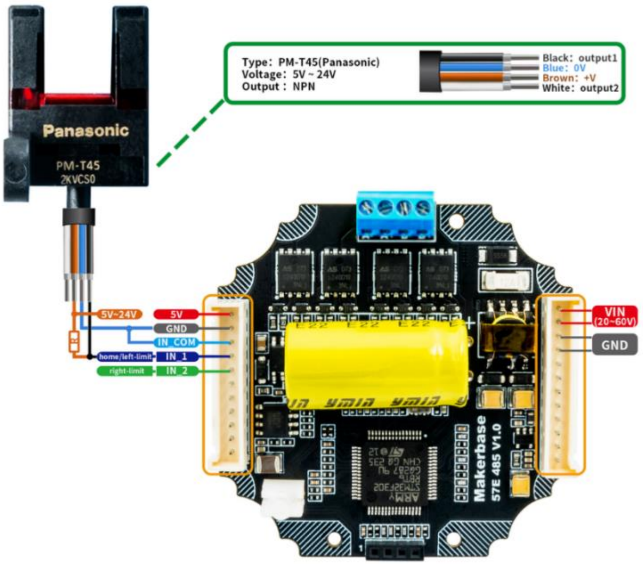


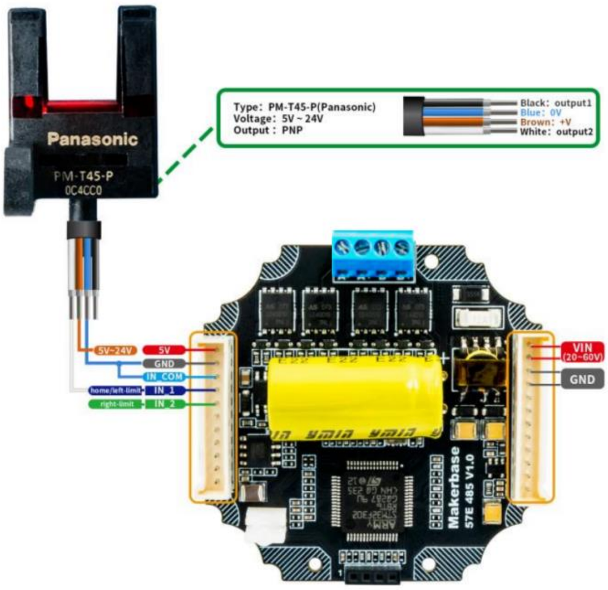
②Connection method of SERVO57E_CAN

(1) TP808 wiring



(2) PM-T45 and PM-T45-P wiring





Part3. CAN command description

Note: Please set the CAN ID first.(default:01)

The default CAN ID for the following chapters is 01.

3.1 Read status parameter command

1. command1 : 01 30 CRC

read the encoder value(carry).

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	30	CRC(31)

Uplink frame (PC ← SERVO42E/57E)							
CAN ID	…	DLC	byte1	byte2…byte5	byte6	byte7	byte8
01		8	code	carry	value		Check
			30	carry(int32_t)	value(uint16_t)		CRC

carry: the carry vaule of the encoder.

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

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

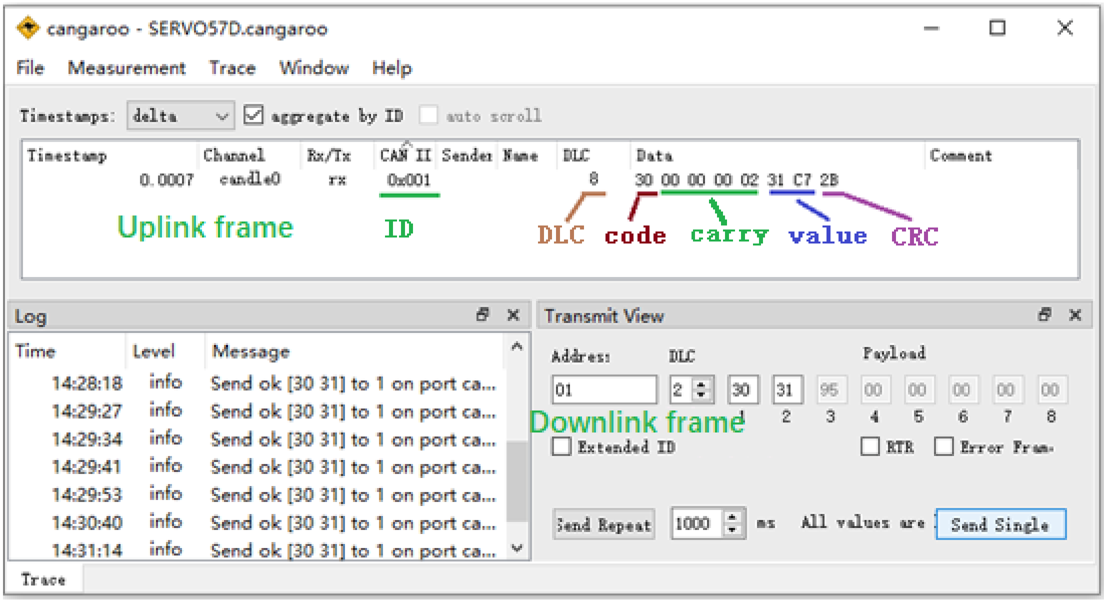
When Value is less than 0, carry -=1.

For example:

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

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

The Cangaroo example is as follows:



cangaroo - SERVO57D.cangaroo

Timestamps: delta ☑ aggregate by ID □ auto scroll

```
Timestamp  Channel  Rx/Tx  CAN ID  Sender  Name  DLC  Data                      Comment
   0.0007  candle0     rx   0x001                 8   30 00 00 00 02 31 C7 2B
```

Log

```
Time      Level  Message
14:28:18  info   Send ok [30 31] to 1 on port ca...
14:29:27  info   Send ok [30 31] to 1 on port ca...
14:29:34  info   Send ok [30 31] to 1 on port ca...
14:29:41  info   Send ok [30 31] to 1 on port ca...
14:29:53  info   Send ok [30 31] to 1 on port ca...
14:30:40  info   Send ok [30 31] to 1 on port ca...
14:31:14  info   Send ok [30 31] to 1 on port ca...
```

Transmit View

Address: 01 DLC: 2 Payload: 30 31 95 00 00 00 00 00

□ Extended ID □ RTR □ Error Frame

Send Repeat 1000 ms All values are Send Single

2. command2 : 01 31 CRC

read the encoder value(addition).

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	31	CRC(32)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2…byte7	byte8
01		8	code	value	Check
			31	value(int48_t)	CRC

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

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

For example:

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

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

3. Command3 : 01 32 CRC

Read the real-time speed of the motor.(RPM)

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	32	CRC(33)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2…byte3	byte4
01		4	code	data	Check
			32	speed(int16_t)	CRC

Note : if it run CCW, the speed > 0 (RPM)
if it run CW, the speed < 0 (RPM)

4. Command4 : 01 33 CRC

Read the number of pulses received.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	33	CRC(34)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2…byte5	byte6
01		6	code	data	Check
			33	pulses(int32_t)	CRC

5. Command5 : 01 34 CRC

read the IO Ports status.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	34	CRC(3B)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			34	status(uint8_t)	CRC

status							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
reserved				ALM	PEND	IN_2	IN_1

PEND	1: Already in place	0: Not in place
ALM	1: No alarm	0: Alarmed

Note: 42E does not have IN_2 port, bit1 corresponds to En port status.

6. Command7 : 01 39 CRC

read the error of the motor shaft angle.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	39	CRC(3A)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2…byte5	Byte6
01		6	code	data	Check
			39	error(int32_t)	CRC

The error is the difference between the angle you want to control minus the real-time angle of the motor, 0~51200 corresponds to 0~360°.

for example, when the angle error is 1°, the return error is 51200/360= 142.222, and so on.

7. Command8 : 01 3A CRC

read the En pins status.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	3A	CRC(3B)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			3A	enable(uint8_t)	CRC

enable =1 Enabled

enable =0 Disabled

8. Command10 : 01 3D CRC

Release the motor shaft locked-rotor protection state.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	3D	CRC(3E)

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			3D	status(uint8_t)	CRC

status =1 release success.

status =0 release fail.

9. Command11 : FA 01 3E CRC

Read the motor shaft protection state.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	3E	CRC(3F)

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			3E	status(uint8_t)	CRC

status =1 protected.

status =0 no protected.

3.2 Set system parameters command

1. Calibrate the encoder

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	80	00	CRC(81)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			80	status(uint8_t)	CRC

status =0 Calibrating….

status =1 Calibrated success.

status =2 Calibrating fail.

Note: The calibration only determines the relationship between the motor direction and the encoder, that is, when the motor rotates clockwise, the encoder value increases or decreases. If the motor phase line is wired according to the factory default connection, no calibration is required.

2. Set the work mode

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			82	mode	CRC

mode = X0 Pulse+Pulse Open-loop mode (X=0 with encoder X=1 without encoder)

mode = X1 Pulse+Direction Open-loop mode (X=0 with encoder X=1 without encoder)

mode = 02 Pulse+Pulse Closed-loop mode

mode = 03 Pulse+Direction Closed-loop mode (default)

mode = X4 RS485 bus Open-loop mode (X=0 with encoder X=1 without encoder)

mode = 05 RS485 bus Closed-loop mode

Note 1: Pulse control mode, maximum input frequency 300KHz Bus control mode, maximum speed 3000RPM

Note 2: X=0 has encoder, that is, the motor shaft has a magnet, the driver board is installed at the back, and the encoder value can be read X=1 has no encoder, that is, the motor shaft has no magnet, the driver board can be installed arbitrarily, and the encoder value cannot be read

Uplink frame（PC ← SERVO42D/57D）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			82	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

3. Set the working current

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2-3	Byte4
01		4	code	data	Check
			83	ma (uint16_t)	CRC

SERVO42E: Maximum Current =3000mA (default 1600mA)
SERVO57E: Maximum Current =5200mA (default 3200mA)

Uplink frame（PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			83	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

4. Set subdivision(Default 16 subdivisions)

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			84	micstep(00~FF)	CRC

Note1:The value range of micstep (decimal) is as follows:
0， 2， 4， 8， 16， 32， 64， 128，
5， 10， 20， 25， 40， 50， 100， 200
Note2:0 corresponds to 256 subdivisions

Uplink frame（PC ← SERVO42D/57D）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			84	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

5. Set the active of the En pin

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			85	enable(00~02)	CRC

enable = 00 active low (L) (default)
enable = 01 active high (H)
enable = 02 active always (Hold)

Uplink frame（PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			85	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

Note1:After successful setting, it will take 100ms to receive the pulse signal.

Note2:Only valid for pulse control mode.

6. Set the direction of motor rotation

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			86	dir(00~01)	CRC

dir = 00 CW (default)
dir = 01 CCW

Uplink frame（PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			86	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

Note: This instruction can also change the bus mode to control the running direction of the motor.

7. Set pulse delay

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			87	delay(uint8_t)	CRC

delay = 00 0ms delay = 01 4ms

delay = 02 20ms(default value) delay = 03 40ms

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			87	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

8. Set the motor shaft locked-rotor protection function

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			88	enable(00~01)	CRC

enable = 01 enabled protection (default)

enable = 00 disabled protection

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			88	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

Note: After the stall protection, the stall protection state can be released through the enable signal , serial port command,Command (3D) mode or EN level invalid mode.

9. Set the Stalling tolerant value

Downlink frame(PC → SERVO42E/57E)					
CAN ID		DLC	byte1	byte2-3	Byte4
01	…	3	code	data	Check
			89	value（0-0x7FFF)	CRC

The default value is 0x64.

When the error exceeds the value, the stall protection is triggered and the motor loosens its shaft.

value = 0x64 corresponds to an angle of 180 degrees

value = 0xC8 corresponds to an angle of 360 degrees

and so on...

Uplink frame（PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			89	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

10. Set the CAN bitRate

Downlink frame(PC → SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			8A	bitRate (00~03)	CRC

bitRate = 00 125K

bitRate = 01 250K

bitRate = 02 500K (default)

bitRate = 03 1M

Uplink frame（PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			8A	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

11. Set the CAN ID

Downlink frame(PC → SERVO42E/57E)						
CAN ID	…	DLC	byte1	byte2	byte3	byte4
01		4	code	data		Check
			8B	ID(00~7FF)		CRC

Note1: the default ID is 01

Note2: 00 is the broadcast address

Uplink frame（PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8B	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

12. Set the slave respond and active

Downlink frame(PC → SERVO42E/57E)						
CAN ID	…	DLC	byte1	byte2	byte3	byte4
01		4	code	data	data	Check
			8C	respon(00~01)	active(00-01)	CRC

respon = 01 enabled respond (default)
respon = 00 disabled respond
active = 01 enabled active (default)
active = 00 disabled active

Note: If disable respond, It can query the running status of the motor by command “F1”.

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8C	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

The difference between respond and active
Take position control mode 1 as an example:
Host sends 01 FD 02 80 02 00 FA 00 7C

a. In no response mode (respon =0, active = xx)
The slave does not return any information.

b. In the mode of not actively initiating data (respon =1, active =0)
Slave returns immediately Position control starts 01 or fails 00.

c. In default mode (respon =1, active =1)
Slave returns immediately Position control starts 01 or fails 00.
Return to 02 or 03 after the motor finishes running or touches the limit stop.

13. Set whether to lock the axis when starting bus mode

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8F	enable(00~01)	CRC

enable = 01 lock the axis(default value)
enable = 00 unlock axis

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8F	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

14. Set the group ID

Downlink frame(PC → SERVO42E/57E)						
CAN ID	…	DLC	byte1	byte2	byte3	byte4
01		4	code	data		Check
			8D	ID(01~0x7FF)		CRC

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8D	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

For example, there are 6 motors with the settings ID:

	Broadcast ID	Slave ID	Group ID
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 will rotate a turn
send 00 FD 01 2C 64 00 0C 80 1A, motor1-6 will rotate a turn
send 50 FD 01 2C 64 00 0C 80 6A, motor1-3 will rotate a turn
send 51 FD 01 2C 64 00 0C 80 6B, motor4-6 will rotate a turn

Note: Slave does not answer if group address is used.

3.3 Write IO port command

<table>
<tr><th colspan="13">Downlink frame(PC → SERVO42E/57E)</th></tr>
<tr><td>CAN ID</td><td rowspan="3">…</td><td>DLC</td><td>byte1</td><td colspan="8">Byte2</td><td>Byte3</td></tr>
<tr><td rowspan="2">01</td><td rowspan="2">3</td><td>code</td><td>bit7</td><td>bit6</td><td>bit5</td><td>bit4</td><td>bit3</td><td>bit2</td><td>bit1</td><td>bit0</td><td>Check</td></tr>
<tr><td>36</td><td colspan="2">alm_mask</td><td colspan="2">pend_mask</td><td>ALM</td><td>PEND</td><td colspan="2">0</td><td>CRC</td></tr>
</table>

alm_mask 0: Do not write to ALM IO port (ALM default alarm signal)

1: Write the ALM value to the ALM IO port

2: ALM IO port value remains unchanged

pend_mask 0: Do not write to the PEND IO port (PEND defaults to the in-place signal)

1: Write the PEND value to the PEND IO port

2: The PEND IO port value remains unchanged

ALM ALM port write value (0/1)

PEND PEND port write value (0/1)

<table>
<tr><th colspan="6">Uplink frame (PC ← SERVO42E/57E)</th></tr>
<tr><td>CAN ID</td><td rowspan="3">…</td><td>DLC</td><td>byte1</td><td>byte2</td><td>byte3</td></tr>
<tr><td rowspan="2">01</td><td rowspan="2">3</td><td>code</td><td>data</td><td>Check</td></tr>
<tr><td>36</td><td>status(uint8_t)</td><td>CRC</td></tr>
</table>

status =1 write success.

status =0 write fail.

3.4 Set Home command

1. Set the parameter of home

Downlink frame(PC → SERVO42E/57E)									
CAN ID	…	DLC	byte1	byte2	byte3	byte4-5	Byte6	Byte7	Byte8
01		8	code	level	dir	speed	enable	hmMode	Check
			90	homeTrig	homeDir	homeSpeed	EndLimit	mode	CRC

HmTrig the effective level of the end stop

0：Low (default value) 1：High

HmDir the direction of go home

0： CW(default value) 1：CCW

HmSpeed the speed of go home

0~3000 (RPM) default value = 60

EndLimit

0: disable endstop-limit(default value)

1: enable endstop-limit

Note : The speed description can be found in Chapter 6.1.

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			90	status(uint8_t)	CRC

status =1 set success.

status =0 set fail.

Note1:When EndLimit = 1， in bus control mode， the left limit is triggered and the motor no longer runs to the left; the right limit is triggered and the motor no longer runs to the right;

Note2: When using the limit function for the first time or changing the limit parameters， it is necessary to execute a limit reset ("91" command).

Note3:The limit function is invalid in pulse control mode.

2. Go home

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	91	CRC(92)

Note: When returning to zero with limit, if the limit switch is already in the closed state, the motor will rotate a certain distance in the opposite direction of homeDir (set by command 94) and then return to zero.

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			91	status(uint8_t)	CRC

status =0 go home fail.
status =1 go home start.
status =2 go home sucess.

3. Set Currnet Axis to zero

It can set the current Axis to Zero. Just as “GoHome” without run the motor.

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	92	CRC(93)

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			92	status(uint8_t)	CRC

status =0 set fail.
status =1 set success.

4. Set the unlimited switch to return to zero current

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2-3	Byte4
01		3	code	data	Check
			93	status(uint8_t)	CRC

SERVO42E maximum return to zero current 3000mA (default 300mA)
SERVO57E maximum return to zero current 5200mA (default 600mA)

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			93	status(uint8_t)	CRC

status =0 set fail.
status =1 set success.

Note1:When the infinite switch returns to zero, the motor runs at a fixed torque (nullHmMa) until it hits an obstacle and stops, then runs in the reverse direction for a distance (retValue) and stops. The stop point is the zero point.

Note2:The unlimited return to zero current value is only valid during unlimited return to zero operation. It should be set to a smaller current as much as possible to avoid damaging the motor.

5. Set the parameter of “noLimit” go home

Downlink frame(PC → SERVO42D/57D)						
CAN ID	…	DLC	byte1	Byte2-5	Byte6-7	Byte8
01		6	code	Reverse Angle	Hm_ma	Check
			94	retValue	ma	CRC

mode 0: used Limit switch for go home(default value).
1: no Limit switch for go home.

trig 0: Disable the zero return trigger function (default value, return to zero through command 91.
1: Automatically return to zero after power on.
2: En signal triggers zero return (valid only in pulse control mode).

retValue: 0~0xFFFFFFFF (Default = 0x2000, returns half a turn, 180 degrees)

for example：

retValue = 0x4000 （it will return 360 degree）

retValue = 0x2000 （it will return 180 degree） （default）

Uplink frame （PC ← SERVO42D/57D）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			94	status(uint8_t)	CRC

status =0 set fail.
status =1 set success.

Note1: In pulse control mode, when hmTrig = 2, when the En signal line generates a 200ms width non-enable level, the motor is triggered to return to zero. (Pulse recognition range 150ms~250ms).

Note2: When En is low level to enable the motor, a 200ms high level signal is generated to trigger the motor to return to zero.
When En is high level to enable the motor, a 200ms low levelsignal is generated to trigger the motor to return to zero.

6. Set limit port remap

Map the En port to the right limit port, which is only suitable for bus control mode.

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			9E	reMap (uint8_t)	CRC

reMap = 01 enable remap limit port

reMap = 00 disable remap limit port (default)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			9E	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

Note: This instruction is invalid for 57E and pulse control mode.

3.5 Setting the pulse division output command

Map the PEND port as the pulse frequency division output port.

Downlink frame(PC → SERVO42D/57D)						
CAN ID	…	DLC	byte1	Byte2	Byte3-6	Byte7
01		7	code	Start level	division period	Check
			9F	divLevel	divPeriod	CRC

divLevel 0: Starting level low; 1: Starting level high (default 0)

divPeriod division period (default 0)

When divPeriod < 100, there is no frequency division output

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

For example, if 16 subdivisions are set and divPeriod = 3200, the PEND port flips once for every motor rotation.

Note: To cancel this function, set divPeriod = 0.

Uplink frame (PC ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			9F	status(uint8_t)	CRC

status =0 set fail.

status =1 set success.

3.6 Restore default parameters and reset and restart instructions

3.5.1 Restore default parameters

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	3F	CRC(40)

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			3F	status(uint8_t)	CRC

status =1 restore success.
status =0 restore fail.

Note: After restored the parameters, It will reboot again.

3.5.2 Restart the motor

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	41	CRC

Uplink frame（PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			41	status(uint8_t)	CRC

status =1 restart success.
status =0 restart fail.

Note:This command only resets the motor and does not modify the configuration parameters.

3.7 Read version information

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	40	CRC

Downlink frame(PC → SERVO42E/57E)								
CAN ID		DLC	byte1	Byte2			Byte3-5	Byte6
01	…	6	code	bit7	bit5-bit4	bit3-bit0		Check
			40	serials	cal	hardVer	firmVer[3]	CRC

series = 1 E series stepper motor series = 0 D series stepper motor

cal = 1 When the motor rotates clockwise, the encoder value increases

cal = 2 When the motor rotates clockwise, the encoder value decreases

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

Corresponding version V1.0.0

The hardware versions correspond to the following

Type	hardVer
S42E_RS485	1
S42E_CAN	2
S57E_RS485	3
S57E_CAN	4
S28E_RS485	5
S28E_CAN	6
S35E_RS485	7
S35E_CAN	8

3.8 Read/Write User ID

1. Write User ID

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2-5	Byte6
01		6	code	User ID	Check
			42	ID (uint32_t)	CRC

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			42	status(uint8_t)	CRC

status =1 Write success.

status =0 Write fail.

2. Read User ID

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	42	CRC

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2-5	Byte6
01		6	code	USER ID	Check
			42	ID(uint32_t)	CRC

3.9 Read system Parameter command

The command format for reading system parameters is as follows:

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte 3
01		3	00	code	CRC

code : corresponding to the system parameters.

For example, if you want to read the "work mode", the corresponding code is 82H.

The format of the returned parameter data is as follows:

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte 1	byte 2-n	byte n+1
01		n+1	CODE	parameters	Check
			code	param	CRC

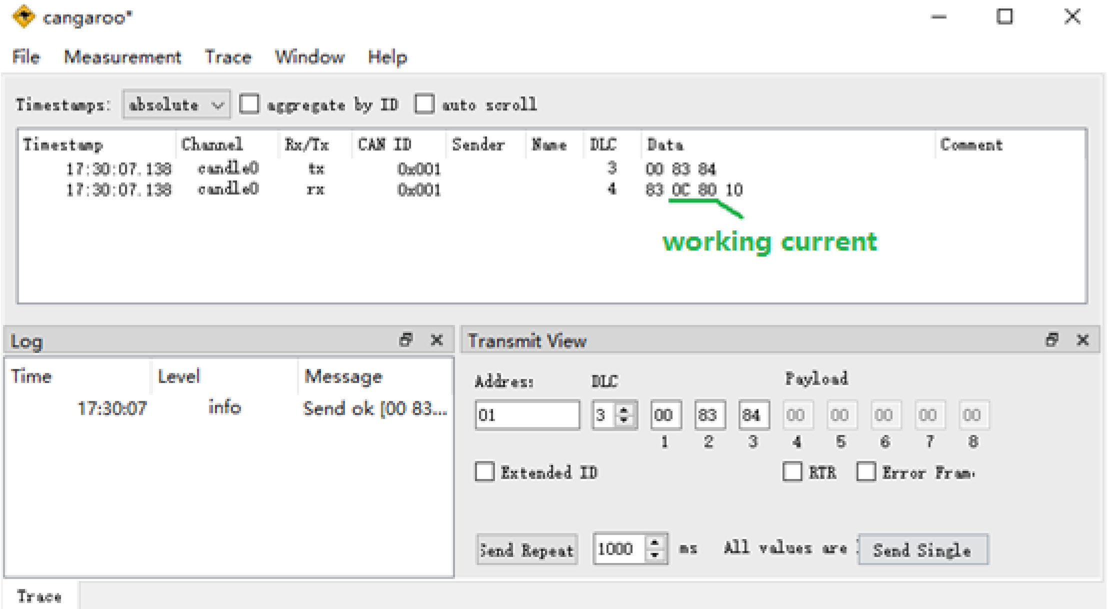
param : the system parameters.

Note: The returned param data format just the same as the data format when setting this parameter.

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

Uplink frame (PC ← SERVO42E/57E)						
CAN ID	…	DLC	byte 1	byte 2	byte 3	byte 4
01		4	CODE			Check
			code	FFH	FFH	CRC

The following figure shows an example of reading "working current":



Timestamps: absolute ☐ aggregate by ID ☐ auto scroll

```
Timestamp      Channel  Rx/Tx  CAN ID  Sender  Name  DLC  Data          Comment
 17:30:07.138  candle0   tx    0x001                  3   00 83 84
 17:30:07.138  candle0   rx    0x001                  4   83 0C 80 10
```

Time	Level	Message
17:30:07	info	Send ok [00 83...

Address: 01 DLC: 3 Payload: 00 83 84 00 00 00 00 00 (1 2 3 4 5 6 7 8)

☐ Extended ID ☐ RTR ☐ Error Fram

Send Repeat 1000 ms All values are

Part4. Run the motor by CAN command

Note: This chapter needs to set the working mode to serial mode. (SR_OPEN/SR_CLOSE/SR_VFOC)

4.1 Description the parameters of speed and acceleration

1. speed

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

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

For example, setting speed=1200
At 8 subdivisions, the speed is 2400 (RPM)
At 16/32/64 subdivisions, the speed is 1200 (RPM)
At 128 subdivisions, the speed is 150 (RPM)

2. acceleration

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

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

① accelerates

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

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

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

For example: acc = 236, speed = 3000

T(ms)	speed (RPM)
0	0
1	1
2	2
3	3
…	…

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

② decelerates

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

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

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

4.2 Query/Enable the motor command

1. Query the motor status

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	F1	CRC(F2)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F1	status(uint8_t)	CRC

status = 0 query fail.
status = 1 motor stop
status = 2 motor speed up
status = 3 motor speed down
status = 4 motor full speed
status = 5 motor is homing
status = 6 motor is Cal…

Note 1: This instruction is only valid in bus control mode.

Note 2: This instruction can only query the motor calibration operation in pulse control mode.

2. Enable the motor

Note:In bus control mode, the enable state of the driver board is no longer controlled by the level of the En pin, but is controlled by this command.

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F3	en(00~01)	CRC

en = 00 disable.
en = 01 enable.

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F3	status(uint8_t)	CRC

status =1 Set success.
status =0 Set fail.

4.3 Emergency stop the motor

Downlink frame(PC → SERVO42E/57E)				
CAN ID	…	DLC	byte1	byte2
01		2	F7	CRC

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F7	status(uint8_t)	CRC

status = 0 stop fail.

status = 1 stop success.

Note: If the motor rotating more than 1000RPM, it is not a good idea to stop the motor immediately!

4.4 Speed mode command

In speed mode, the motor can be run with a fixed acceleration and speed.

1. Run the motor in speed mode

Downlink frame(PC → SERVO42E/57E)									
CAN ID	…	DLC	byte1	byte 2			byte 3	byte 4	byte 5
01		5	code	dir	Rev	speed		acc	Check
			F6	b7	b6-b4	b3-b0	b7-b0	acc	CRC
				dir	——	speed			

byte2: The highest bit indicates the direction, the lower 4 bits and byte3 together indicate the speed

byte3: The lower 4 bits of byte2 and byte3 together indicate speed

The parameter description is as follows:

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

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

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

for example:

Send “01 F6 01 40 02 3A”,

the motor rotates forward at acc=2, speed=320RPM

Send “01 F6 81 40 02 BA”,

the motor reverses at acc=2, speed=320RPM

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F6	status(uint8_t)	CRC

status = 1 run success.

status = 0 run fail.

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

2. Stop the motor in speed mode

Downlink frame(PC → SERVO42E/57E)									
CAN ID		DLC	byte1	byte 2			byte 3	byte 4	byte 5
01	…	5	code	dir	Rev	speed		acc	Check
			F6	b7	b6-b4	b3-b0	b7-b0	acc	CRC
				0	0	0			

The stop command can stop the motor slowly, or stop the motor immediately.

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

When setting acc = 0, the motor stops immediately

① Deceleration and stop the motor slowly (acc ≠ 0)

for example:

Send 01 F6 00 00 02 F9

Stop the motor with deceleration acc=2

② Immediate stop command (acc = 0)

for example:

Send 01 F6 00 00 00 F7

Stop the motor immediately

Note: If the motor rotating more than 1000RPM, it is not a good idea to stop the motor immediately!

Uplink frame (PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			F6	status(uint8_t)	CRC

status = 0 stop the motor fail.

status = 1 start to stop the motor.

status = 2 stop the motor success.

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

3. Save/Clean the parameter in speed mode

Downlink frame(PC → SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			FF	state	CRC

state = C8 Save.

state = CA Clean.

Uplink frame （PC ← SERVO42E/57E）					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			FF	status(uint8_t)	CRC

status = 1 success.

status = 0 fail.

Note: The motor can rotates clockwise or counterclockwise at a constant speed when powered on.

4.5 Position mode1: relative motion by pulses

In the position control mode1, the motor can be run to the specified position with the set acceleration and speed.

1. Run the motor in position mode1

Downlink frame(PC → SERVO42E/57E)										
CAN ID		DLC	byte1	byte 2			byte 3	byte 4	byte 5-7	byte 8
01	…	8	code	dir	Rev	speed		acc	pulses	Check
			FD	b7	b6-b4	b3-b0	b7-b0	acc	pulses	CRC
				dir	——	speed				

byte2: The highest bit indicates the direction, the lower 4 bits and byte3 together indicate the speed

byte3: The lower 4 bits of byte2 and byte3 together indicate speed

The parameter description is as follows:

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

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

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

pulses: the motor run steps, the value range is 0 - 0xFFFFFF

for example:

Send 01 FD 01 40 02 00 FA 00 3B,

the motor rotates 20 times in the forward direction with acc=2, speed=320RPM (16 subdivisions);

Send 01 FD 81 40 02 00 FA 00 BB,

the motor rotates 20 times in the reverse direction with acc=2, speed=320RPM (16 subdivisions);

Uplink frame (PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			FD	status(uint8_t)	CRC

status = 0 run fail.

status = 1 run starting….

status = 2 run complete.

status = 3 end limit stoped.

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

2. Stop the motor in position mode1

Downlink frame(PC → SERVO42E/57E)										
CAN ID		DLC	byte1	byte 2			byte 3	byte 4	byte 5-7	byte 8
			code	dir	Rev	speed		acc	pulses	Check
01	…	8	FD	b7	b6-b4	b3-b0	b7-b0	acc	0	CRC
				0	0	0				

The stop command can stop the motor slowly, or stop the motor immediately.

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

When setting acc = 0, the motor stops immediately

① Deceleration and stop the motor slowly (acc ≠ 0)

for example:

Send 01 FD 00 00 04 00 00 00 02

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send 01 FD 00 00 00 00 00 00 FE

Stop the motor immediately

Note: If the motor rotating more than 1000RPM, it is not a good idea to stop the motor immediately!

Uplink frame (PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			FD	status(uint8_t)	CRC

status = 0 stop the motor fail.

status = 1 stop the motor starting….

status = 2 stop the motor complete.

status = 3 end limit stoped.

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

4.6 Position mode2: absolute motion by pulses

In the position control mode2, the motor can be run to the specified axis with the set acceleration and speed.

1. Run the motor in position mode2

Downlink frame(PC → SERVO42E/57E)								
CAN ID		DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01	…	8	code	speed		acc	absolute axis	Check
			FE	speed		acc	absPulses	CRC

The parameter description is as follows:
speed: the speed, the value range is 0-3000(RPM)
acc: the acceleration, the value range is 0-255
absPulses: the absolute pulses, int24_t (-8388607, +8388607)

For example:
If the current axis is any value
Send 01 FE 02 58 02 00 40 00 9B
The motor will move to 0x4000 (speed = 600(RPM),acc =2)
After move the pulses is 0x4000.

If the current axis is any value
Send 01 FE 02 58 02 FF C0 00 1A
The motor will move to -0x4000 (speed = 600(RPM),acc =2)
After move the pulses is -0x4000.

Uplink frame (PC ← SERVO42E/57E)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			FE	status(uint8_t)	CRC

status = 0 run fail.
status = 1 run starting….
status = 2 run complete.
status = 3 end limit stoped.

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

2. Stop the motor in position mode2

Downlink frame(PC → SERVO42E/57E)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01		8	code	speed		acc	absolute axis	Check
			FE	0		acc	0	CRC

The stop command can stop the motor slowly, or stop the motor immediately.

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

When setting acc = 0, the motor stops immediately

① Deceleration and stop the motor slowly (acc ≠ 0)

for example:

Send 01 FE 00 00 04 00 00 00 03

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send 01 FE 00 00 00 00 00 00 FF

Stop the motor immediately

Note: If the motor rotating more than 1000RPM, it is not a goog idea to stop the motor immediately!

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			FE	status(uint8_t)	CRC

status = 0 stop the motor fail.
status = 1 stop the motor starting….
status = 2 stop the motor complete.
status = 3 end limit stoped.

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

4.7 Position mode3: relative motion by axis

In the position control mode3, the motor can be run to the specified axis with the set acceleration and speed.

Note1: the axis is the encoder value(addition). It can be read by command "31".

1. Run the motor in position mode3

Downlink frame(PC → SERVO42E/57E)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01		8	code	speed		acc	Relative axis	Check
			F4	speed		acc	relAxis	CRC

The parameter description is as follows:
speed: the speed, the value range is 0-3000(RPM)
acc: the acceleration, the value range is 0-255
relAxis: the relative axis, int24_t (-8388607, +8388607)

For example:
If the current axis is 0x8000. (read by code "31")
Send 01 F4 02 58 02 00 40 00 91
The motor will relative move 0x4000 (speed = 600(RPM), acc =2)
After move the axis is 0xC000. (0x8000+0x4000=0xC000)

If the current axis is 0x8000. (read by code "31")
Send 01 F4 02 58 02 FF C0 00 09
The motor will relative move -0x4000 (speed = 600(RPM), acc =2)
After move the axis is 0x4000. (0x8000-0x4000=0x4000)

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F4	status(uint8_t)	CRC

status = 0 run fail.
status = 1 run starting….
status = 2 run complete.
status = 3 end limit stoped.

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

2. Stop the motor in position mode3

Downlink frame(PC → SERVO42E/57E)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01		8	code	speed		acc	Relative axis	Check
			F4	0		acc	0	CRC

The stop command can stop the motor slowly, or stop the motor immediately.

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

When setting acc = 0, the motor stops immediately

① Deceleration and stop the motor slowly (acc ≠ 0)

for example:

Send 01 F4 00 00 04 00 00 00 F9

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send 01 F4 00 00 00 00 00 00 F5

Stop the motor immediately

Note: If the motor rotating more than 1000RPM, it is not a good idea to stop the motor immediately!

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F4	status(uint8_t)	CRC

status = 0 stop the motor fail.

status = 1 stop the motor starting….

status = 2 stop the motor complete.

status = 3 end limit stoped.

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

4.8 Position mode4: absolute motion by axis

In the position control mode4, the motor can be run to the specified axis with the set acceleration and speed.

Note1: the axis is the encoder value(addition). It can be read by command “31”.

Note2: Support real-time updates of speed and coordinates, that is, new commands can be issued to change speed and coordinates when the previous command is running

1. Run the motor in position mode4

Downlink frame(PC → SERVO42E/57E)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01		8	code	speed		acc	absolute axis	Check
			F5	speed		acc	absAxis	CRC

The parameter description is as follows:
speed: the speed, the value range is 0-3000(RPM)
acc: the acceleration, the value range is 0-255
absAxis: the absolute axis, int24_t (-8388607, +8388607)

For example:
If the current axis is any value
Send 01 F5 02 58 02 00 40 00 92
The motor will move to 0x4000 (speed = 600(RPM),acc =2)
After move the axis is 0x4000.

If the current axis is any value
Send 01 F5 02 58 02 FF C0 00 11
The motor will move to -0x4000 (speed = 600(RPM),acc =2)
After move the axis is -0x4000.

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F5	status(uint8_t)	CRC

status = 0 run fail.
status = 1 run starting….
status = 2 run complete.
status = 3 end limit stoped.

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

2. Stop the motor in position mode4

Downlink frame(PC → SERVO42E/57E)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	byte5-byte7	字节 8
01		8	code	speed		acc	absolute axis	Check
			F5	0		acc	0	CRC

The stop command can stop the motor slowly, or stop the motor immediately.

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

When setting acc = 0, the motor stops immediately

① Deceleration and stop the motor slowly (acc ≠ 0)

for example:

Send 01 F5 00 00 04 00 00 00 FA

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send 01 F5 00 00 00 00 00 00 F6

Stop the motor immediately

Note: If the motor rotating more than 1000RPM, it is not a goog idea to stop the motor immediately!

Uplink frame (PC ← SERVO42E/57E)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F5	status(uint8_t)	CRC

status = 0 stop the motor fail.

status = 1 stop the motor starting….

status = 2 stop the motor complete.

status = 3 end limit stoped.

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

Part5. CAN command example

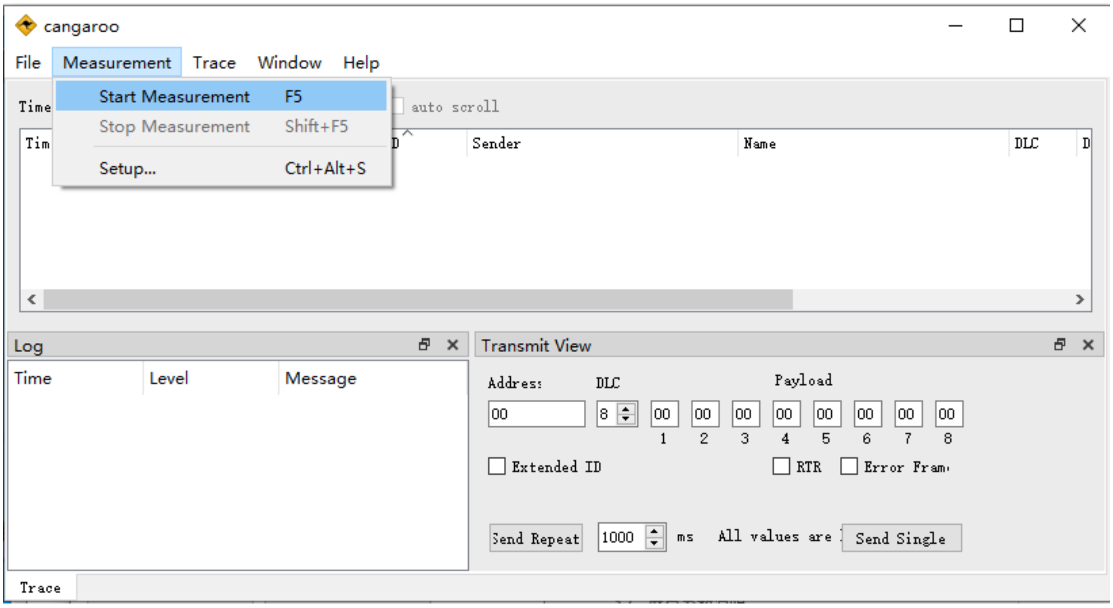
The following example uses "cangaroo.exe" PC software and "MKS CANable" USB to CAN module.

5.1 Config the SERVO42E/57E

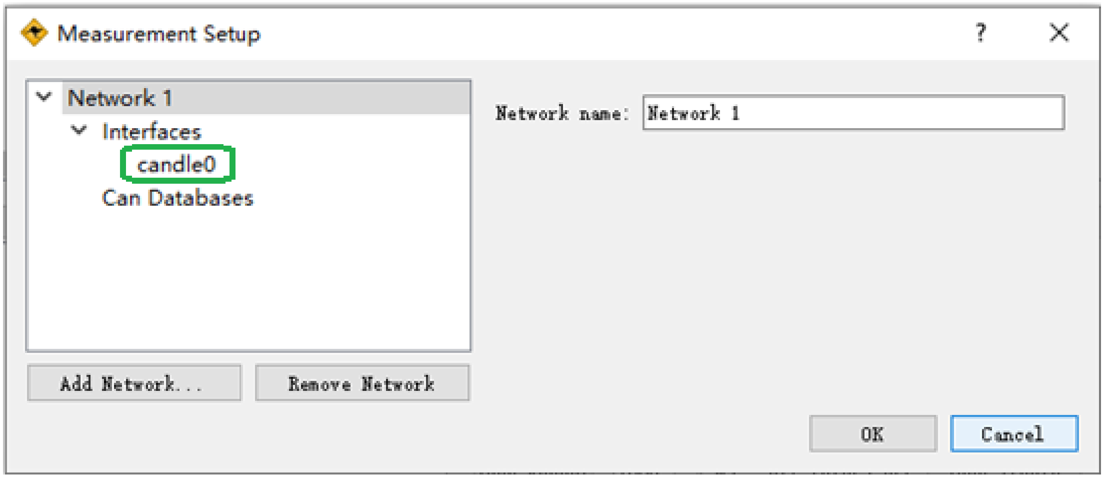
1. Mode → 05 RS485 bus closed loop FOC mode
2. CanRate → 500K.
3. CanID → 01.

5.2 Config the cangaroo

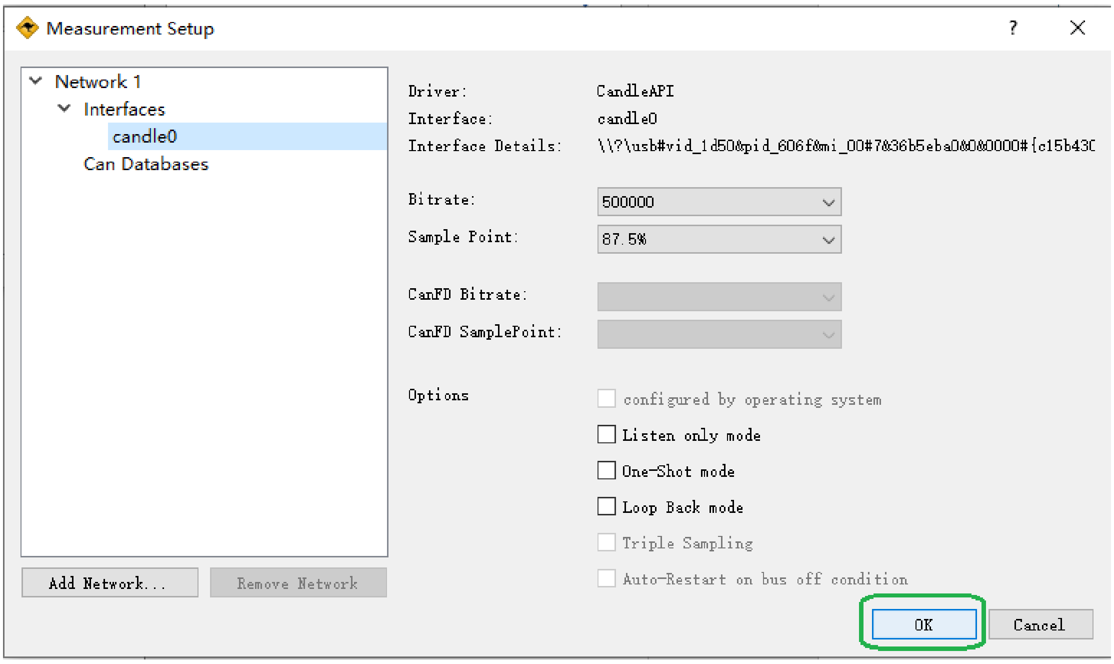
1. run the "cangaroo.exe".
2. Select t "Measurement" -> "Start Measurement", as show below.



3. In the pop-up "Measurement Setup" window, click "candle0", as shown below.



4. Use the default parameters without any modification, click "ok", as shown below.



Driver: CandleAPI

Interface: candle0

Interface Details: \\?\usb#vid_1d50&pid_606f&mi_00#7&36b5eba0&0&0000#{c15b43C

Bitrate: 500000

Sample Point: 87.5%

CanFD Bitrate:

CanFD SamplePoint:

Options

□ configured by operating system

□ Listen only mode

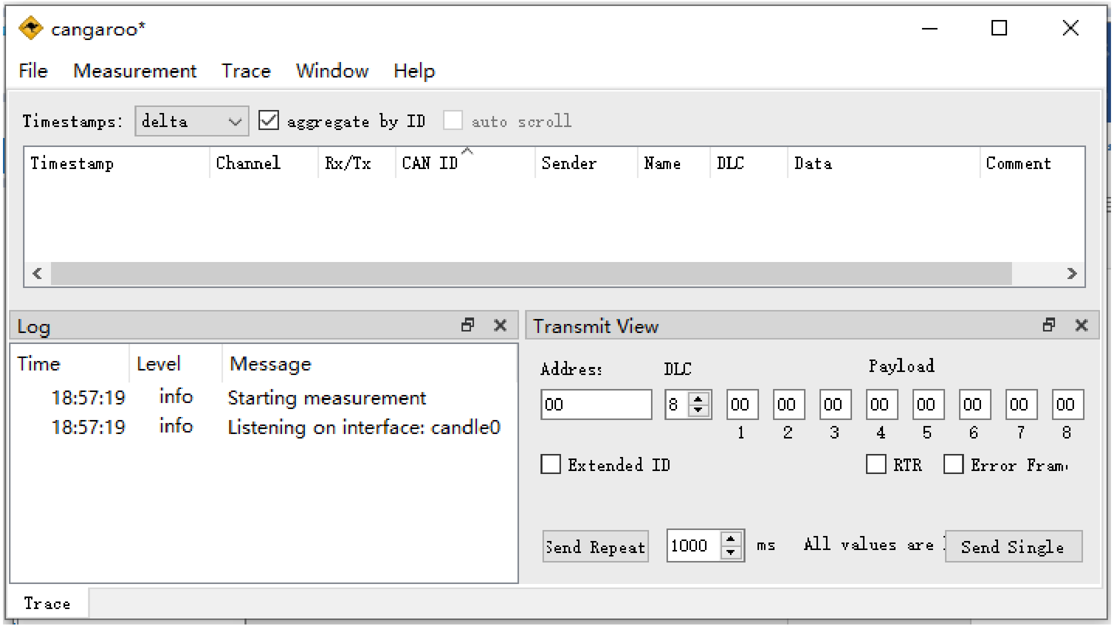
□ One-Shot mode

□ Loop Back mode

□ Triple Sampling

□ Auto-Restart on bus off condition

5. The configuration is complete, as shown below.

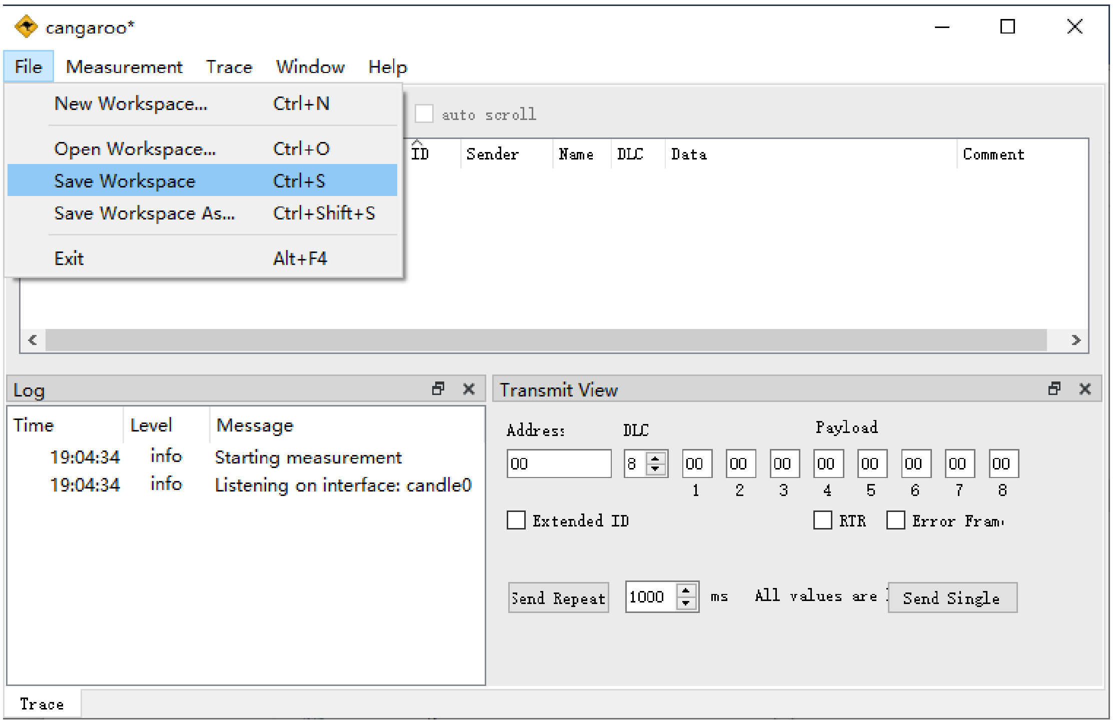


Time	Level	Message
18:57:19	info	Starting measurement
18:57:19	info	Listening on interface: candle0

Address: 00 DLC: 8 Payload: 00 00 00 00 00 00 00 00 (1 2 3 4 5 6 7 8)

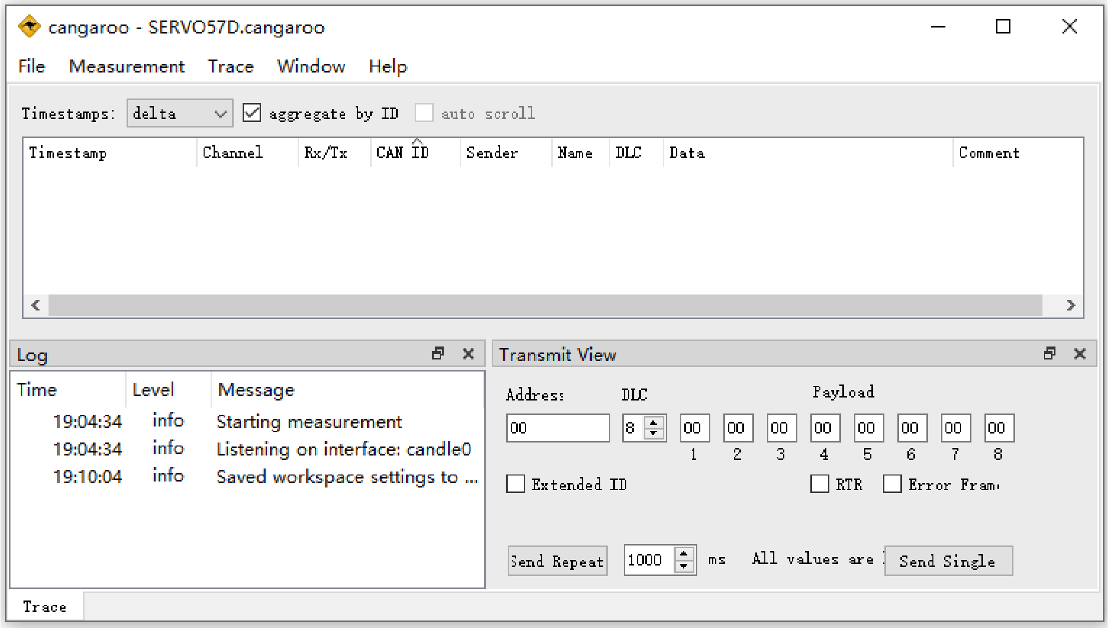
Send Repeat: 1000 ms

6. Select "File" -> "Save Workspace", select the save path and name, and save the configuration.



```
Time      Level  Message
19:04:34  info   Starting measurement
19:04:34  info   Listening on interface: candle0
```

7. After the save is completed, as shown below.

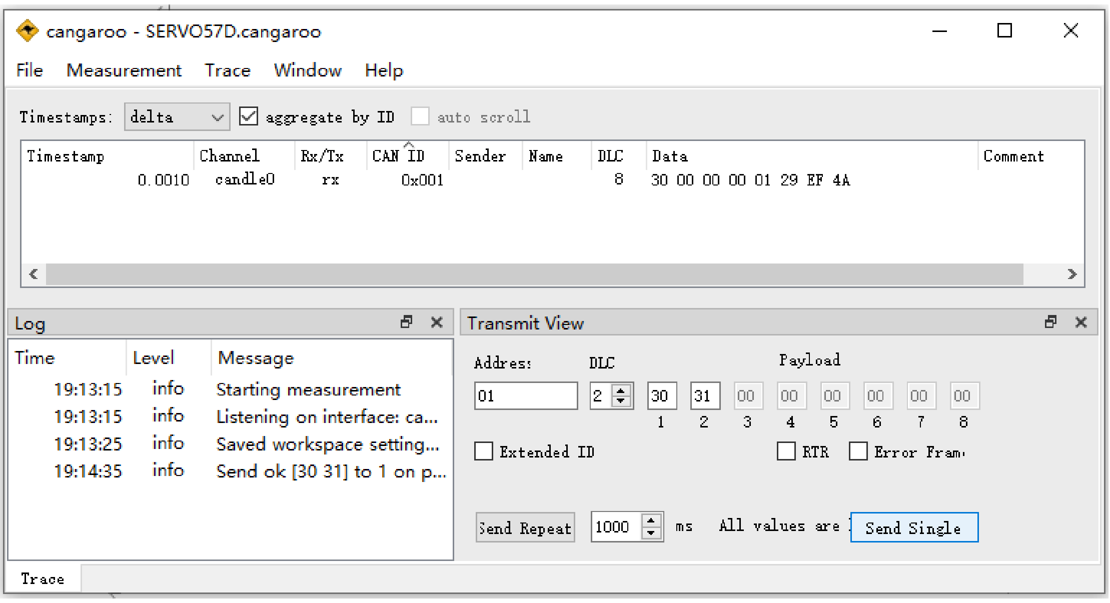


```
Time      Level  Message
19:04:34  info   Starting measurement
19:04:34  info   Listening on interface: candle0
19:10:04  info   Saved workspace settings to ...
```

5.3 Read the encoder value

send “01 30 31”

return “01 30 00 00 00 01 29 EF 4A”


cangaroo - SERVO57D.cangaroo
File Measurement Trace Window Help
Log
Transmit View
Send Repeat
ms
All values are
Send Single
Trace

Timestamps: delta ☑ aggregate by ID □ auto scroll

Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
0.0010	candle0	rx	0x001			8	30 00 00 00 01 29 EF 4A	

Time	Level	Message
19:13:15	info	Starting measurement
19:13:15	info	Listening on interface: ca...
19:13:25	info	Saved workspace setting...
19:14:35	info	Send ok [30 31] to 1 on p...

Address: 01 DLC: 2 Payload: 1: 30 2: 31 3: 00 4: 00 5: 00 6: 00 7: 00 8: 00

□ Extended ID □ RTR □ Error Frame

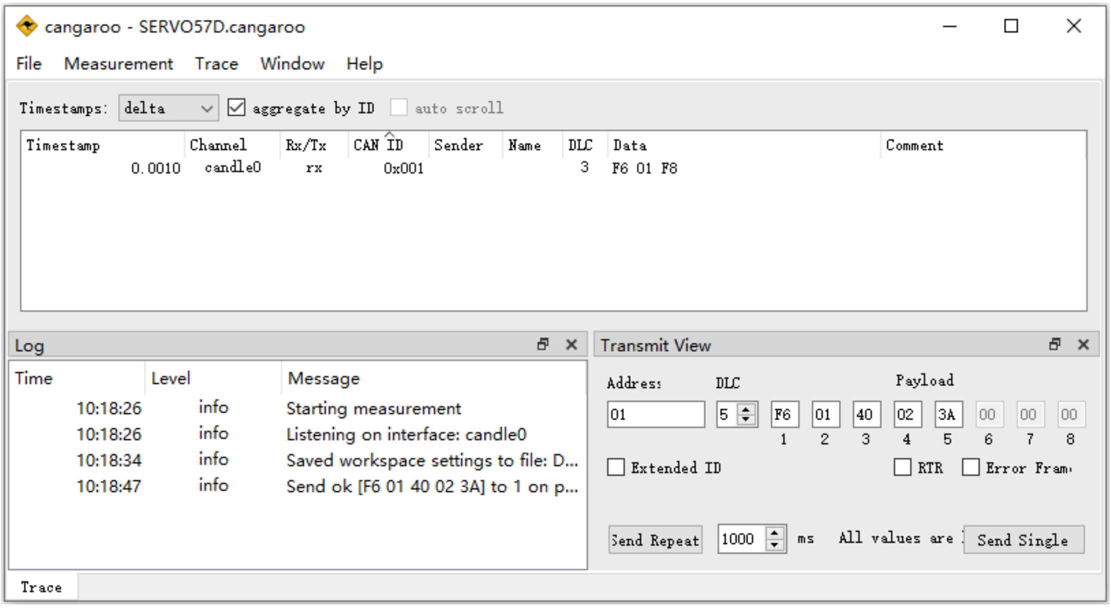
Send Repeat 1000 ms

5.4 Run the motor in speed mode

Note : Please configure the working mode to 05 RS485 bus closed loop FOC mode.

1. Send 01 F6 01 40 02 3A, the motor will rotate at ″speed = 600RPM, acc=2″;

Return 01 F6 01 F8, the motor run in speed mode successful;



cangaroo - SERVO57D.cangaroo

Timestamps: delta ☑ aggregate by ID ☐ auto scroll

```
Timestamp   Channel   Rx/Tx   CAN ID   Sender   Name   DLC   Data       Comment
   0.0010   candle0    rx     0x001                     3    F6 01 F8
```

Log

```
Time       Level   Message
10:18:26   info    Starting measurement
10:18:26   info    Listening on interface: candle0
10:18:34   info    Saved workspace settings to file: D...
10:18:47   info    Send ok [F6 01 40 02 3A] to 1 on p...
```

Transmit View

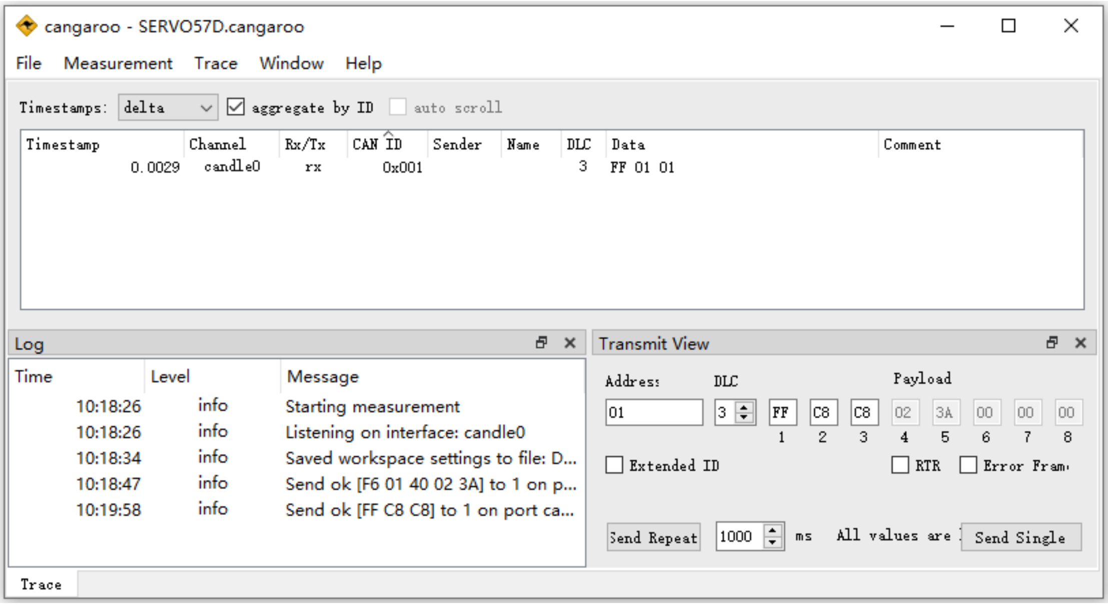
Address	DLC	Payload 1	2	3	4	5	6	7	8
01	5	F6	01	40	02	3A	00	00	00

☐ Extended ID ☐ RTR ☐ Error Fram

1000 ms

2. Send 01 FF C8 C8 to save the speed mode parameters;

Return 01 FF 01 01, save successful;



cangaroo - SERVO57D.cangaroo

Timestamps: delta ☑ aggregate by ID ☐ auto scroll

```
Timestamp   Channel   Rx/Tx   CAN ID   Sender   Name   DLC   Data       Comment
   0.0029   candle0    rx     0x001                     3    FF 01 01
```

Log

```
Time       Level   Message
10:18:26   info    Starting measurement
10:18:26   info    Listening on interface: candle0
10:18:34   info    Saved workspace settings to file: D...
10:18:47   info    Send ok [F6 01 40 02 3A] to 1 on p...
10:19:58   info    Send ok [FF C8 C8] to 1 on port ca...
```

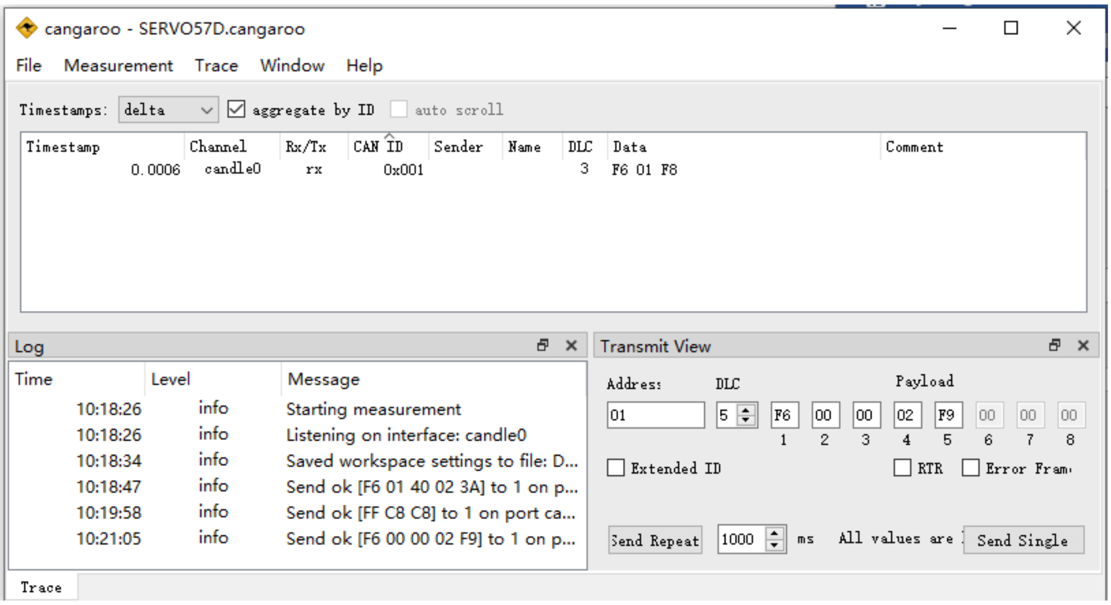
Transmit View

Address	DLC	Payload 1	2	3	4	5	6	7	8
01	3	FF	C8	C8	02	3A	00	00	00

☐ Extended ID ☐ RTR ☐ Error Fram

1000 ms

3. Send 01 F6 00 00 02 F9 to stop the motor;
Return 01 F6 01 F8, the motor stops successfully;


cangaroo - SERVO57D.cangaroo
File Measurement Trace Window Help
Log
Transmit View
Send Repeat
ms
All values are
Send Single
Trace

Timestamps: delta ☑ aggregate by ID □ auto scroll

```
Timestamp    Channel   Rx/Tx   CAN ID   Sender   Name   DLC   Data        Comment
   0.0006    candle0    rx     0x001                      3   F6 01 F8
```

```
Time       Level   Message
10:18:26   info    Starting measurement
10:18:26   info    Listening on interface: candle0
10:18:34   info    Saved workspace settings to file: D...
10:18:47   info    Send ok [F6 01 40 02 3A] to 1 on p...
10:19:58   info    Send ok [FF C8 C8] to 1 on port ca...
10:21:05   info    Send ok [F6 00 00 02 F9] to 1 on p...
```

Address: 01 DLC: 5 Payload: 1: F6 2: 00 3: 00 4: 02 5: F9 6: 00 7: 00 8: 00

□ Extended ID □ RTR □ Error Fram

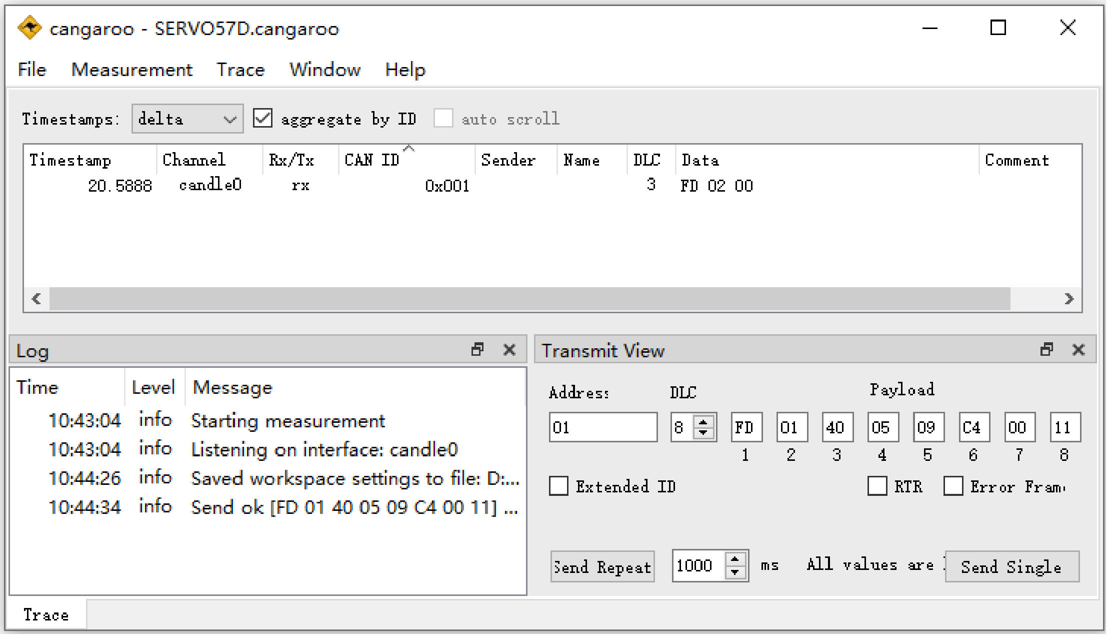
1000 ms

After power-on again, the motor will run according to the save speed mode parameters.

5.5 Run the motor in position mode1

Note : Please configure the working mode to 05 RS485 bus closed loop FOC mode.

1. Send 01 FD 01 40 05 09 C4 00 11, the motor will rotate forward 200 circles (16 subdivisions) with ″speed = 320RPM, acc = 5″;
Return 01 FD 01 FF, the motor starts to run;
Return 01 FD 02 00, the motor is run completed;


cangaroo - SERVO57D.cangaroo
File Measurement Trace Window Help
Log
Transmit View
Trace

Timestamps: delta ☑ aggregate by ID □ auto scroll

Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
20.5888	candle0	rx	0x001			3	FD 02 00	

Time	Level	Message
10:43:04	info	Starting measurement
10:43:04	info	Listening on interface: candle0
10:44:26	info	Saved workspace settings to file: D:...
10:44:34	info	Send ok [FD 01 40 05 09 C4 00 11] ...

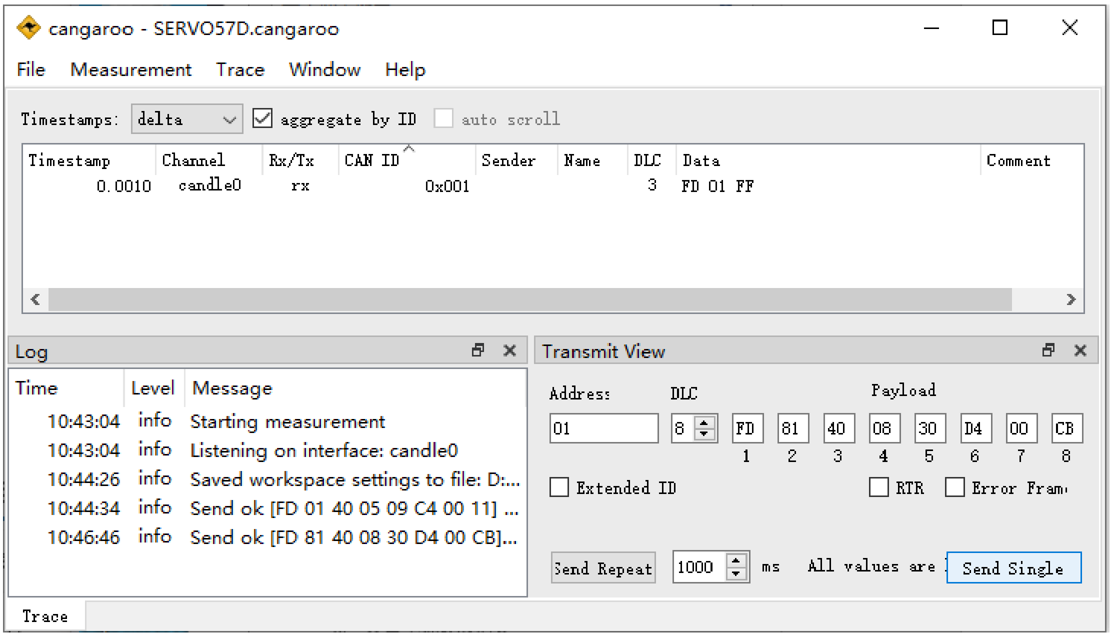
Address: 01 DLC: 8

Payload	1	2	3	4	5	6	7	8
	FD	01	40	05	09	C4	00	11

□ Extended ID □ RTR □ Error Fram

Send Repeat 1000 ms All values are Send Single

2. Send 01 FD 81 40 08 30 D4 00 CB, the motor to reverse 1000 circles with ″speed = 320RPM, acc = 8″ (16 subdivisions);
Return 01 FD 01 FF, the motor starts to run.
Now, stop the motor by step 3.


cangaroo - SERVO57D.cangaroo
File Measurement Trace Window Help
Log
Transmit View
Trace

Timestamps: delta ☑ aggregate by ID □ auto scroll

Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
0.0010	candle0	rx	0x001			3	FD 01 FF	

Time	Level	Message
10:43:04	info	Starting measurement
10:43:04	info	Listening on interface: candle0
10:44:26	info	Saved workspace settings to file: D:...
10:44:34	info	Send ok [FD 01 40 05 09 C4 00 11] ...
10:46:46	info	Send ok [FD 81 40 08 30 D4 00 CB]...

Address: 01 DLC: 8

Payload	1	2	3	4	5	6	7	8
	FD	81	40	08	30	D4	00	CB

□ Extended ID □ RTR □ Error Fram

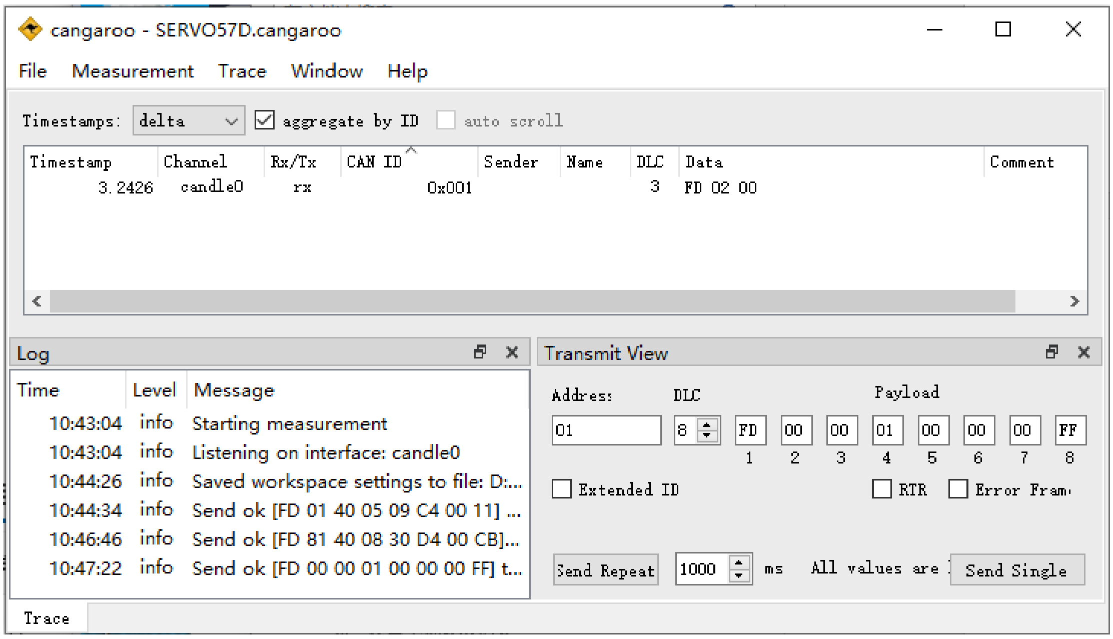
Send Repeat 1000 ms All values are Send Single

3. While the motor is running…

Send 01 FD 00 00 01 00 00 00 FF, the motor to stop with acc=1;

Return 01 FD 01 FF, the motor starting to stop;

Return 01 FD 02 00, the motor has stopped;


cangaroo - SERVO57D.cangaroo
File Measurement Trace Window Help
Log
Transmit View
Trace
Send Repeat
Send Single

Timestamps: delta ☑ aggregate by ID □ auto scroll

Timestamp	Channel	Rx/Tx	CAN ID	Sender	Name	DLC	Data	Comment
3.2426	candle0	rx	0x001			3	FD 02 00	

Time	Level	Message
10:43:04	info	Starting measurement
10:43:04	info	Listening on interface: candle0
10:44:26	info	Saved workspace settings to file: D:...
10:44:34	info	Send ok [FD 01 40 05 09 C4 00 11] ...
10:46:46	info	Send ok [FD 81 40 08 30 D4 00 CB]...
10:47:22	info	Send ok [FD 00 00 01 00 00 00 FF] t...

Address: 01 DLC: 8

Payload	1	2	3	4	5	6	7	8
	FD	00	00	01	00	00	00	FF

□ Extended ID □ RTR □ Error Fram

1000 ms All values are

Part6. FAQ

6.1 NOTE

1. Power input voltage is 12V-24V.
2. Don’t hot plug motor cable and data cable.
3. The phase lines A+, A -/B+, B - should be connected correspondingly. （A -, A+/B+, B - is incorrect）

6.2 FAQ

No	Question	Solution
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

Part7. Schematic

Part8. contact us

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