MKS SERVO42D/57D_CAN V1.0.5

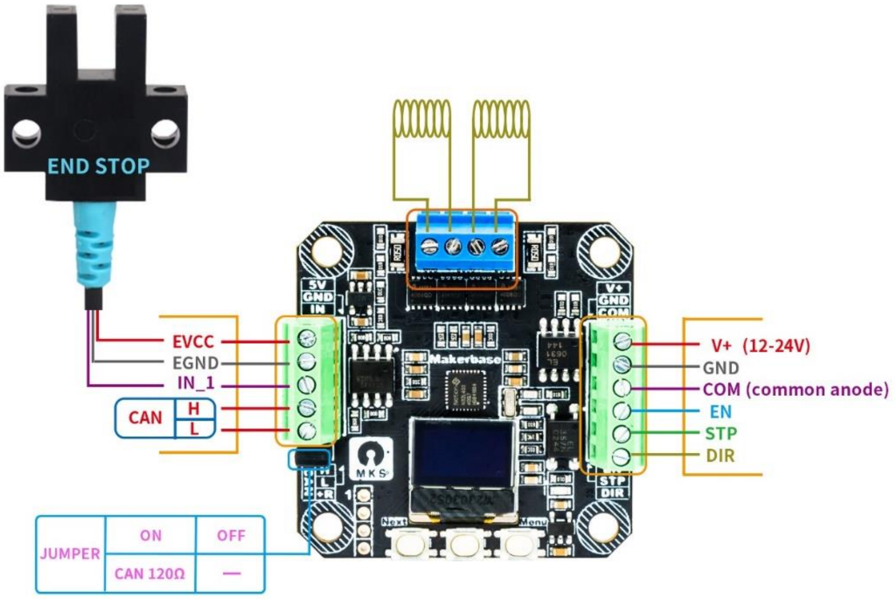
USER MANUAL

MKS SERVO42D/57D_CAN Manual Release			
manual	discription	firmware	date
V1.0.0	First release	V1.0.0	Mar-2023
V1.0.1	1.Added SR_OPEN, SR_CLOSE control mode.	V1.0.1	Apr-2023
	2.It can set any working current.		
	3.Redefined speed and acceleration for serial Mode.		
	4. Add the "92" command , It can set the current position to 0 point.		
	5. Add the "8D" command, It can set the group address.		
V1.0.2	1.Slave does not answer if broadcast address or group address is used.	V1.0.2	May-2023
	2.OUT_1 port output stall indication.		
V1.0.3	1. Add the "9A" command, It can set the parameter of 0_Mode.	V1.0.3	Jul-2023
	2. Add the "8F" command, It can locked the key.		
	3. Add the "34" command, It can read the IO Ports status.		
	4. the number of slave addresses can be set by menu is change to 16.		
	5. add left and right endstop limit function.		
V1.0.4	1. Added menu or command (9BH) to set holding current percentage function.	V1.0.4	Sep-2023
	2. Added absolute motion by pulses(FEH).		
	3. Modify the 8CH command and add the option of active.		
	4. Add emergency stop command(F7H).		
	5. Add limit port remap command (9EH).		
V1.0.5	1. Support no limit switch for “go home” function.	V1.0.5	May-2024
	2. Added menu options: “Hm_Mode” and “Hm_Ma”.		
	3. Add the "94H" command.		
	4. Fix the bug of command “F4H” and “F5H”.		
	5. The command”F5H”supports real-time data update.		
	6. Add the restart motor command(41H).		
V1.0.6	1. Add the "35H" command to read RAW data of encoder.	V1.0.6	Sep-2024
	2. Add the “9DH”command , En triggers single-turn zero return and position error protection function.		
	3. Read system parameter function, see section 5.9.		
	4. It can release the protection status by En.		
	5. Add the “36H”command to write the IO port.		

Part1. Feature

1.1 Interface

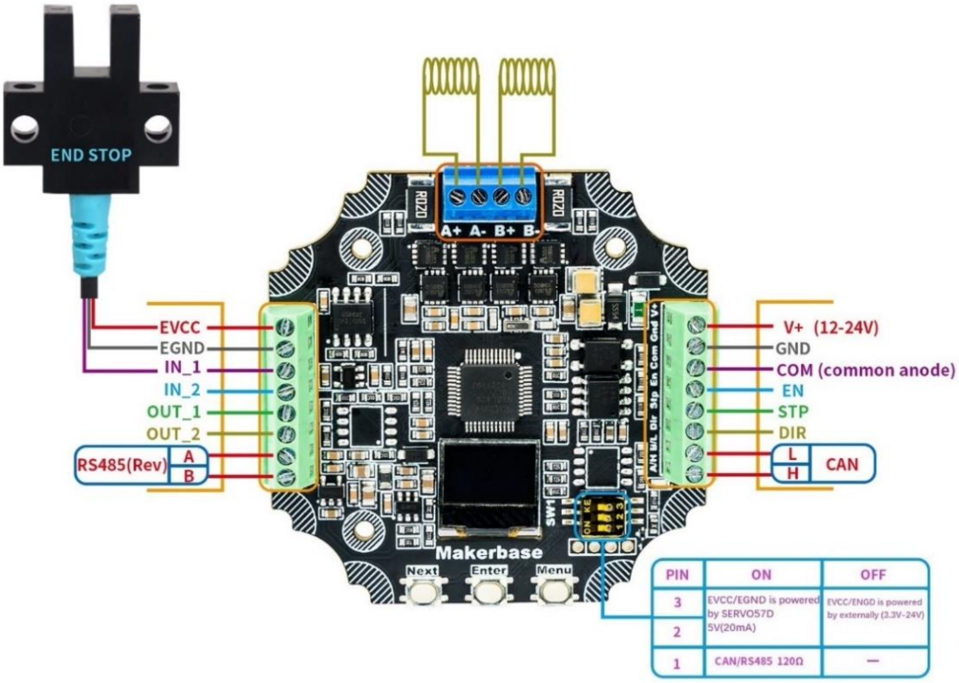
1. SERVO42D_CAN Interface



JUMPER	ON	OFF
	CAN 120Ω	—

Note: EVCC/EGND is powered by SERVO42D 5V(20mA)

2. SERVO57D_CAN Interface



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

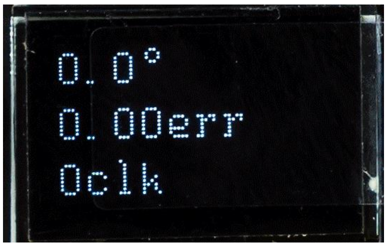
1.2 Key Operation

Key	Function
Next	move down
Enter	Confirm
Menu	Enter/exit parameter setting menu

3. How to View parameter
    Press the “Menu” key to Enter the Menu
    press the “Next” key to move to the sub-option
    press the “Enter” key, then it show the value.
4. How to setting Parameter:
    Press the “Menu” key to Enter the Menu
    press the “Next” key to move to sub-option
    press the “Enter” key, it show the value.
    press the “Next” key to move to the value
    press the “Enter” key to set the value.

1.3 Parameter description

1. 0.0° - the angle of the motor shaft.(unit degree).
   (Note : It calculated based on the read encoder value, dynamically displayed)
2. 0.00err - the err of the motor shaft angle.
3. 0clk - the pulses have been received.



0.0°
0.00err
0clk

1.4 Work mode

	Work Mode		MAX RPM	Work Current
OPEN	pulse interface	CR_OPEN	400RPM	Fix, the work current is Ma
	serial interface	SR_OPEN		
CLOSE	pulse interface	CR_CLOSE	1500RPM	Fix, the work current is Ma
	serial interface	SR_CLSOE		
vFOC	pulse interface	CR_vFOC	3000RPM	self-adaption, the Max current is Ma
	serial interface	SR_vFOC		

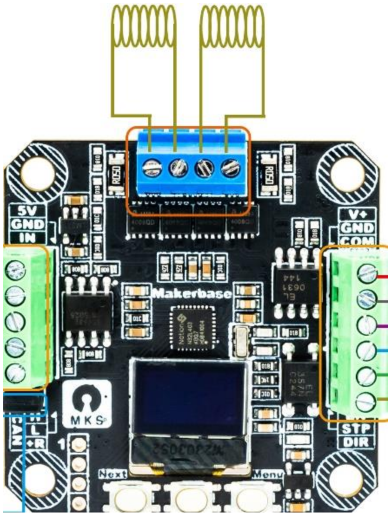
Note1: The default work mode is CR_vFOC.
Note2: It can work without encoder in “OPEN” work mode.

Part2. Wire

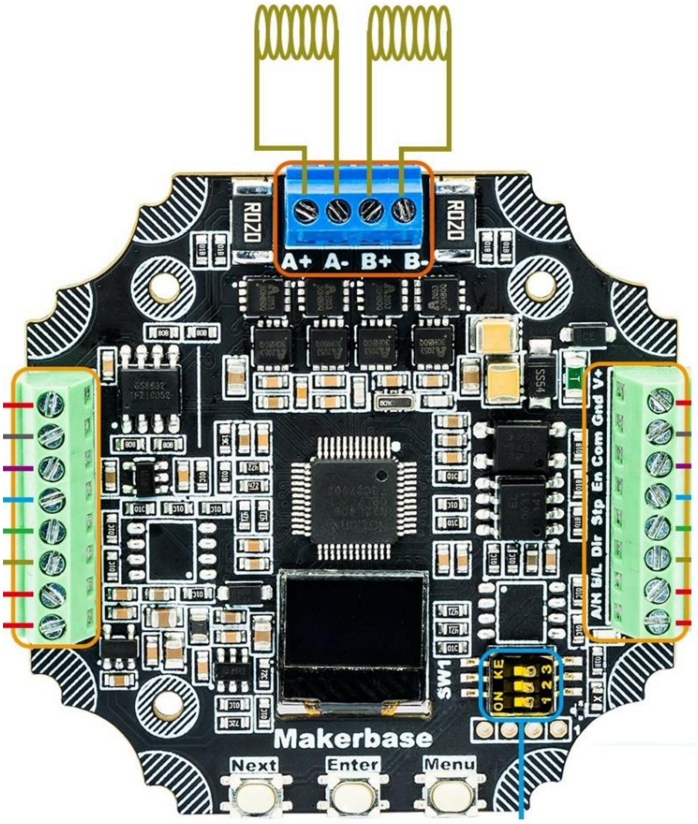
2.1 Motor wire

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

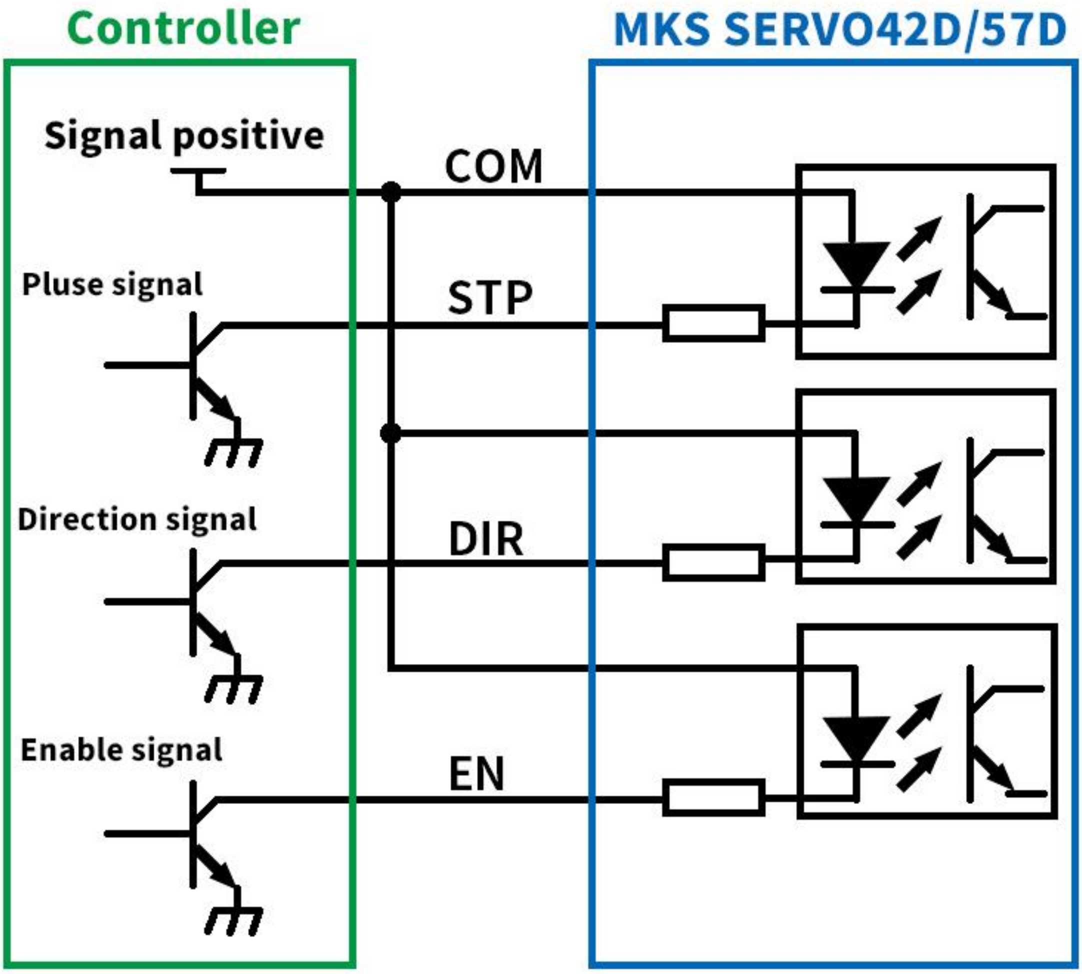
1. SERVO42D_CAN motor wire



2. SERVO57D_CAN motor wire



2.2 Pulse interface wire

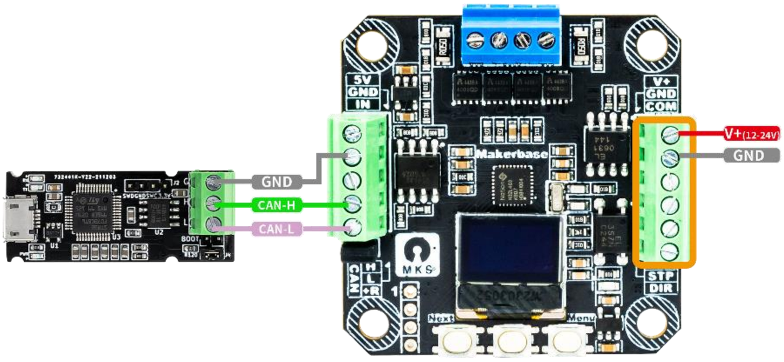


Note: if the (STP/DIR/EN)signal high level is 3.3V,the COM must be 3.3V
if the (STP/DIR/EN)signal high level is 5.0V,the COM must be 5.0V
…
and so on.

2.3 CAN wire

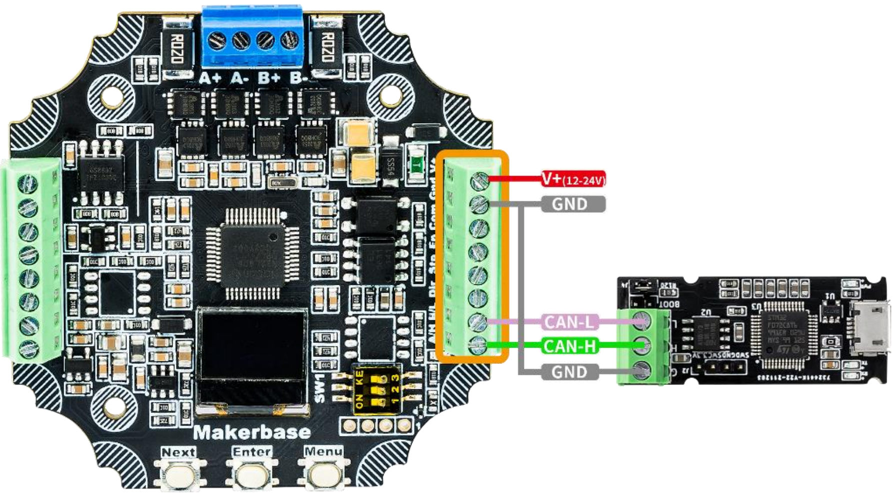
Note: Tip: In order to reduce bus interference, the host-gnd and the motor-gnd must connected togeter, and CAN signals are transmitted using shielded twisted pairs.

1. SERVO42D_CAN Single-slave



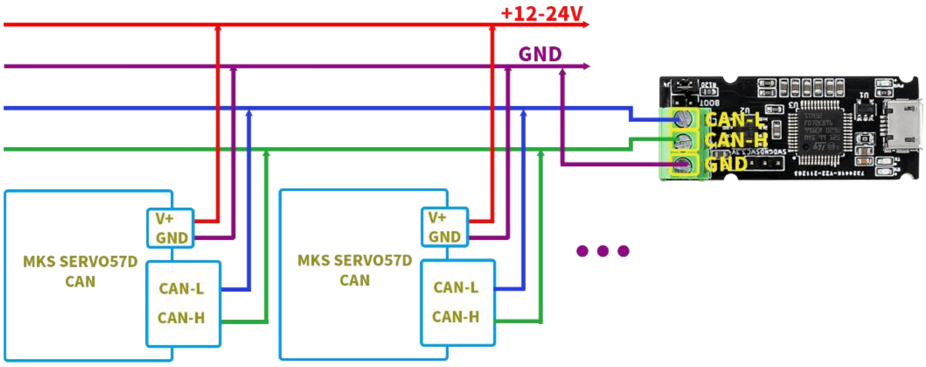
Note: Single slave communication does not need 120Ω Terminal.

2. SERVO57D_CAN Single-slave



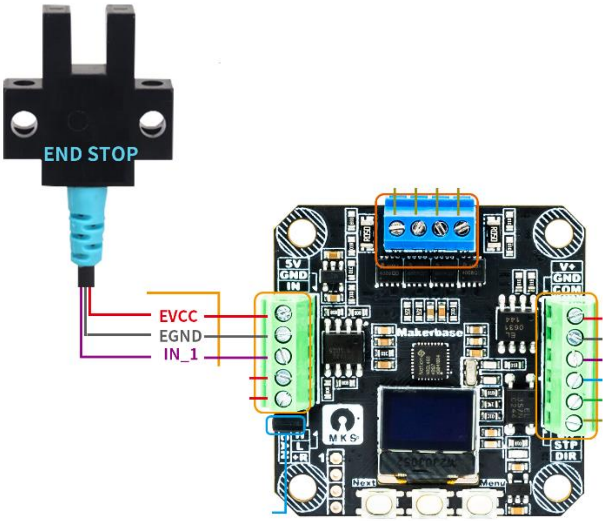
Note: Single slave communication does not need 120Ω Terminal.

3. Multiple-slave



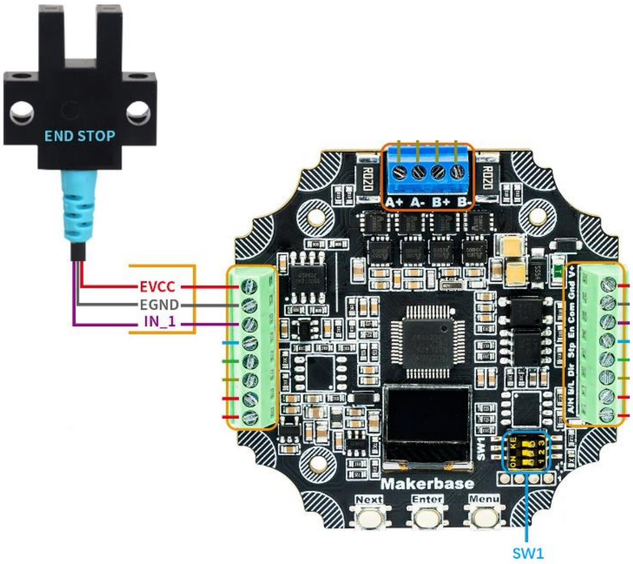
2.4 End stop wire

1. SERVO42D_CAN end stop wire



Note: EVCC/EGND is powered by SERVO42D 5V(20mA)

2. SERVO57D_CAN end stop wire



SW1		
PIN	ON	OFF
3	EVCC/EGND is powered by SERVO57D 5V(20mA)	EVCC/EGND is powered by externally power.(3.3V-24V)
2		
1	CAN 120Ω Terminal	NULL

Note: The mechanical switch only needs to be connected the ″EGND, IN_1″, and the SW1 pin2 must be in the ON state.

2.5 IO Port Description

PORT	Function	57D	42D
IN_1	home or left-limit	√	√
IN_2	right-limit	√	X
OUT_1	stall indication: 0-protected; 1-unprotected	√	X
OUT_2	reserved	√	X

Note: After enabling the limit remapping function, IN_1 maps to En, IN_2 maps to Dir.

2.6 EndStop-limit Description

1. The EndStop-limit function needs to be turned on:
   (Menu -> EndLimit or serial command ″90″)
2. When first time to using the limit function or changing the limit parameters, it is necessary to go home;
   (Menu -> GoHome or serial command ″91″)
3. After the left-endstop is triggered, the motor will no longer run to the left;
4. After the right-endstop is triggered, the motor will no longer run to the right; (only for 57D)
5. Limit remapping function can be turned on (serial mode only)
   Left limit -> En port
   Right limit -> Dir port
   The Com port must be connected to the corresponding high level

Part3. Menu description

1. **CAL** : Calibrate the motor.

2. **Mode** : Work mode selection.

CR_OPEN : pulse interface Open mode, the motor run without encoder
CR_CLOSE : pulse interface Close mode, the motor run with encoder.
CR_vFOC : pulse interface FOC mode, the motor run with encoder.
SR_OPEN : serial interface Open mode, the motor run without encoder
SR_CLOSE : serial interface Close mode, the motor run with encoder.
SR_vFOC : serial interface FOC mode, the motor run with encoder.
(Default: CR_vFOC)

	Mode		MAX RPM	Work Current
OPEN	pulse interface	CR_OPEN	400RPM	Fix, the work current is Ma
	serial interface	SR_OPEN		
CLOSE	pulse interface	CR_CLOSE	1500RPM	Fix, the work current is Ma
	serial interface	SR_CLSOE		
vFOC	pulse interface	CR_vFOC	3000RPM	self-adaption, the Max current is Ma
	serial interface	SR_vFOC		

Note: CR_CLOSE is better than CR_vFOC for 3D printing.

3. **Ma** : Set the working current.

SERVO42D: 0, 200, 400..., 3000(mA) (default 1600mA)
SERVO57D: 0, 400, 800..., 5200(mA) (default 3200mA)
SERVO28D: 0, 200, 400..., 3000(mA) (default 600mA)
SERVO35D: 0, 200, 400..., 3000(mA) (default 800mA)

Other Current such as 123mA need to be set by serial command .It will be added to the last options.

4. **HoldMa** : Set holding current percentage.

10%, 20%, ..., 90%
(Default: 50%, the holding current at half the working current)
Note: Only for OPEN mode and CLOSE mode, vFOC mode is invalid.

5. **MStep** : Set subdivisions.

Supports subdivision from 1 to 256.
(Default: 16)
subdivisions 1,2,4,8,16,32,64,128, and 256 can be set by Menu.

Other subdivisions such as 67 subdivisions need to be set by serial command.

6. **En** : Set the effective level of EN pin.

H : High level is valid.

L : Low level is effective.

Hold : the driver board is always enabled.

(Default: L)

7. **Dir** : Set the positive direction of motor rotation.

CW : Clockwise rotation is positive

CCW : Counterclockwise rotation is positive

(Default: CW)

Note: onldy for pulse interface,the direction of serial interface is set by command.

8. **AutoSDD** : Set auto turn off the OLED screen.

Disable : disable auto turn off the OLED

Enable : enable auto turn off the OLED

(Default: Disable)

If set to Enable, the screen will automatically turn off after about 15 seconds, and any button can wake up the screen again.

9. **Protect** : Set the motor shaft locked-rotor protection function.

Disable: disable protection

Enable: enable protection

(Default: Disable)

After this option is enabled, the protection will be triggered when it is detected to be locked-rotor, and the motor will be release.

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

1. Press the Enter button to release the stall protection;
2. Use the serial port command (3D) to release the stall protection;
3. In the pulse interface mode, the En signal is invalid (not locking the shaft), and the stall protection can be released.

10. **MPlyer** : Set internal 256 subdivision.

(Default: Enable)

Note: After this option is Enabled, it automatically enable internal 256 subdivision, it can reduce the vibration and noise when the motor at low speed.

11. **CanRate** : Set the bit rate of CAN interface.

125K，250K，500K,1M。

(Default: 500K)

12. **CanID**: Set the the slave address of CAN interface.
    01
    …
    09
    10
    (Default: 01)
    Note: The addresses greater than 10 need to be set by serial command. After it is set, it will be added to this option.

13. **CanRSP** : Choose whether the slave respond in speed/positon mode.
    Disable: disable respond
    Enable: enable respond
    (Default: Enable)
    Note: If disable respond, It can query the running status of the motor by serial command "F1".

14. **0_Mode** : The motor will go back to zero when power on.
    Disable : do not go back to zero.
    DirMode : go back to zero with direction of CW or CCW (the direction is set in 0_Dir menu).
    NearMode : go back to zero with minimum angle.
    (Default: Disable)

15. **Set 0** : Set the zero point for go back when power on.
    (0_Mode must not be Disable)

16. **0_Speed** : Set the speed of go back to zero point.
    0 : slowest.
    ...
    4 : fastest.

17. **0_Dir** : Set the direction of go back to zero point.
    CW : Clockwise.
    CCW : Counterclockwise.
    (Default: CW)

18. **Hm_Trig** : Set the effective level of the end stop.
    Low : Low level is effective
    High : High level is valid
    (Default: Low)

19. **Hm_Dir** ： Set the direction of go home。

CW ： Clockwise rotation is positive

CCW ： Counterclockwise rotation is positive

(Default: CW)

20. **Hm_Speed** ： Set the speed (RPM) of go home。

30
60
90
120
150
180

Other speed such as 600(RPM) need to be set by serial command . It will be added to the last options.

21. **Hm_Mode** ： Set the method of go home.

Limited ： used Limit switch for go home(default)

noLimit ： no Limit switch for go home

When “noLimit” for go home, the motor will runs with a fixed torque (Hm_Ma setting) until it stops when it encounters an obstacle, and then runs in reverse for a certain distance (94H command setting) and then stops. The stopping point is the zero point.

22. **Hm_Ma** ： Set the current of “noLimit” go home.

SERVO42D: 0, 200, 400..., 3000(mA) (default 800mA)

SERVO57D: 0, 400, 800..., 5200(mA) (default 400mA)

SERVO28D: 0, 200, 400..., 3000(mA) (default 200mA)

SERVO35D: 0, 200, 400..., 3000(mA) (default 200mA)

Note: The “Hm_Ma” is only valid during “noLimit” go home operation. It should be set to a smaller current as much as possible to avoid damaging the motor.

23. **EndLimit** : Set the endstop-limit function.

Disable: disable endstop-limit

Enable: enable endstop-limit

(Default: Disable)

Note1:When first time to using the “EndLimit” function or changing the limit parameters, it is necessary to go home;
(Menu -> GoHome or serial command ″91″)

Note 2: When EndLimit=Enable, when the motor goHome touches the left limit switch, the motor will unlocked the shaft.

If you want to locked the axis after goHome, you should set EndLimit=Disable.

24. **GoHome：** Go home
    Note1: It need an “end stop”. The motor will keep running until it hits the limit switch.
    Note2: If the limit switch is already closed, the motor will rotate in the opposite direction to homeDir until the limit switch is opened, and then go home.

25. **Restore** : Reload the default parameters.
    After restored the default parameters, it needs to Calibrate the motor.
    Note: Press the "Next" key first, then power on, it can quickly restore the default parameters.

26. **About** : Show version parameters.

27. **Exit** :Exit the parameter setting menu.

Part4. CAN data format

The CAN uses standard frames.

Downlink package(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2…byte(n-1)	Byte(n)(n≤8)
ID		DLC(n)	code	data	Check(CRC)

Uplink frame (PC ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2…byte(n-1)	Byte(n) (n≤8)
ID		DLC(n)	(code)	data	Check (CRC)

1. The CAN ID range is 00~2047. (default is 01) .
   00 is the broadcast address;
   01~10 can be set in the CanID option of the display menu;
   greater than 10 need to be set by serial commands.
2. The function code (code) executes the corresponding command.
   for example, 0x80 executes the calibration command.
3. The Check code is CHECKSUM 8bit
   CRC = (ID + byte1 + … + byte(n)) & 0xFF
   For example: command “01 30 CRC”
   CRC = (0x01 + 0x30) & 0xFF = 0x31 & 0xFF = 0x31

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

Part5. CAN command description

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

The default CAN ID for the following chapters is 01.

5.1 Read status parameter command

1. command1 : 01 30 CRC

read the encoder value(carry).

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	30	CRC(31)

Uplink frame (PC ← SERVO42D/57D)							
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.

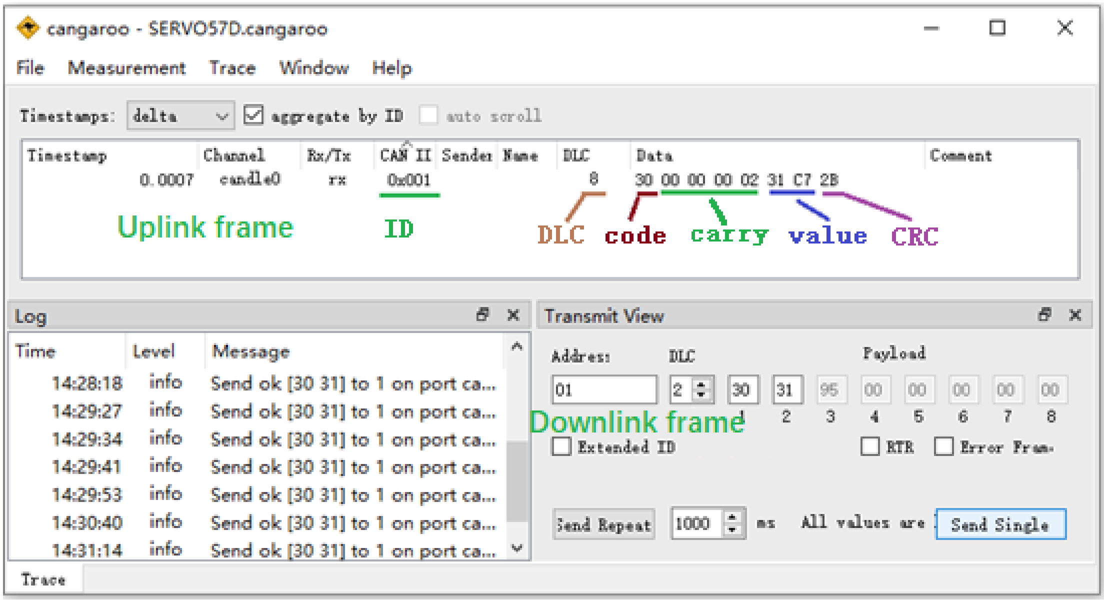
When Value is less than 0, carry -=1.

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:



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	

```
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...
```

Address: 01

DLC: 2

Payload:

1	2	3	4	5	6	7	8
30	31	95	00	00	00	00	00

□ Extended ID □ RTR □ Error Fram-

1000 ms

2. command2 : 01 31 CRC

read the encoder value(addition).

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	31	CRC(32)

Uplink frame (PC ← SERVO42D/57D)					
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 → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	32	CRC(33)

Uplink frame (PC ← SERVO42D/57D)					
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 → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	33	CRC(34)

Uplink frame (PC ← SERVO42D/57D)					
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 → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	34	CRC(3B)

Uplink frame (PC ← SERVO42D/57D)					
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				OUT_2	OUT_1	IN_2	IN_1

Note: After enabling the limit remapping function, IN_1 maps to En, IN_2 maps to Dir.

6. Command6 : 01 35 CRC

read the RAW encoder value(addition).

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	35	CRC

Uplink frame (PC ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2…byte7	byte8
01		8	code	value	Check
			35	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.

7. Command7 : 01 39 CRC

read the error of the motor shaft angle.

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	39	CRC(3A)

Uplink frame（PC ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2…byte5	Byte6
01		4	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.

8. Command8 : 01 3A CRC

read the En pins status.

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	3A	CRC(3B)

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

enable =1 Enabled

enable =0 Disabled

9. Command9 : 01 3B CRC

Read the go back to zero status when power on.

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	3B	CRC(3C)

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

status =0 going to zero.

status =1 go back to zero success.

status =2 go back to zero fail.

10. Command10 : 01 3D CRC

Release the motor shaft locked-rotor protection state.

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	3D	CRC(3E)

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

status =1 release success.

status =0 release fail.

11. Command11 : FA 01 3E CRC

Read the motor shaft protection state.

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	3E	CRC(3F)

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

status =1 protected.

status =0 no protected.

5.2 Set system parameters command

1. Calibrate the encoder

(Same as the **"Cal"** option on screen)

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	80	00	CRC(81)

Uplink frame (PC ← SERVO42D/57D)					
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 motor must be unloaded.

2. Set the work mode

(Same as the **"Mode"** option on screen)

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			82	mode (0~5)	CRC

mode = 0	CR_OPEN
mode = 1	CR_CLOSE
mode = 2	CR_vFOC
mode = 3	SR_OPEN
mode = 4	SR_CLOSE
mode = 5	SR_vFOC

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.

	Mode		MAX RPM	Work Current
OPEN	pulse interface	CR_OPEN	400RPM	Fix, the work current is the **Ma**
	serial interface	SR_OPEN		
CLOSE	pulse interface	CR_CLOSE	1500RPM	Fix, the work current is the **Ma**
	serial interface	SR_CLSOE		
vFOC	pulse interface	CR_vFOC	3000RPM	self-adaption, the Max current is the **Ma**
	serial interface	SR_vFOC		

3. Set the working current

(Same as the **"Ma"** option on screen)

Downlink frame(PC → SERVO42D/57D)					
CAN ID		DLC	byte1	byte2-3	Byte4
01	…	4	code	data	Check
			83	ma (uint16_t)	CRC

Note:the new current will show in the screen of Ma option.
SERVO42D/28D/35D: Maximum Current =3000mA
SERVO57D: Maximum Current =5200mA

Uplink frame (PC ← SERVO42D/57D)					
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 the holding current percentage

(Same as the **"HoldMa"** option on screen)

Downlink frame(PC → SERVO42D/57D)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			9B	holdMa (00~08)	CRC

holdMa = 00 10%
holdMa = 01 20%
…
holdMa = 08 90%

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

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

Note: Only for OPEN mode and CLOSE mode, vFOC mode is invalid.

5. Set subdivision

(Same as the "**MStep**" option on screen)

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

Note:the new micstep will show in the screen of MStep option.

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.

6. Set the active of the En pin

(Same as the "**En**" option on screen)

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			85	enable(00~02)	CRC

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

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

status =1 Set success.

status =0 Set fail.

7. Set the direction of motor rotation

(Same as the "Dir" option on screen)

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

dir = 00　CW

dir = 01　CCW

Uplink frame（PC ← SERVO42D/57D)					
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: onldy for pulse interface, the direction of serial interface is set by command.

8. Set auto turn off the screen function

(Same as the " AutoSDD " option on screen)

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

enable = 01　enabled

enable = 00　disabled

If set to Enable, the screen will automatically turn off after about 15 seconds, and any button can wake up the screen again.

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

status =1　Set success.

status =0　Set fail.

9. Set the motor shaft locked-rotor protection function

(Same as the "**Protect**" option on screen)

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

enable = 01 enabled protection

enable = 00 disabled protection

Uplink frame (PC ← SERVO42D/57D)					
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 is activated, there are three ways to release it:

1. Press the Enter button to release the stall protection;
2. Use the serial port command (3D) to release the stall protection;
3. In the pulse interface mode, the En signal is invalid (not locking the shaft), and the stall protection can be released.

10. Set the subdivision interpolation function

(Same as the "**Mplyer**" option on screen)

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

enable = 01 enabled interpolation function.

enable = 00 disabled interpolation function.

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

status =1 Set success.

status =0 Set fail.

11. Set the CAN bitRate

(Same as the **"CanRate"** option on screen)

Downlink frame(PC → SERVO42D/57D)					
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

bitRate = 03 1M

Uplink frame（PC ← SERVO42D/57D)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			8A	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

12. Set the CAN ID

(Same as the **"CanID"** option on screen)

Downlink frame(PC → SERVO42D/57D)						
CAN ID		DLC	byte1	byte2	byte3	byte4
01	…	4	code	data		Check
			8B	ID(00~7FF)		CRC

Note1:the new address will show in the screen of CanID option.

Note2: 0 is the broadcast address

Uplink frame（PC ← SERVO42D/57D)					
CAN ID		DLC	byte1	byte2	byte3
01	…	3	code	data	Check
			8B	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

13. Set the slave respond and active

Downlink frame(PC → SERVO42D/57D)						
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 ← SERVO42D/57D)					
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 FA 01 FD 02 80 02 00 00 FA 00 76

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.

14. Set the key lock or unlock

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8F	enable(00~01)	CRC

enable = 01 lock the key

enable = 00 unlock the key

Uplink frame (PC ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			8F	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

15. Set the group ID

Downlink frame(PC → SERVO42D/57D)						
CAN ID	…	DLC	byte1	byte2	byte3	byte4
01		4	code	data		Check
			8D	ID(01~0x7FF)		CRC

Uplink frame (PC ← SERVO42D/57D)					
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.

5.3 Write IO port command

Downlink frame(PC → SERVO42D/57D)												
CAN ID	…	DLC	byte1	Byte2								Byte3
01		3	code	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Check
			36	OUT2_mask		OUT1_mask		OUT_2	OUT_1	0		CRC

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

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

OUT_2 OUT_2 port write value (0/1)

OUT_1 OUT_1 port write value (0/1)

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

status =1 write success.

status =0 write fail.

5.4 Set Home command

1. Set the parameter of home

(Same as the “HmTrig、HmDir、HmSpeed” option on screen)

Downlink frame(PC → SERVO42D/57D)									
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

homeTrig the effective level of the end stop
0: Low 1: High

homeDir the direction of go home
0: CW 1: CCW

homeSpeed the speed of go home
0~3000 (RPM)

EndLimit
0: disable endstop-limit
1: enable endstop-limit

mode 0: used Limit switch for go home
1: no Limit switch for go home

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

Uplink frame（PC ← SERVO42D/57D）					
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 first time to using the “EndLimit” function or changing the limit parameters, it is necessary to go home;

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

Note 2: When EndLimit=1, when the motor goHome touches the left limit switch, the motor will unlocked the shaft.

If you want to locked the axis after goHome, you should set EndLimit=0.

2. Go home

(Same as the “GoHome” option on screen)

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	91	CRC(92)

Note1: If the limit switch is already closed, the motor will rotate in the opposite direction to homeDir until the limit switch is opened, and then go home.

Uplink frame (PC ← SERVO42D/57D)					
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 → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	92	CRC(93)

Uplink frame (PC ← SERVO42D/57D)					
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 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

ma: the current of “noLimit”go home

retValue: 0~0xFFFFFFFF

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: Reference the menu of “Hm_Mode” and “Hm_Ma”.

Note2: Reference the Command 90/91.

5. Set limit port remap

(only for serial control mode)

The 28/35/42D motor has only a left limit port. In serial control mode, limit port remapping can be enabled to add a right limit port.

For the 57D motor, limit port remapping can also be enabled if required to facilitate wiring.

Left limit -> En port

Right limit -> Dir port

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

Downlink frame(PC → SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			9E	enable(00~01)	CRC

enable = 01 enable remap limit port

enable = 00 disable remap limit port

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

status =1 Set success.

status =0 Set fail.

5.5 Set 0_Mode command

In 0_Mode, the motor can automatically return to the 0 point position when power on. The maximum angle is 359 degrees.

1. Set the parameter of 0_Mode

(Same as the **“0_Mode、Set 0、0_Speed、0_Dir”** option on screen)

Downlink frame(PC → SERVO42D/57D)								
CAN ID	…	DLC	byte1	byte2	byte3	byte4	Byte5	Byte6
01		5	Function	0_Mode	Set 0	0_Speed	0_Dir	Check
			9A	mode	enable	speed	dir	CRC

mode:

0: Disable do not go back to zero

1: DirMode go back to zero with direction

2: NearMode go back to zero with minimum angle

enable:

0: clean zero

1: set zero

2: not modify the zero

speed:

0 ~ 4 (0:slowest 4:fastest)

dir:

0: CW

1: CCW

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

status =0 set fail.

status =1 set success.

5.6 Restore the default parameter

(Same as the "Restore" option on screen)

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	3F	CRC(40)

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

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

Note1: After restored the parameters, It will reboot again,and need to calibrate the motor.
Note2: Press the "Next" key, then power on the motor, the default parameter will be restored.

5.7 Restart the motor

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	41	CRC

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

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

5.8 En triggers single-turn zero return and position error protection function

1. En-triggered single-turn zeroing function description:

In pulse control mode, after setting the power-on single-turn zeroing function and turning on the En-triggered zeroing function, the motor will no longer automatically return to zero when powered on, but will be triggered to return to zero through the 200ms width pulse signal of the En signal line.

2. Position error protection description:

Position error protection and stall protection can be turned on or off separately. They monitor the motor independently. When any protection condition is triggered, the motor can be started to protect it.

The triggering condition of position error protection is: within x time, if the motor position error is greater than y, the protection is started. (x, y can be set)

Note: When the stall protection is triggered, the screen displays "Wrong..."
When the position error protection is triggered, the screen displays "Wrong2..."

The instructions are as follows:

Downlink frame(PC → SERVO42D/57D)									
CAN ID	…	DLC	byte1	Byte2			byte3-4	byte5-6	byte7
01		7	code	rev	g0En	pEn	times	errors	Check
			9D	b7-b2	b1	b0	Tim	Errors	CRC
				0	g0En	pEn			

g0En 0: Disable En trigger return to zero function (default value)
1: Enable En trigger return to zero function

Note: When enabled, the motor automatically returns to zero position in a single turn after detecting the En enable signal pulse of about 200ms. It will no longer automatically return to zero position in a single turn when powered on.

pEn 0: Disable position error protection (default value)
1: Enable position error protection

Tim: uint16_t Set the error statistics time length

Note: 1 Tim unit is approximately equal to 15ms

Errors: uint16_t Set the number of startup protection errors

Note: When Errors = 28000, the motor is misaligned by 360 degrees

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

status =1 Write success.
status =0 Write fail.

5.9 Read system Parameter command

The command format for reading system parameters is as follows:

Downlink frame(PC → SERVO42D/57D)					
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 ← SERVO42D/57D)					
CAN ID	…	DLC	byte 1	byte 2-n	byte n+1
01		n+1	CODE	parameters	Check
			code	param	CRC

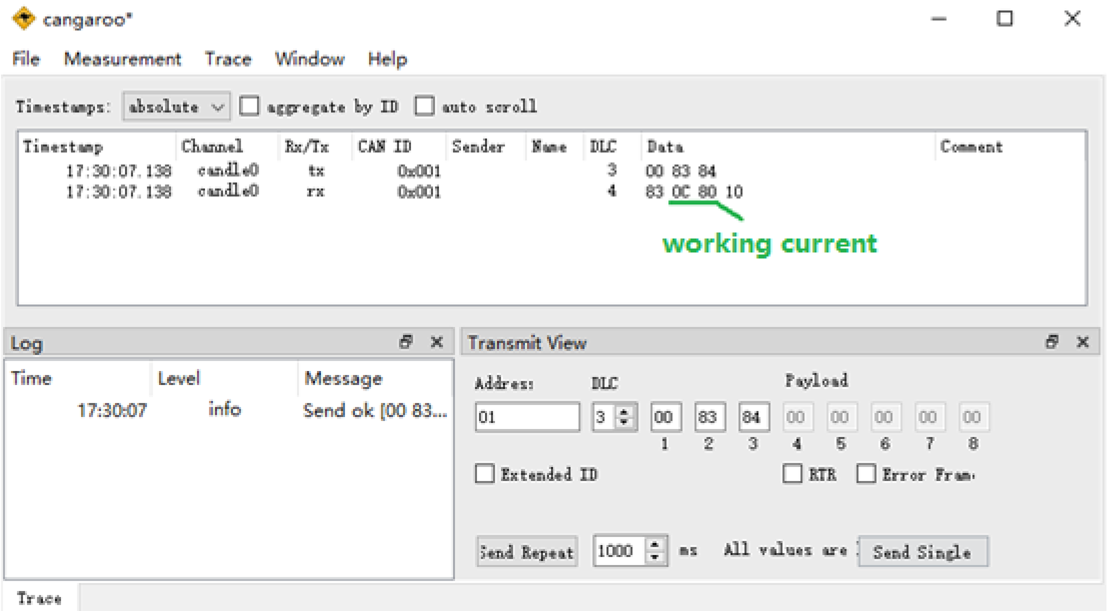
param : the system parameters.

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 ← SERVO42D/57D)						
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

1000 ms All values are

Part6. Run the motor by CAN command

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

6.1 Description the parameters of speed and acceleration

1. speed

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

When speed = 0, the motor stops rotating.

The maximum speed of the control mode is as follows:

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

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

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

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

2. acceleration

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

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

① accelerates

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

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

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

For example: acc = 236, speed = 3000

T(ms)	speed (RPM)
0	0
1	1
2	2
3	3
…	…
…	…
…	…
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:

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

6.2 Query/Enable the motor command

1. Query the motor status

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	F1	CRC(F2)

Uplink frame (PC ← SERVO42D/57D)					
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…

2. Enable the motor

Downlink frame(PC → SERVO42D/57D)					
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 ← SERVO42D/57D)					
CAN ID	…	DLC	byte1	byte2	byte3
01		3	code	data	Check
			F3	status(uint8_t)	CRC

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

6.3 Emergency stop the motor

Downlink frame(PC → SERVO42D/57D)				
CAN ID	…	DLC	byte1	byte2
01		2	F7	CRC

Uplink frame (PC ← SERVO42D/57D)					
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!

6.4 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 → SERVO42D/57D)									
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 ← SERVO42D/57D)					
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: the "Uplink frame" can be disable by Menu "CanRSP" or Command "8C".

2. Stop the motor in speed mode

Downlink frame(PC → SERVO42D/57D)									
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

3. Save/Clean the parameter in speed mode

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

state = C8 Save.

state = CA Clean.

Uplink frame（PC ← SERVO42D/57D）					
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.

6.5 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 → SERVO42D/57D)										
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 ← SERVO42D/57D)					
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: the "Uplink frame" can be disable by Menu "CanRSP" or Command "8C".

2. Stop the motor in position mode1

Downlink frame(PC → SERVO42D/57D)										
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	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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

6.6 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 → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

2. Stop the motor in position mode3

Downlink frame(PC → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

6.7 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 → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

2. Stop the motor in position mode3

Downlink frame(PC → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

6.8 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 → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

2. Stop the motor in position mode4

Downlink frame(PC → SERVO42D/57D)								
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 ← SERVO42D/57D)					
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: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

Part7. CAN command example

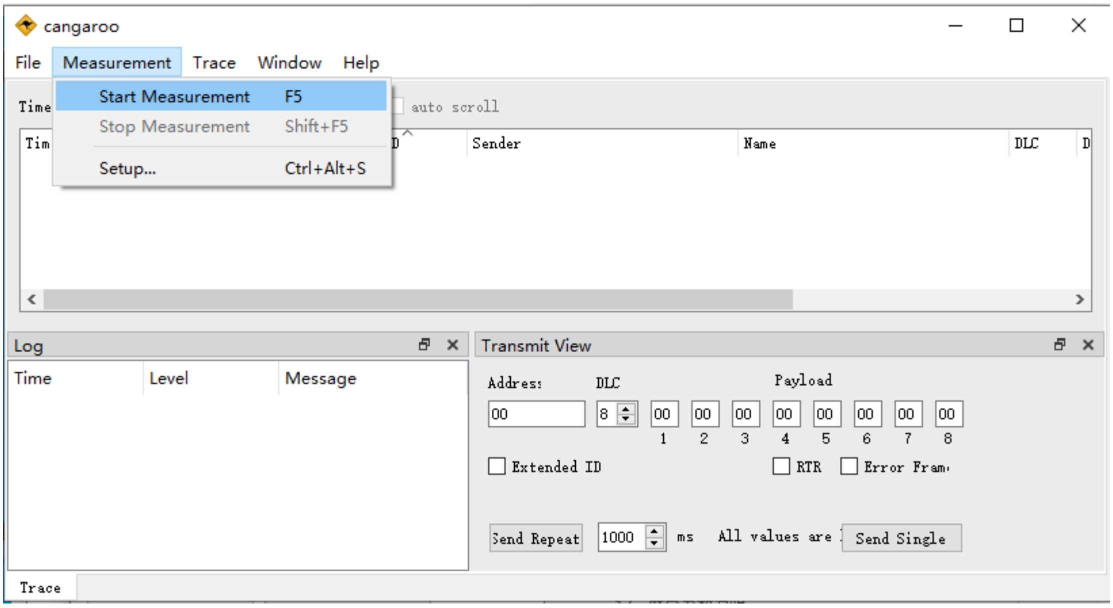
The following example uses ″cangaroo.exe″ PC software and ″MKS CANable″ USB to CAN module.

7.1 Config the SERVO42D/57D

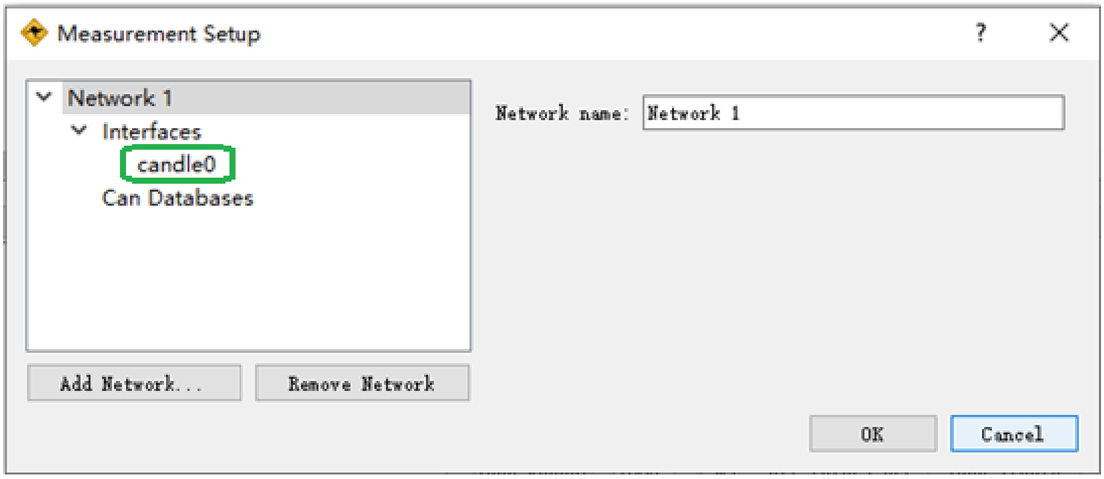
1. Menu → Mode → SR_vFOC.
2. Menu → CanRate → 500K.
3. Menu → CanID → 01.

7.2 Config the cangaroo

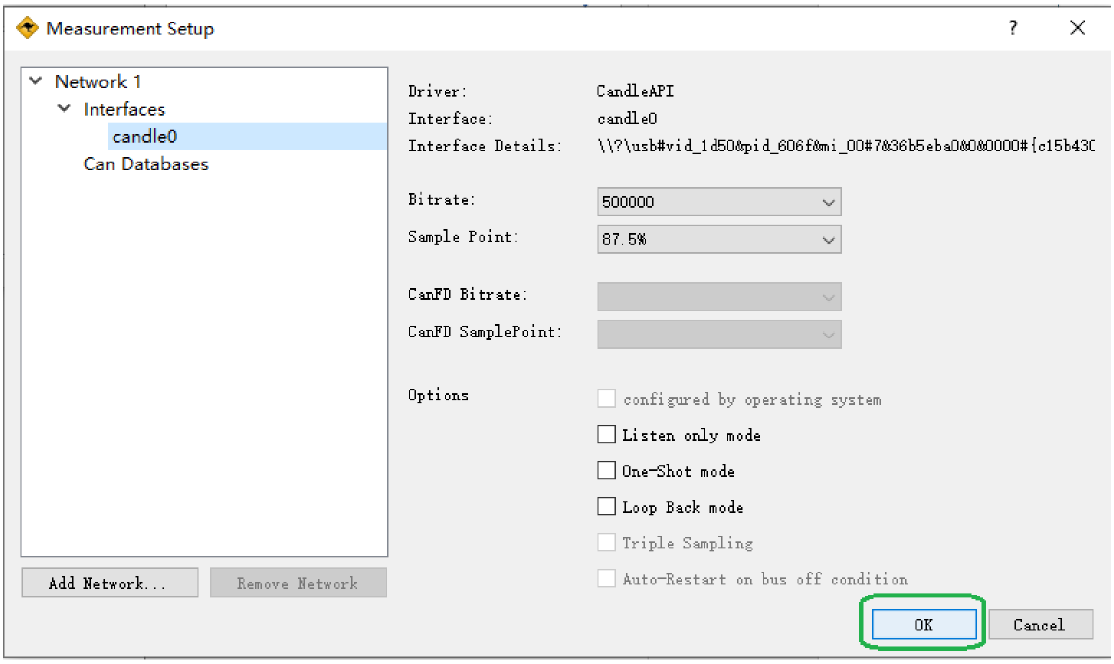
1. run the “cangaroo.exe”.
2. Select t “Measurement” -> “Start Measurement”, as show below.



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.



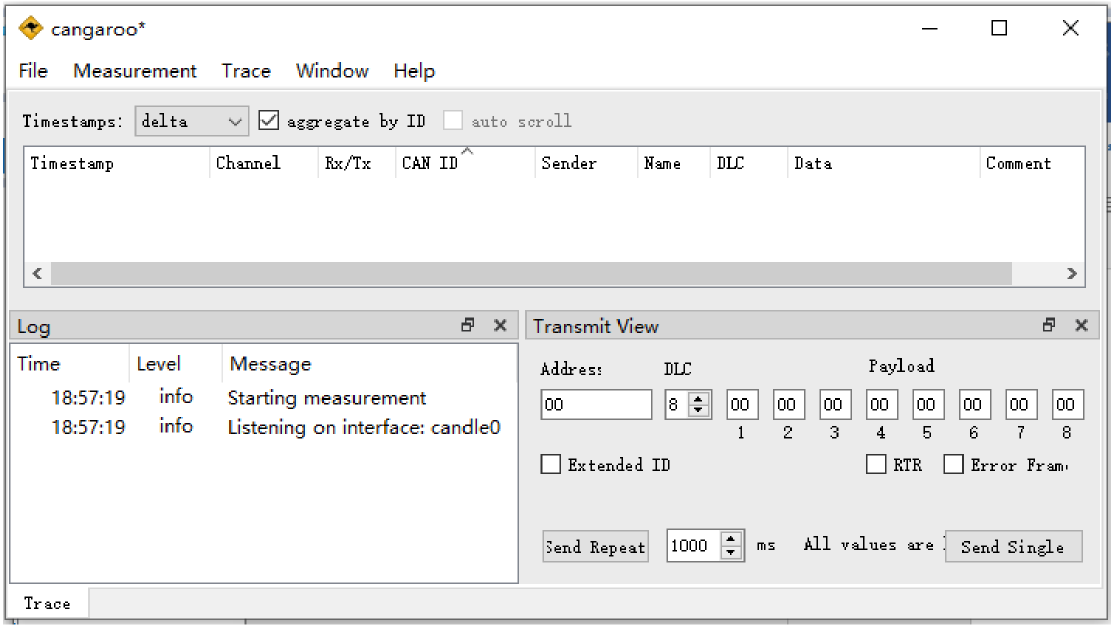
Driver: CandleAPI
Interface: candle0
Interface Details: \\?\usb#vid_1d50&pid_606f&mi_00#7&36b5eba0&0&0000#{c15b43C

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.

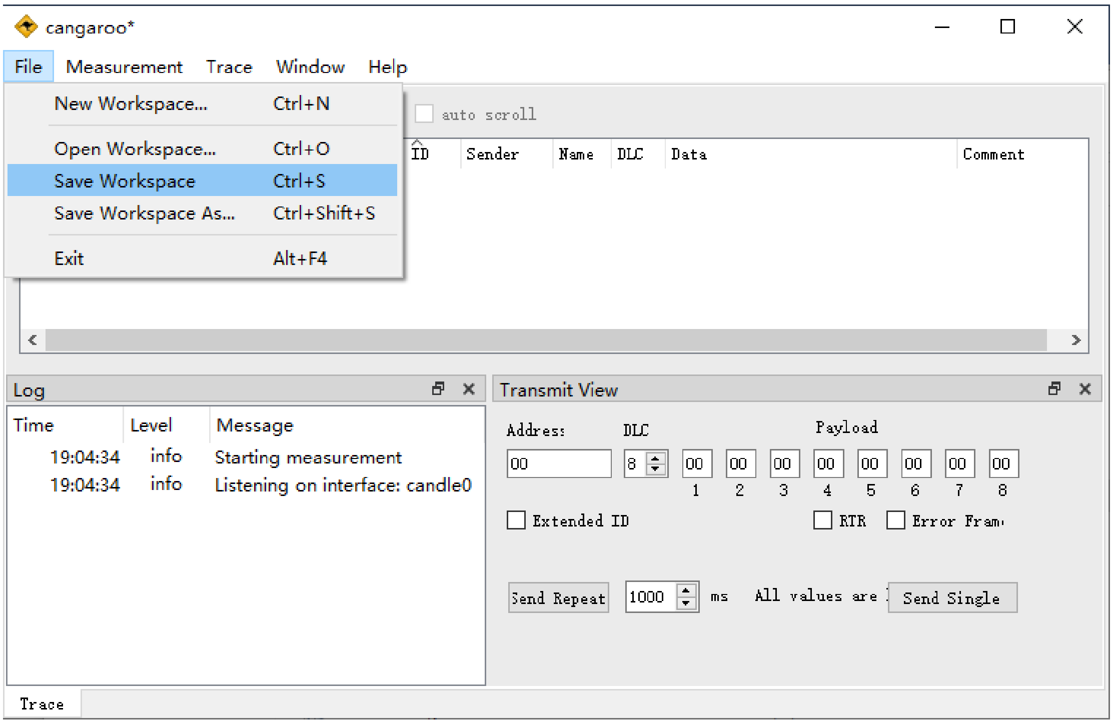


Timestamps: delta ☑ aggregate by ID □ auto scroll

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

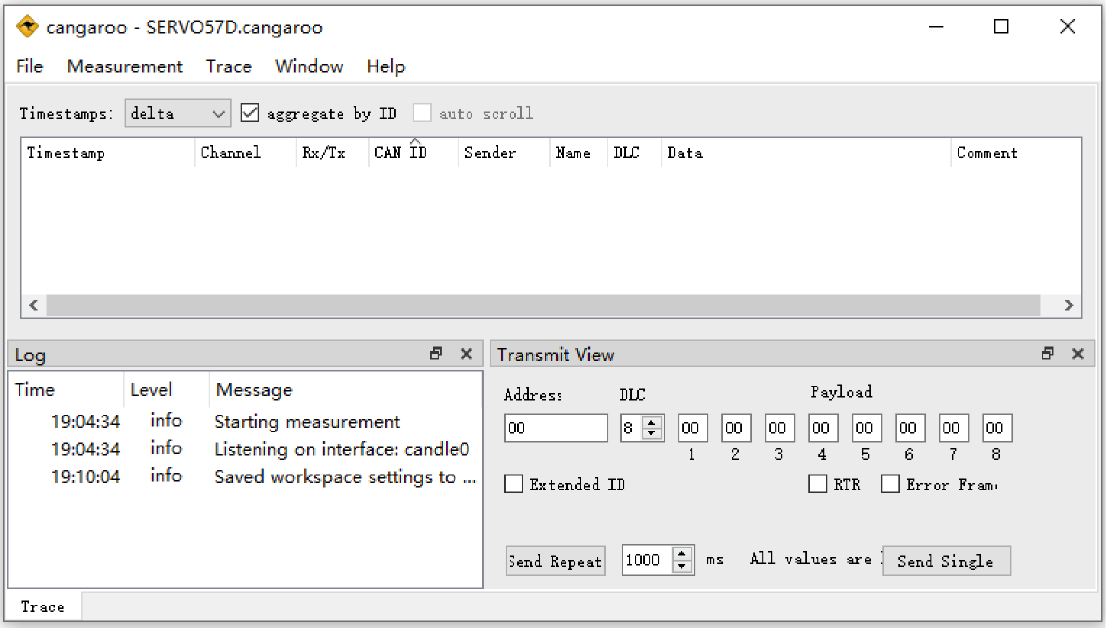
Address: 00 DLC: 8 Payload: 00 00 00 00 00 00 00 00
□ Extended ID □ RTR □ Error Fram
1000 ms

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

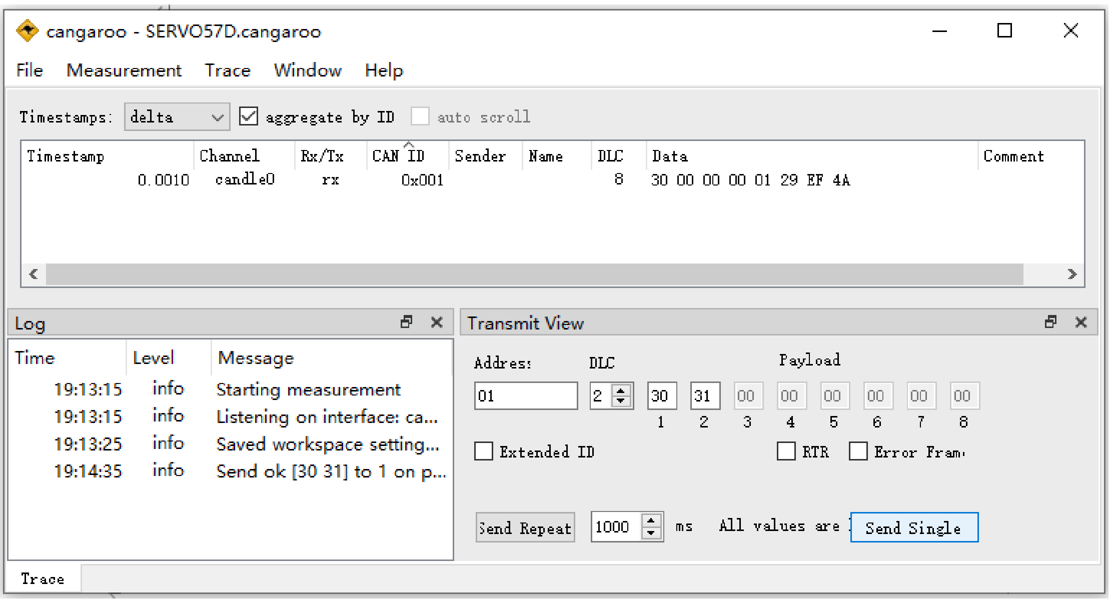
7. After the save is completed, as shown below.



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 ...

7.3 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
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: 30 31 00 00 00 00 00 00 (1 2 3 4 5 6 7 8)

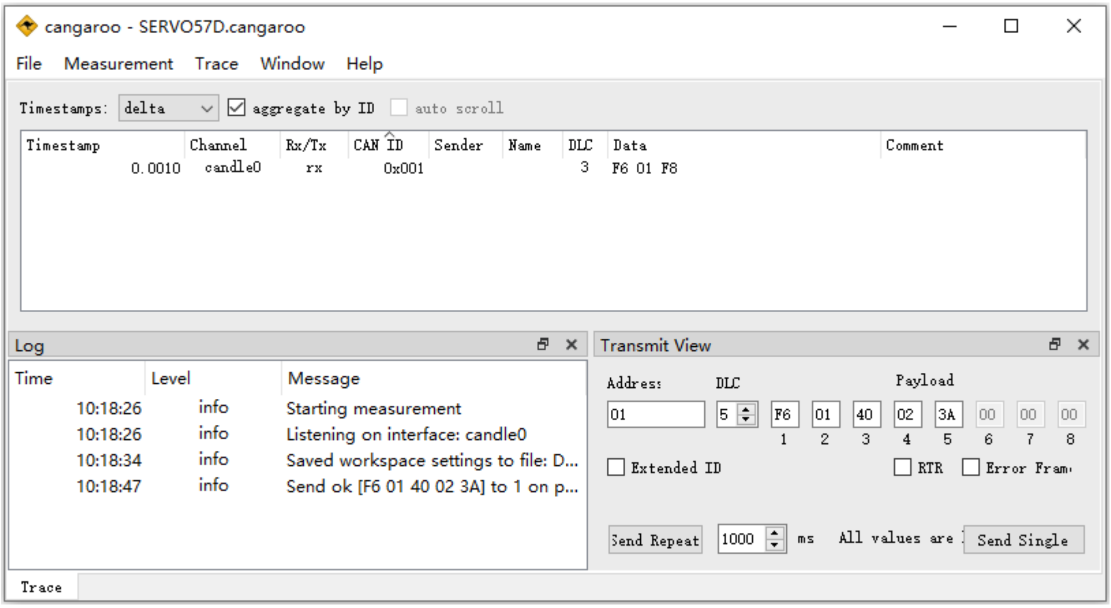
□ Extended ID □ RTR □ Error Fram

1000 ms All values are

7.4 Run the motor in speed mode

Note : Please configure the working mode to “SR_vFOC”.
Menu-> Mode -> SR_vFOC

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
File Measurement Trace Window Help
Log
Transmit View
Trace

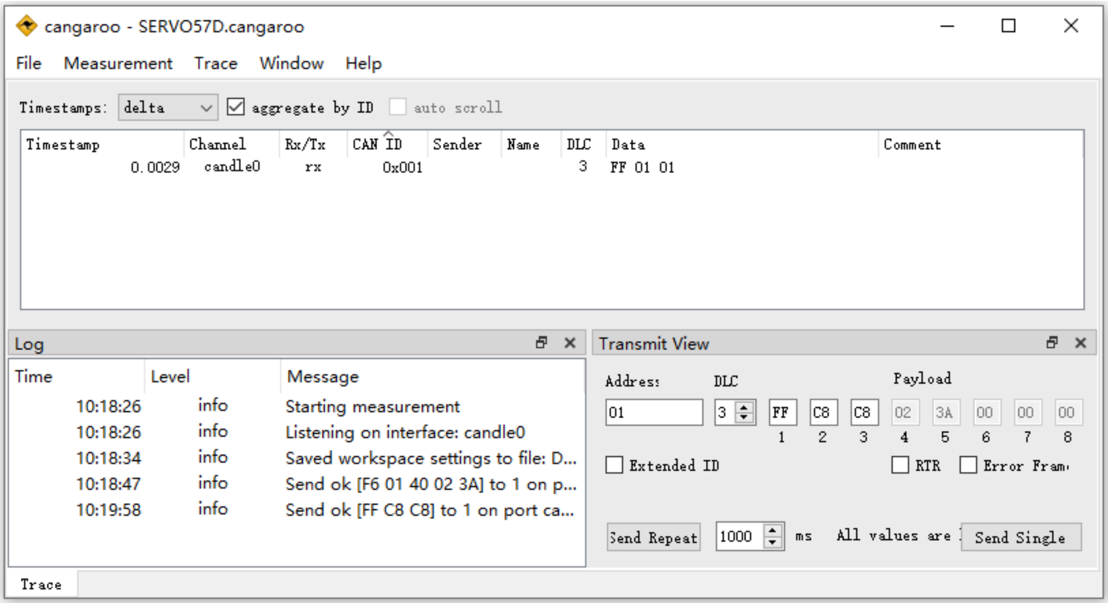
Timestamps: delta ☑ aggregate by ID □ auto scroll

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

Address: 01 DLC: 5 Payload: F6 01 40 02 3A 00 00 00 (1 2 3 4 5 6 7 8)
□ Extended ID □ RTR □ Error Fram
Send Repeat 1000 ms All values are Send Single

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


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

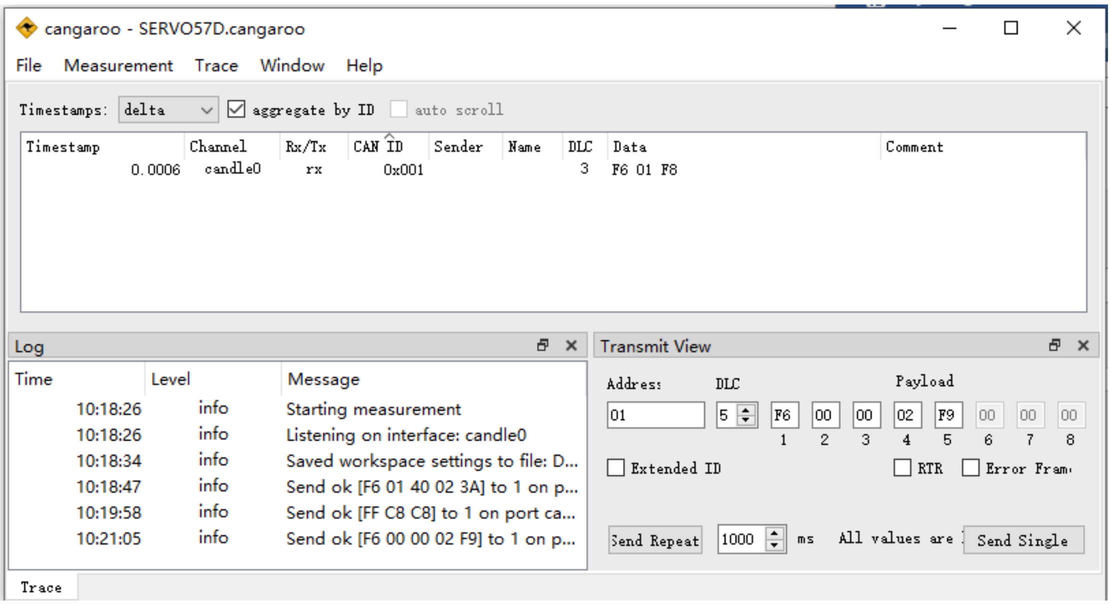
Timestamps: delta ☑ aggregate by ID □ auto scroll

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

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...

Address: 01 DLC: 3 Payload: FF C8 C8 02 3A 00 00 00 (1 2 3 4 5 6 7 8)
□ Extended ID □ RTR □ Error Fram
Send Repeat 1000 ms All values are Send Single

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
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 All values are

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

7.5 Run the motor in position mode1

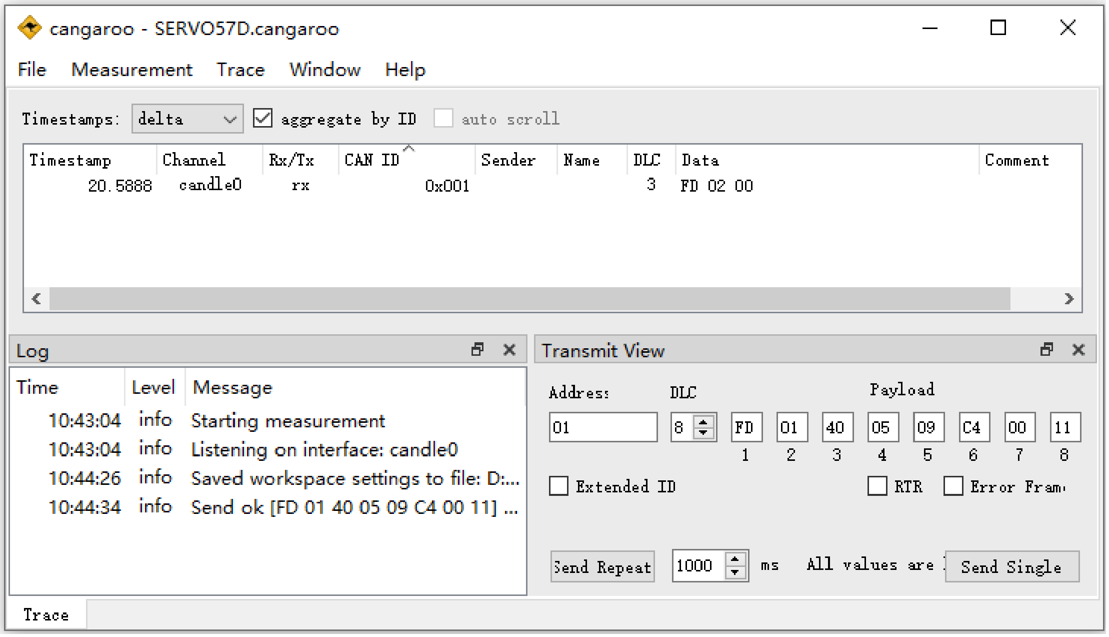
Note : Please configure the working mode to “SR_vFOC”.

Menu-> Mode -> SR_vFOC

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;



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

Log

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] ...

Transmit View

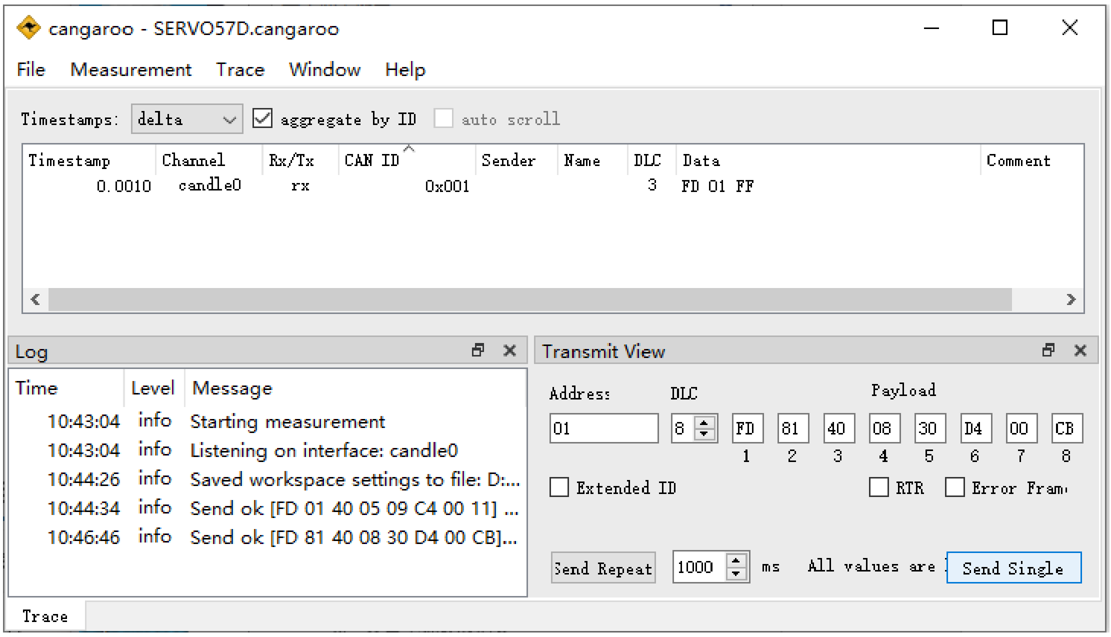
Address: 01 DLC: 8

Payload: FD 01 40 05 09 C4 00 11

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.



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

Log

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]...

Transmit View

Address: 01 DLC: 8

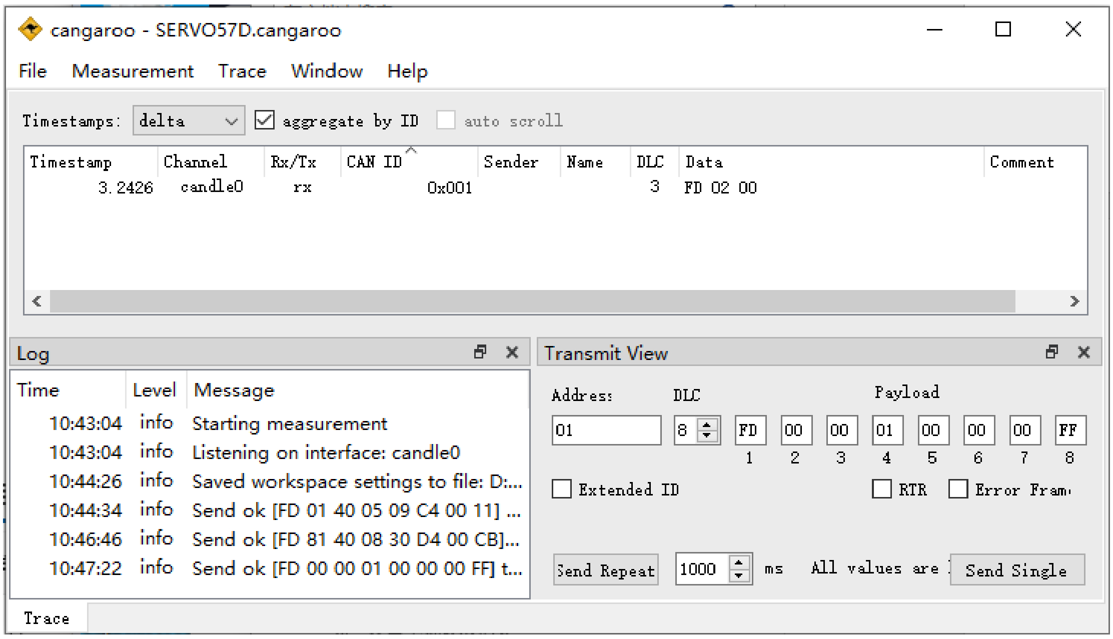
Payload: FD 81 40 08 30 D4 00 CB

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
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
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 Frame

1000 ms

Part8. FAQ

8.1 NOTE

1. Power input voltage is 12V-24V.
2. Don’t hot plug motor cable and data cable.
3. When the motor is calibrating, do not carry a load.
4. After installed the motor, or changed the motor wiring sequence, you need to re-calibrate the motor again.
5. The default work mode is CR_vFOC(EN/STP/DIR interface).
6. Press the "Next" key first, then power on, it can quickly restore the default parameters.
7. If “Phase Line Error!” is displayed before calibration:
   b) Check the motor connection line sequence;
   c) Check the power supply voltage and output power (24V/1A, 12V/2A);;
   d) If the power supply is connected to the motherboard through the MKS APT module, try to connect the MKS APT module to ports such as X, Y, Z, E, etc., and then restart again.
   e) Do not use the MKS APT module for power supply before calibration, and the power supply is directly connected to V+ and Gnd.

8.2 FAQ

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

Part9. Schematic

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

Part10. contact us

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