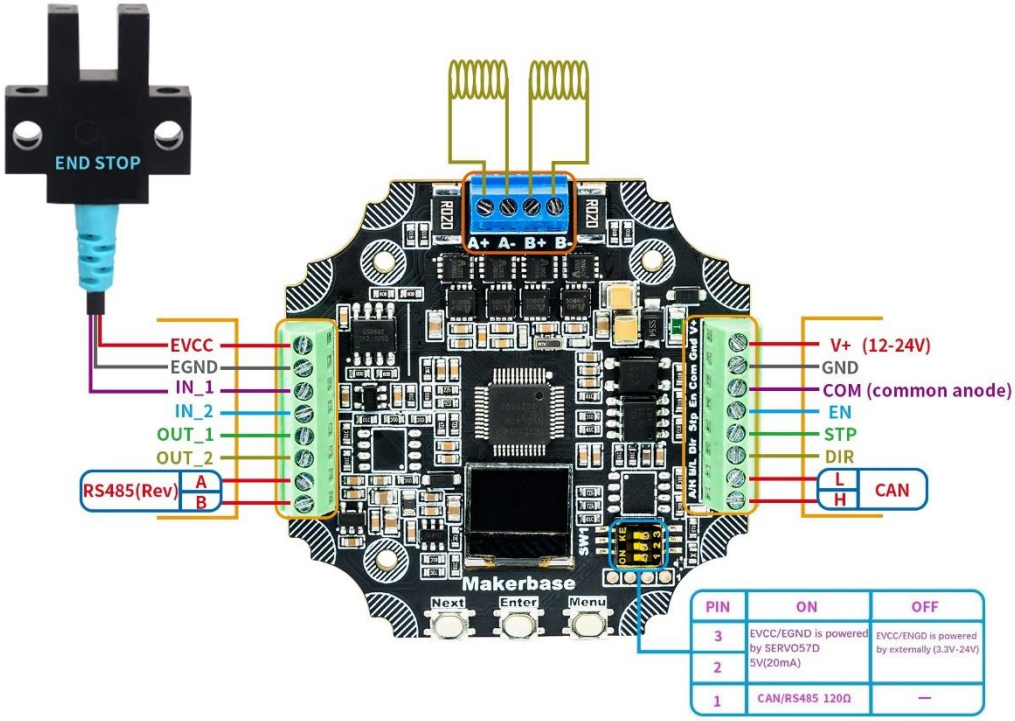
MKS SERVO42D/57D_CAN V1.0.0

USER MANUAL

MKS SERVO42D/57D Manual Release			
Manual	discription	firmware	date
V1.0	First release.	V1.0	03/2023

Part1. Feature

1.1 Interface



1.2 Key Operation

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

1. 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.

2. 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.

1.4 Work mode

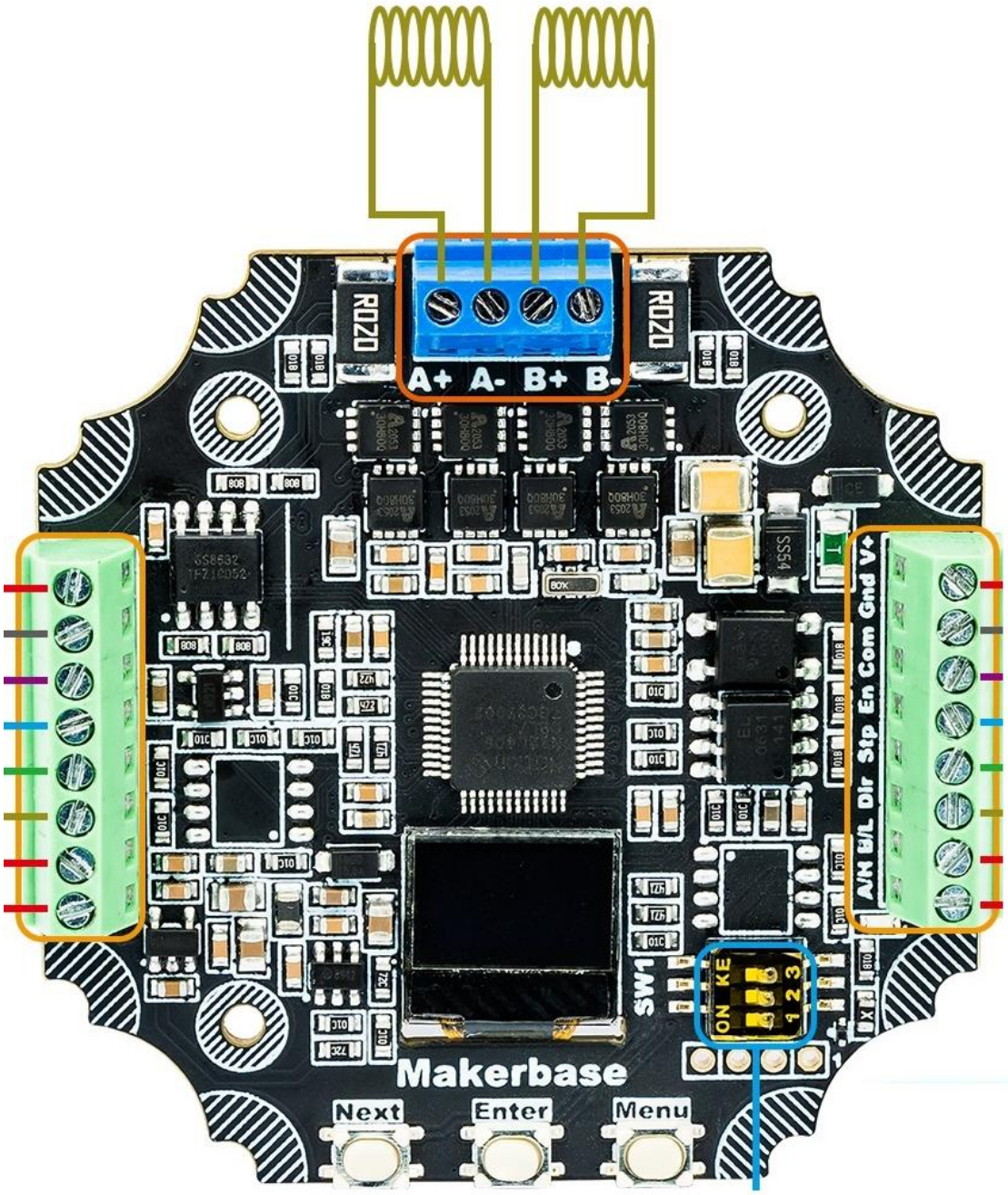
Work mode	Signal input	Encoder	Current
CR_OPEN	EN,STP,DIR	NO	Fixed
CR_CLOSE	EN,STP,DIR	YES	Fixed
CR_vFOC	EN,STP,DIR	YES	adaptive
CR_CAN	CAN serial	YES	adaptive

Note: The default work mode is CR_vFOC

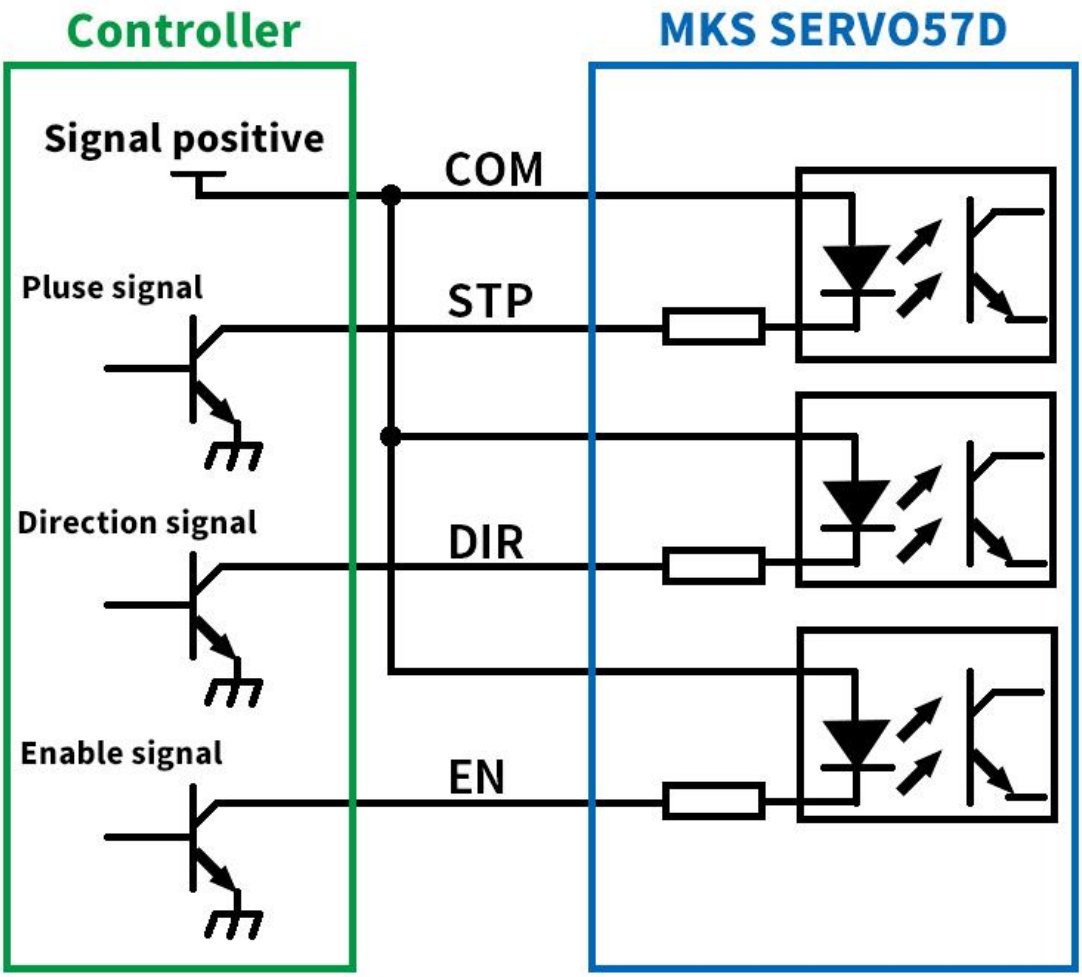
Part2. Wire

2.1 Motor wire

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



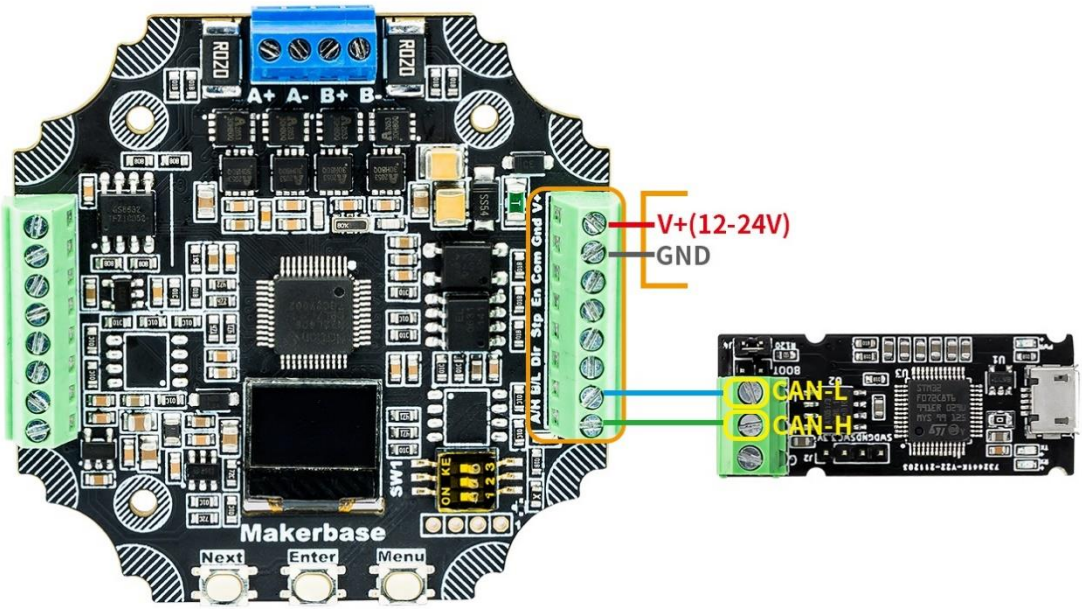
2.2 Pulse interface wire



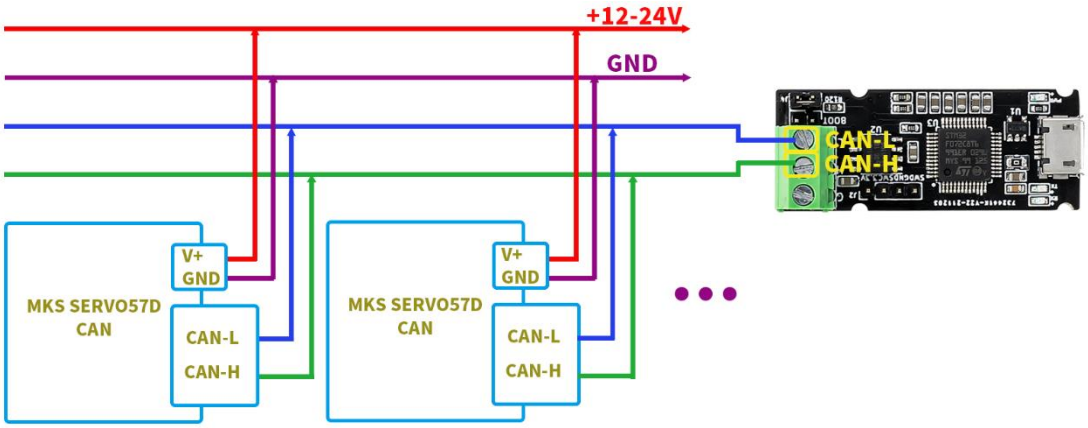
Note: if the (STP/DIR/EN)signal high level is 3.3V,the COM must be 3.3V
if the (STP/DIR/EN)signal high level is 5.0V,the COM must be 5.0V
…
and so on.

2.3 CAN wire

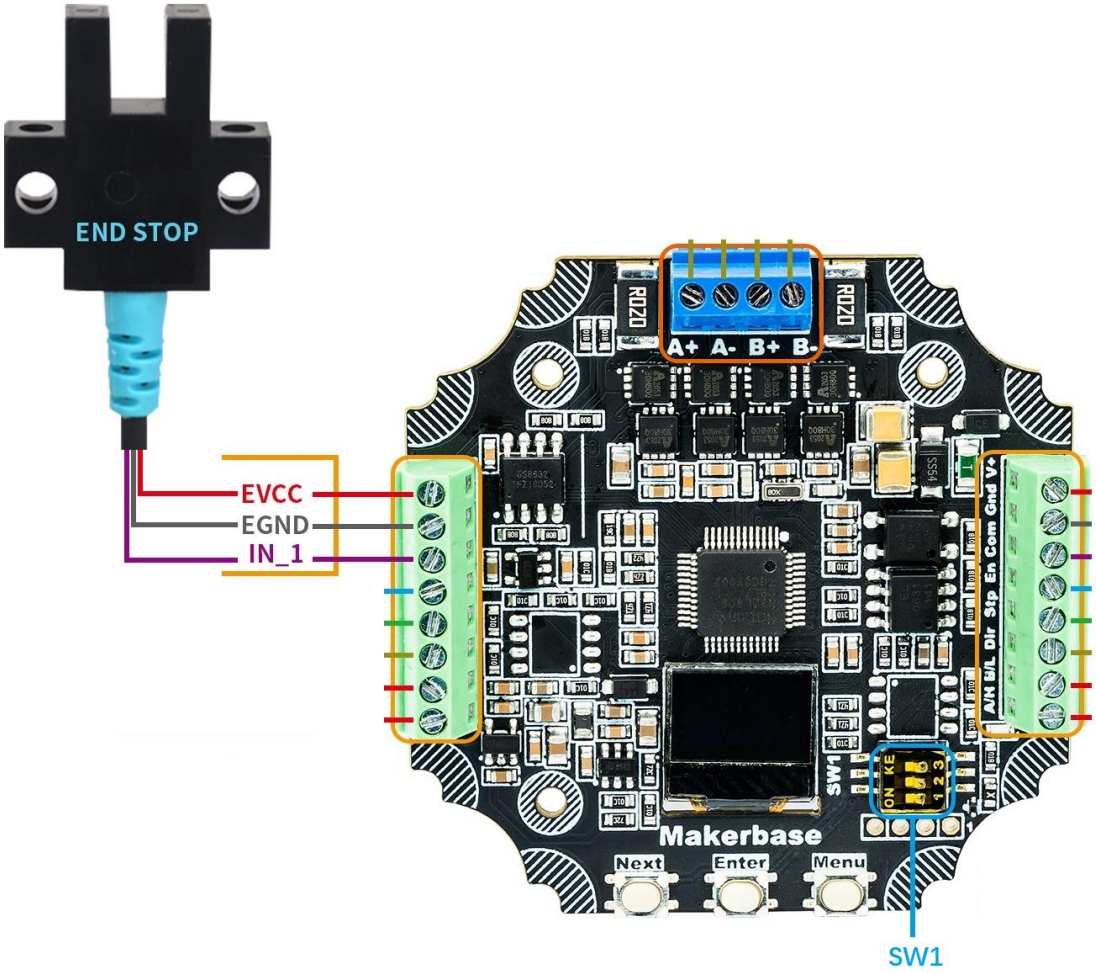
1. Single-slave



2. Multiple-slave



2.4 End stop wire



SW1		
PIN	ON	OFF
3	EVCC/EGND is powered by SERVO57D 5V(20mA)	EVCC/ENGD 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.

Part3. Menu description

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

2. **Mode** : Work mode selection.
   CR_OPEN : Open mode, the motor run without encoder
   CR_CLOSE : Close mode, the motor run with encoder.
   CR_vFOC : FOC mode, pulse(En,Stp,Dir) interface.
   CR_CAN : FOC mode, CAN interface.
   (Default: CR_vFOC)

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

Work mode	Current (Ma)
CR_OPEN	The working current is fixed, and the working current is Ma.
CR_CLOSE	
CR_vFOC	The working current is adaptive, the maximum current is Ma.
CR_CAN	

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

4. **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.

5. **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)

6. **Dir** : Set the positive direction of motor rotation.
   CW : Clockwise rotation is positive
   CCW : Counterclockwise rotation is positive
   (Default: CW)

7. **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: you can release the protection status by pressing the Enter button or the serial command.

8. **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.

9. **CanRate** : Set the bit rate of CAN interface.
125K, 250K, 500K。
(Default: 500K)

10. **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.

11. **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”.

12. **O_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 O_Dir menu).
NearMode : go back to zero ith minimum angle.
(Default: Disable)

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

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

15. **O_Dir** : Set the direction of go back to zero point.
    CW : Clockwise.
    CCW : Counterclockwise.
    (Default: CW)

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

17. **HmDir** : Set the direction of go home。
    CW : Clockwise rotation is positive
    CCW : Counterclockwise rotation is positive
    (Default: CW)

18. **HmSpeed** : Set the speed of go home。
    0 : slowest
    ...
    4 : fastest

19. **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.

20. **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.

21. **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

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 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.

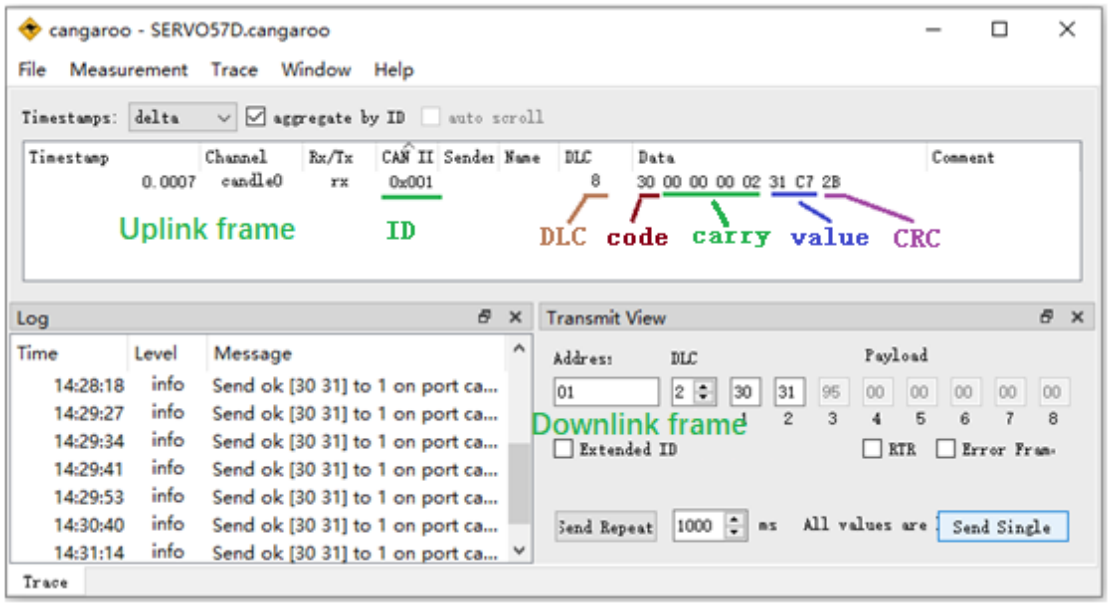
When Value is less than 0, carry -=1.

For example:

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

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

The Cangaroo example is as follows:



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 clockwise,the value(+0x4000) is 0x7FF0.

If the current value is 0x3FF0, After one turn CCW,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 CW,the speed > 0 (RPM)
if it run CCW,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 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…byte3	byte4
01	…	4	code	data	Check
			39	error(int16_t)	CRC

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

for example, when the angle error is 1° , the return error is 65536/360= 182.444, and so on.

6. Command6 : 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

7. Command7 : 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.

8. Command8 : 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.

9. Command9 : 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 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 (0~3)	CRC

mode = 0 CR_OPEN
mode = 1 CR_CLOSE
mode = 2 CR_vFOC
mode = 3 CR_CAN

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 current

（Same as the "Ma" option on screen）

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

SERVO42D: working current = ma× 200 (0≤ma≤15)
SERVO57D: working current = ma× 400 (0≤ma≤13)

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 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.

5. 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.

6. 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.

7. 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: you can release the protection status by pressing the Enter button or the serial command.

8. 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.

9. 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

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.

10. Set the CAN ID

（Same as the "CanID" option on screen）

下行报文(上位机 → 驱动板)						
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.

11. Set the slave respond

（Same as the "CanRSP" option on screen）

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

enable = 01 enabled respond
enable = 00 disabled respond
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.

5.3 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	byte5
01	...	5	code	level	dir	speed	Check
			90	homeTrig	homeDir	homeSpeed	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 : slowest

...

4 : fastest

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.

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: the working mode should be “CR_CAN”

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.

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.

5.4 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.

Part6. Run the motor by CAN command

Note: This chapter needs to set the working mode to “CR_CAN”.

6.1 Description the parameters of speed and acceleration

1.speed

The speed parameter ranges from 0 to 1600. The larger the value, the faster the motor rotates.

When speed = 0, the motor stops rotating.

The calculation relationship between the speed parameter and the motor Rotating speed (Vrpm) is as follows:

Vrpm = (speed × 6000)/(Mstep × 200) RPM

for example:

speed = 1600, Mstep=16

Vrpm = (1600 × 6000)/(16 × 200) = 3000 RPM

Part of the speed - Vrpm as follows:

speed	Vrpm (RPM)	
	Mstep=16	Mstep=32
1	1.875	0.9375
40	75	37.5
80	150	75
160	300	150
320	600	300
400	750	375
640	1200	600
1000	1875	937.5
1280	2400	1200
1600	3000	1500

Note: the Max Vrpm is 3000RPM

2. acceleration

The value of the acceleration(acc) ranges from 0 to 32. The larger the value, the faster the motor accelerates/decelerates.

When 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 = 10 (mS)

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

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

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 = 10 (mS)

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

$V_{t2} = V_{t1} - acc$ (V_{t2} >= 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

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 Speed mode command

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

1. Run the motor in speed mode

<table>
<tr><td colspan="10">Downlink frame(PC → SERVO42D/57D)</td></tr>
<tr><td>CAN ID</td><td rowspan="3">...</td><td>DLC</td><td>byte1</td><td colspan="4">byte 2</td><td>byte 3</td><td>byte 4</td><td>byte 5</td></tr>
<tr><td rowspan="2">01</td><td rowspan="2">5</td><td>code</td><td>dir</td><td>Rev</td><td colspan="3">speed</td><td>acc</td><td>Check</td></tr>
<tr><td rowspan="2">F6</td><td>b7</td><td>b6-b4</td><td>b3-b0</td><td>b7-b0</td><td rowspan="2">acc</td><td rowspan="2">CRC</td></tr>
<tr><td>dir</td><td>——</td><td colspan="3">speed</td></tr>
</table>

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 (CW/CCW)

speed: the speed, the value range is 0-1600

acc: the acceleration, the value range is 0-32

for example:

Send “01 F6 01 40 02 3A”,

the motor rotates forward at acc=2, speed=0x140

Send “01 F6 81 40 02 BA”,

the motor reverses at acc=2, speed=0x140

Note: When speed=0x140, motor speed=600RPM

<table>
<tr><td colspan="6">Uplink frame (PC ← SERVO42D/57D)</td></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>F6</td><td>status(uint8_t)</td><td>CRC</td></tr>
</table>

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 goog 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/Clear 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.

6.4 Position mode1: relative motion by pulses

In the position control mode1, the motor can be run to the specified position at 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 (CW/CCW)

speed: the speed, the value range is 0-1600

acc: the acceleration, the value range is 0-32

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=0x140 (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=0x140 (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.

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 goog 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.

Note: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

6.5 Position mode2: relative motion by axis

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

Note1: the axis is the encoder value(addition).

Note2: In this mode ,the axis err about ±15.

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	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 0-32

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.

Note: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

2. Stop 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	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 goog 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.

Note: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

6.6 Position mode3: absolute motion by axis

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

Note1: the axis is the encoder value(addition).

Note2: In this mode, the axis err about ±15.

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	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 0-32

absAxis: the relative 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 10

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.

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
			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.

Note: the “Uplink frame” can be disable by Menu “CanRSP” or Command “8C”.

Part7. CAN command example

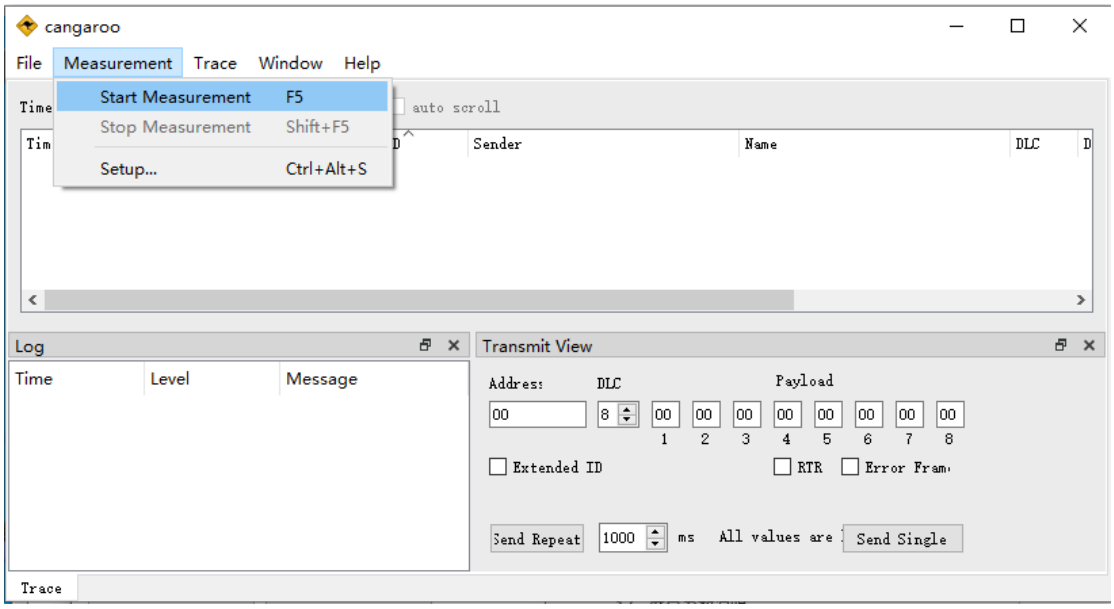
The following example uses "cangaroo.exe" PC software and "MKS CANable" USB to CAN module.

7.1 Config the SERVO42D/57D

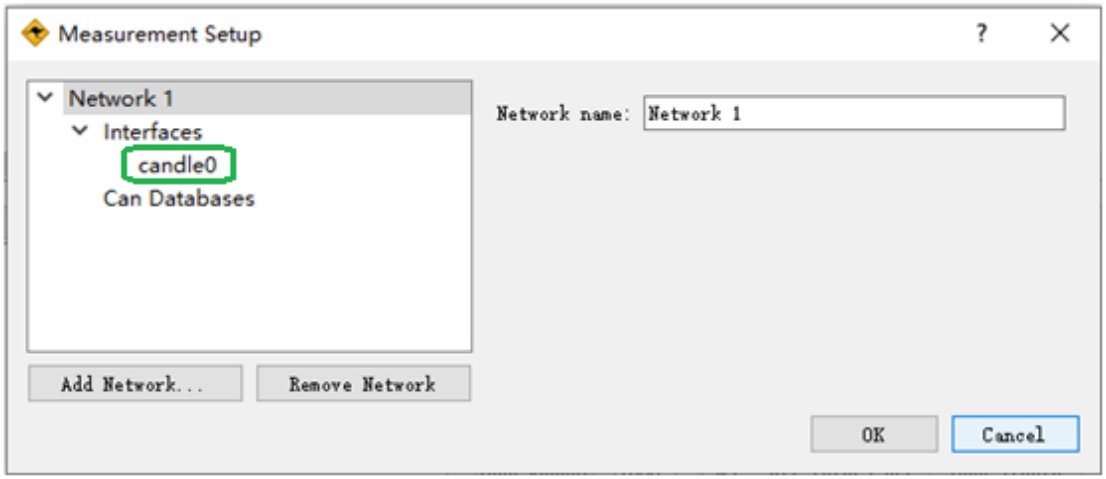
1. Menu → Mode → CR_CAN.
2. Menu → CanRate → 500K.
3. Menu → CanID → 01.

7.2 Config the cangaroo

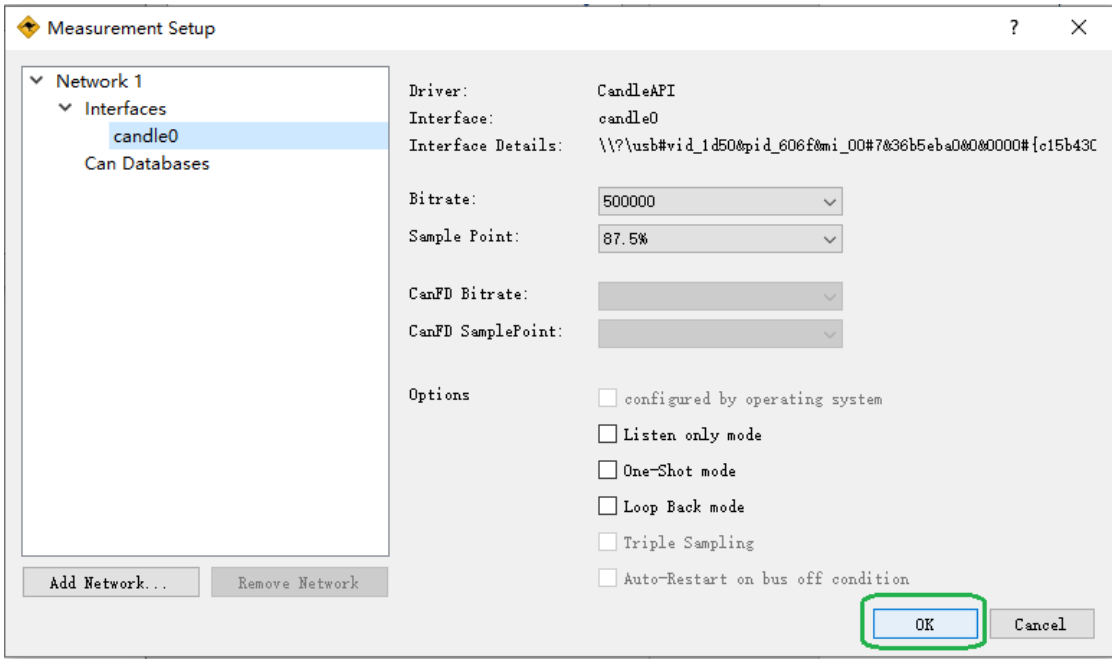
1. run the "cangaroo.exe".
2. Select t "Measurement" -> "Start Measurement",as show below.



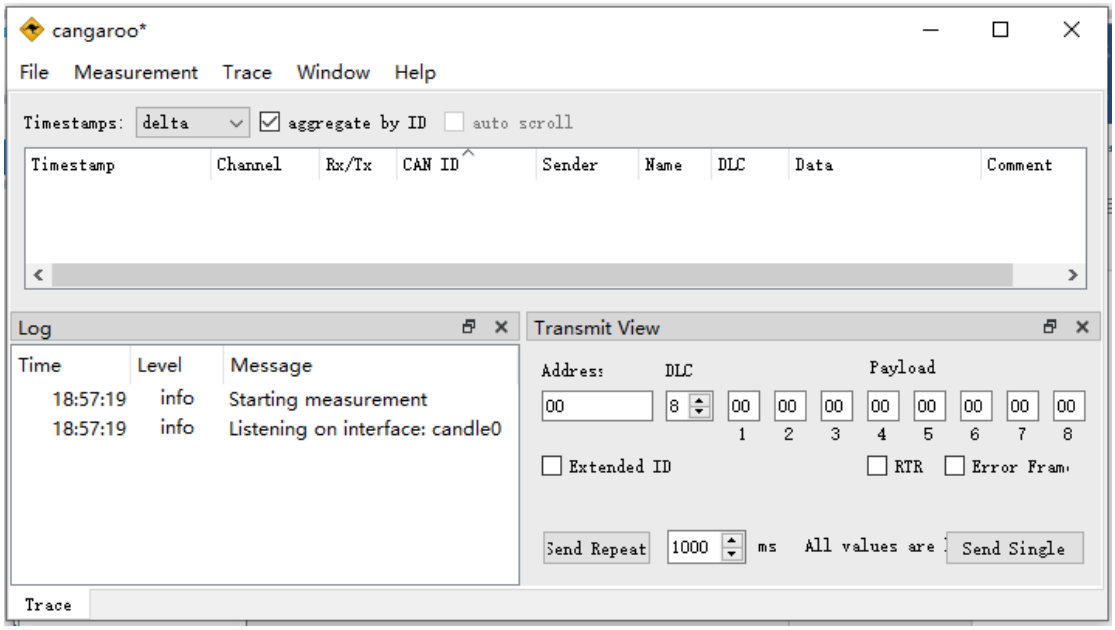
3. In the pop-up "Measurement Setup" window, click "candle0", as shown below.



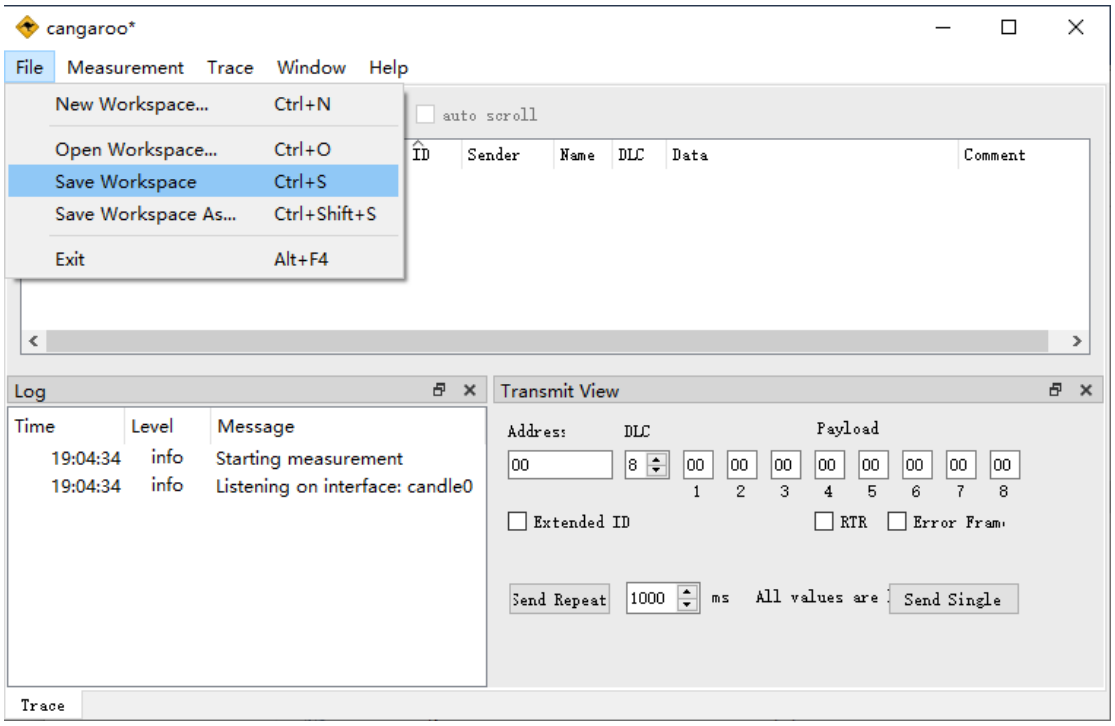
4. Use the default parameters without any modification, click "ok", as shown below.



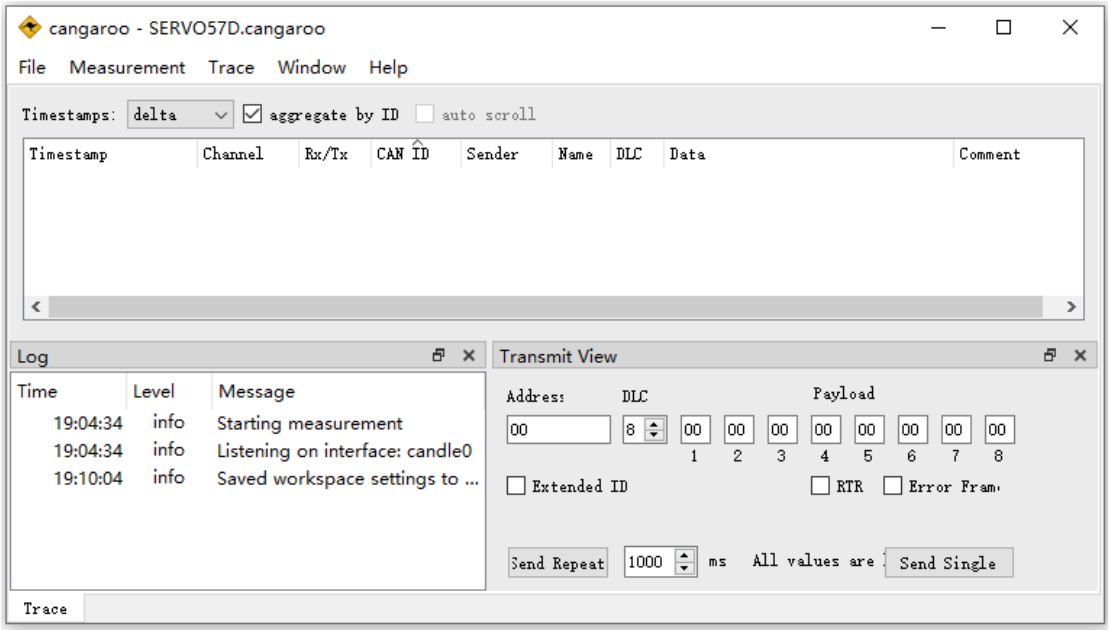
5. The configuration is complete, as shown below.



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

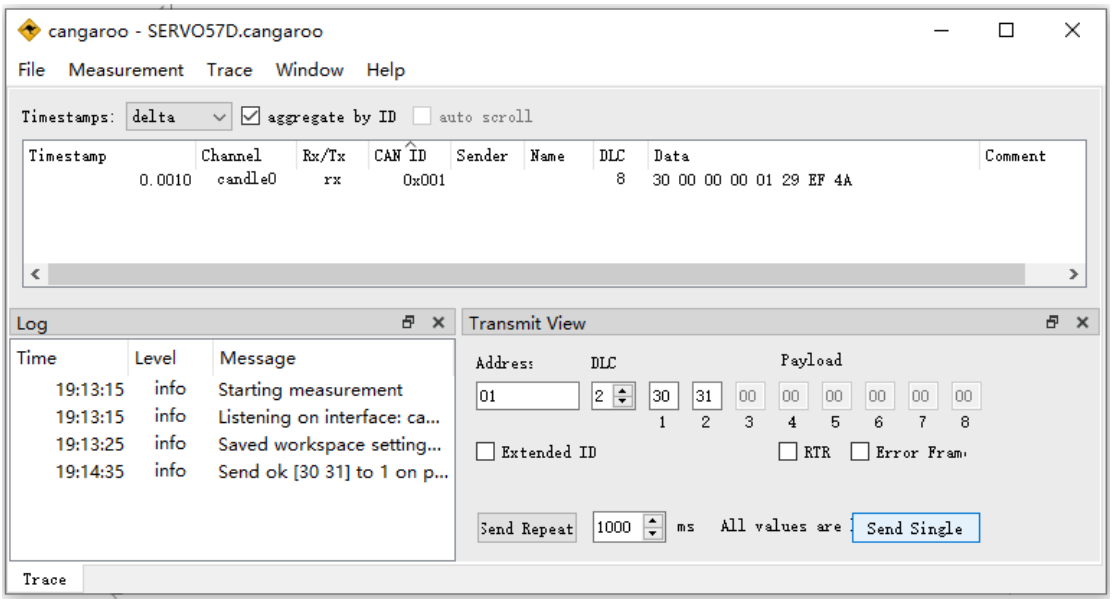


7. After the save is completed, as shown below.



7.3 Read the encoder value

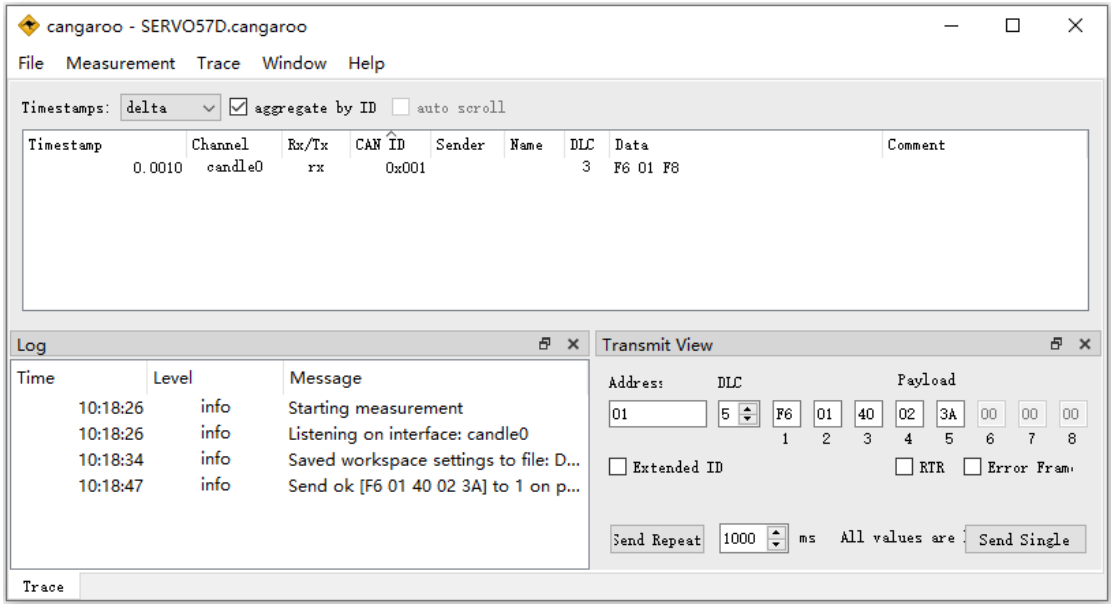
send “01 30 31”
return “01 30 00 00 00 01 29 EF 4A”



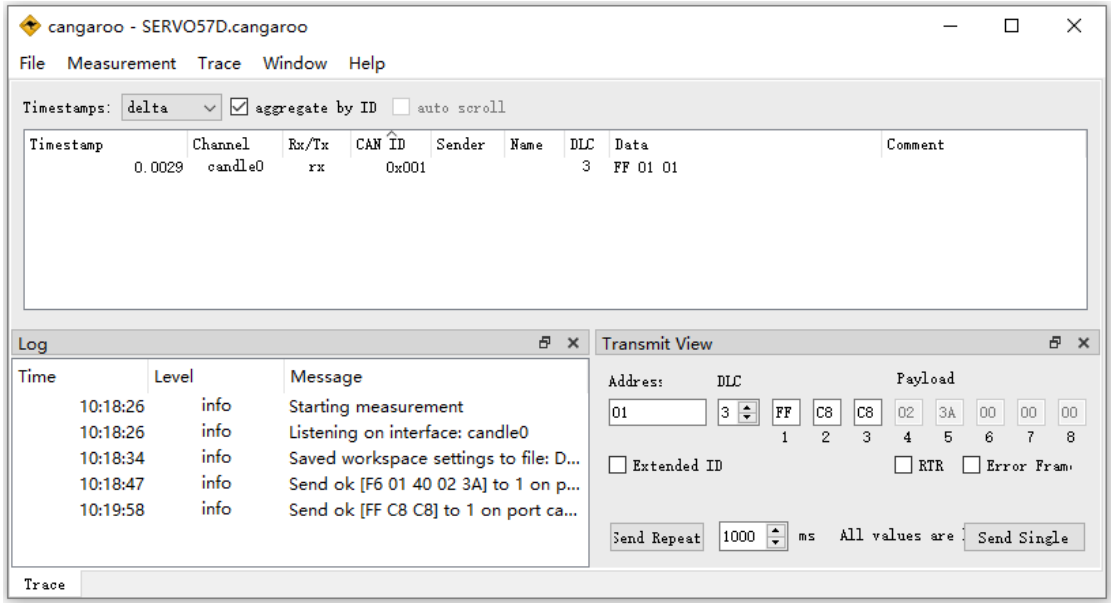
7.4 Run the motor in speed mode

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

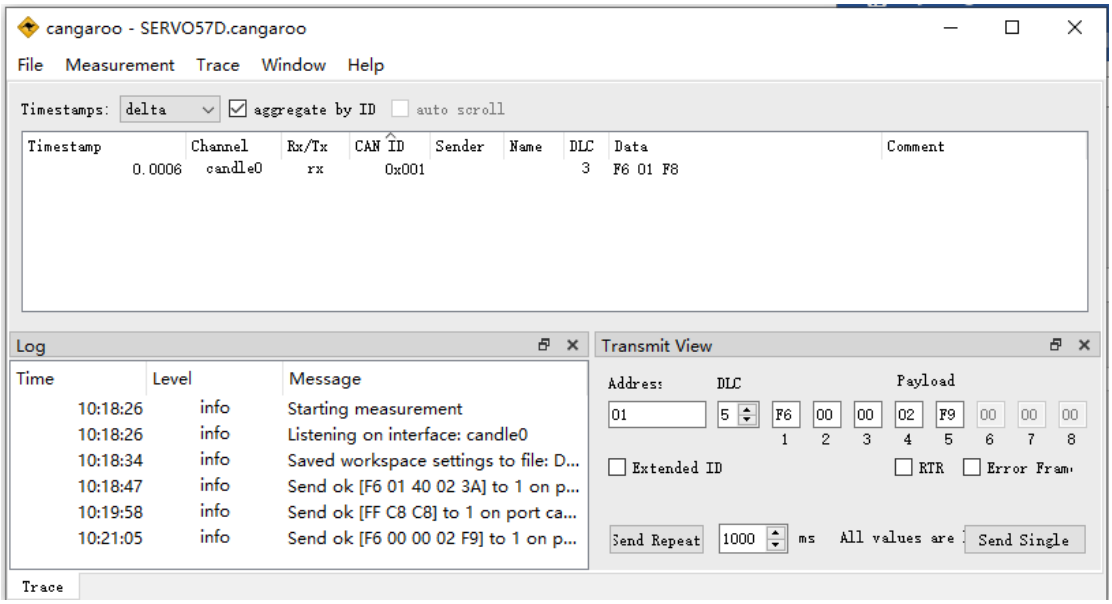
1. Send 01 F6 01 40 02 3A, the motor will rotate at ″speed = 0x140, acc=2″;
   Return 01 F6 01 F8, the motor run in speed mode successful;



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



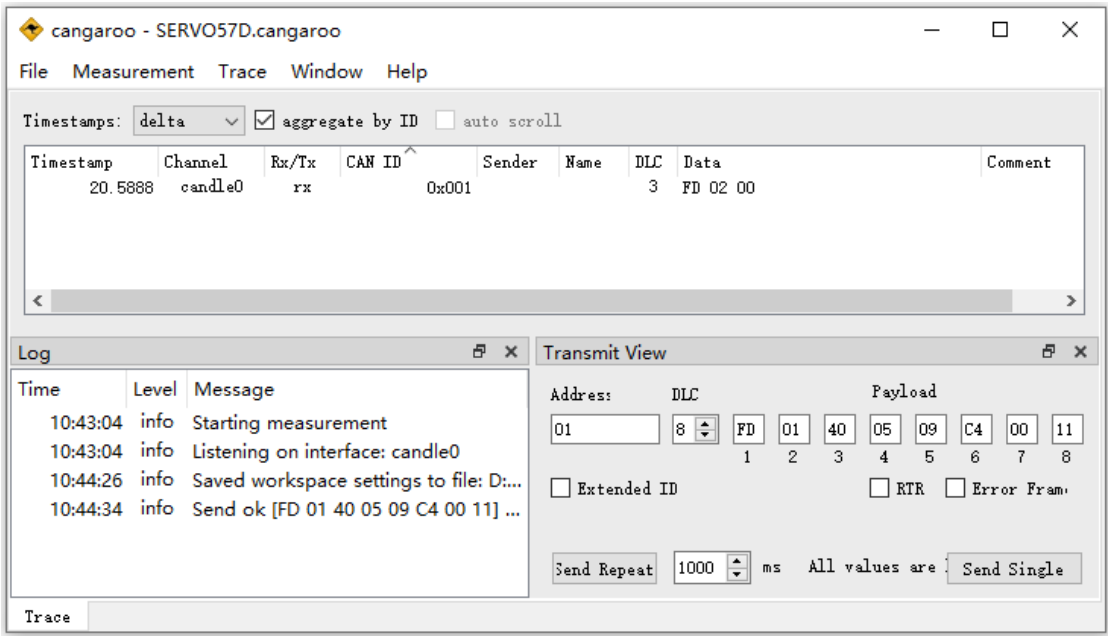
3. Send 01 F6 00 00 02 F9 to stop the motor;
Return 01 F6 01 F8, the motor stops successfully;



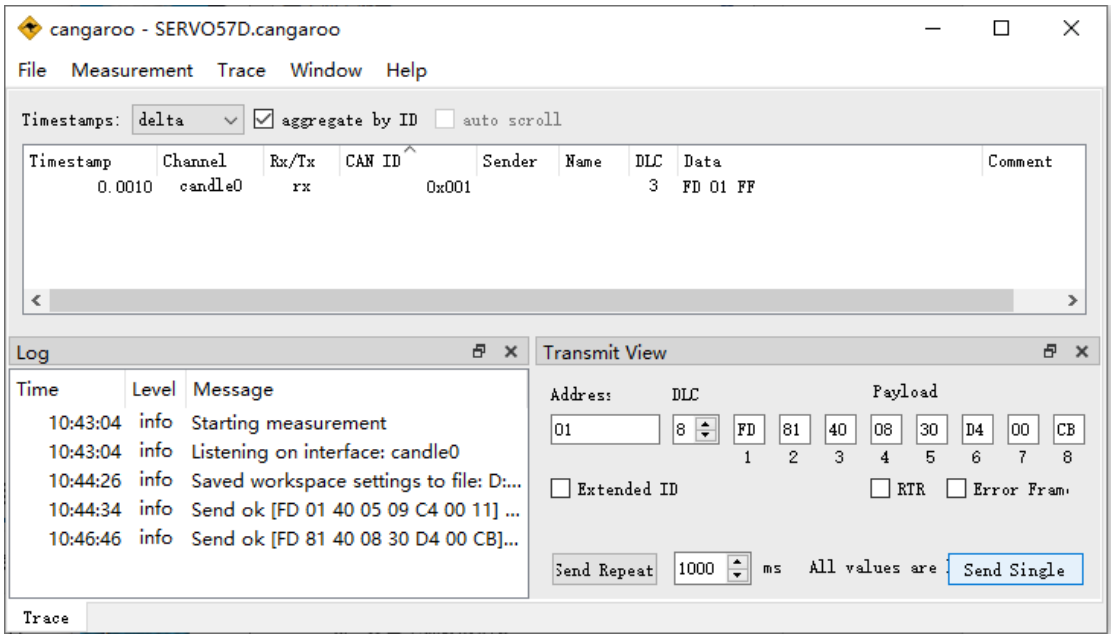
After power-on again, the motor will run according to the save speed mode parameters.

7.5 Run the motor in position model

Note : Please configure the working mode to “CR_CAN”.
Menu-> Mode -> CR_CAN
1. Send 01 FD 01 40 05 09 C4 00 11, the motor will rotate forward 200 circles (16 subdivisions) with "speed = 0x140, acc = 5";
Return 01 FD 01 FF, the motor starts to run;
Return 01 FD 02 00, the motor is run completed;



2. Send 01 FD 81 40 08 30 D4 00 CB, the motor to reverse 1000 circles with "speed = 0x140, acc = 8" (16 subdivisions);
Return 01 FD 01 FF, the motor starts to run.
Now, stop the motor by step 3.

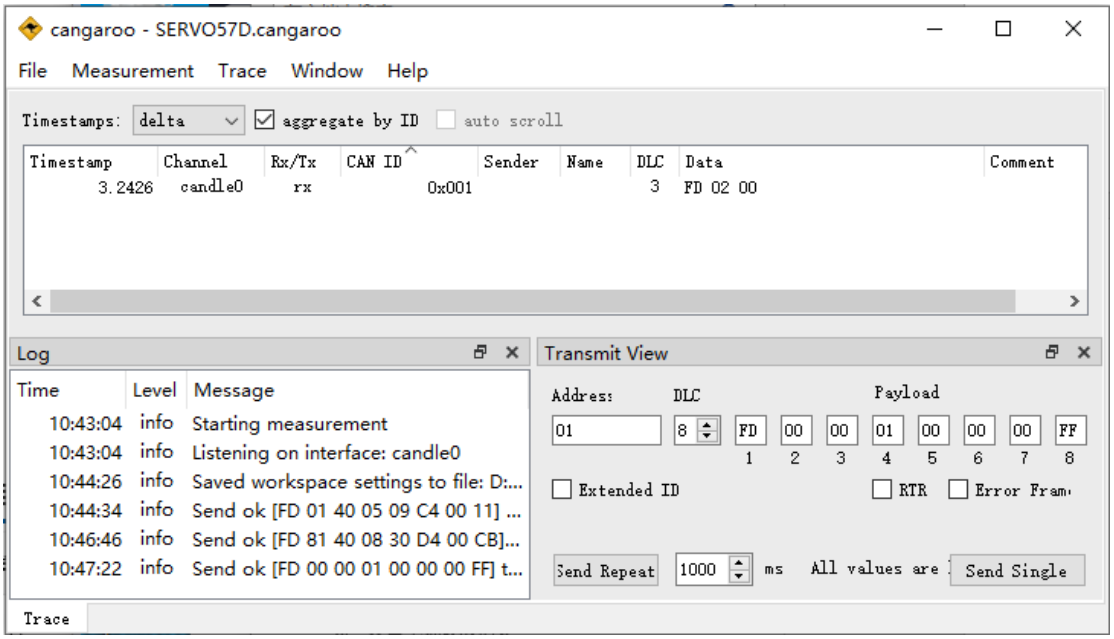


3. While the motor is running…

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;



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.

Part9. Schematic

Please download 《MKS SERVO57D V1.0 Schematic.pdf》 in
https://github.com/makerbase-mks/MKS-SERVO57C

Part10. contact us

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