MKS SERVO42C V1.1.2

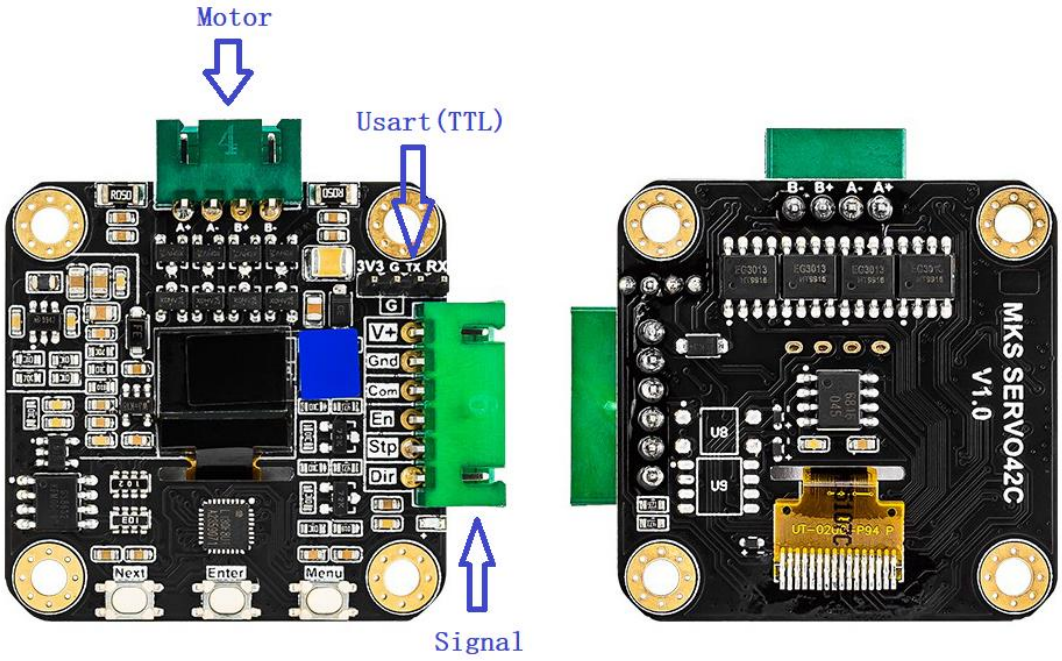
USER MANUAL

Note: This manual applies to firmware version V1.1.2

Part1. Feature

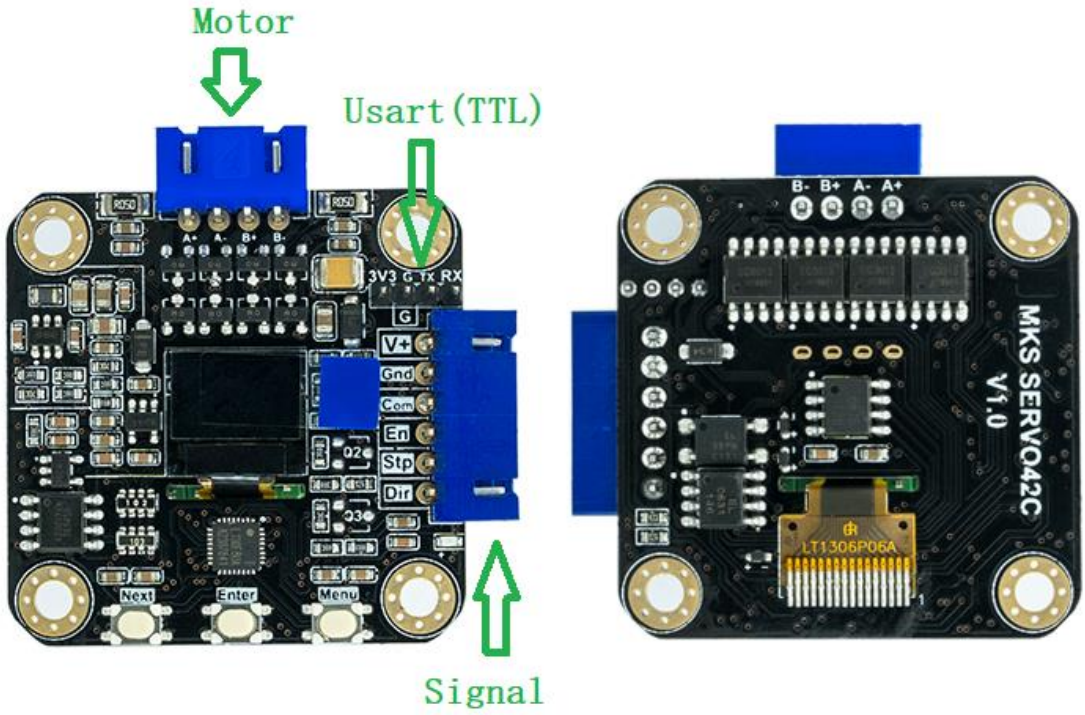
1.1 Interface

1. MKS SERVO42C



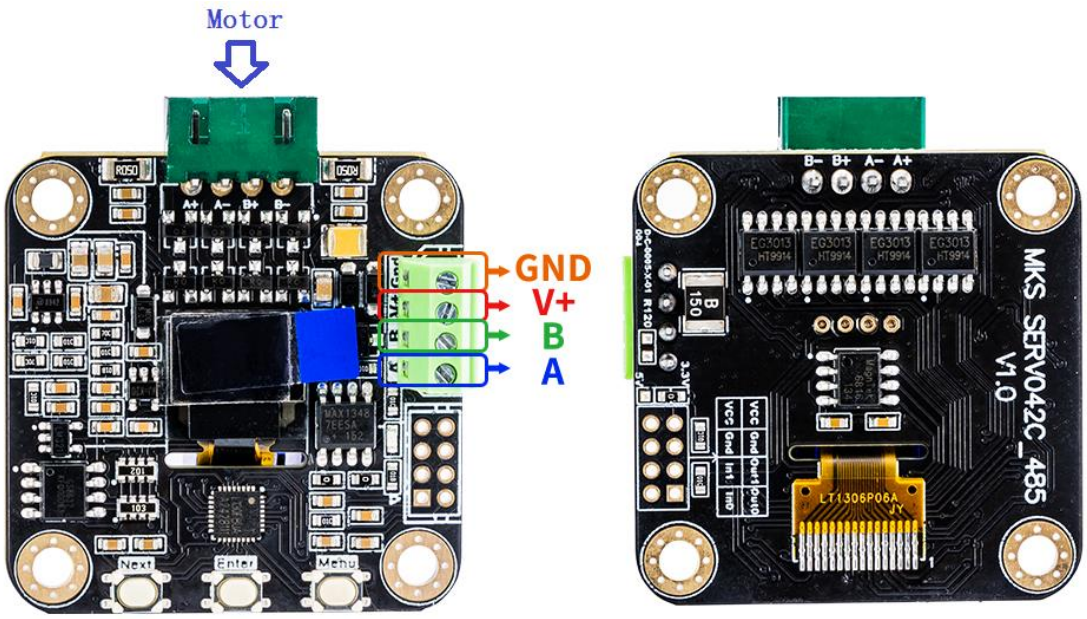
V+	12V~24V
Gnd	GND
Com	Floating
En	Enable signal (3.3V-24V)
Stp	Pulse signal (3.3V-24V)
Dir	Direction signal (3.3V-24V)
Usart(TTL)	
3V3	floating
G	GND
Tx	Serial output data (TTL) SERVO42C → PC
Rx	Serial input data (TTL) SERVO42C ← PC
Motor	
A+	Motor coil A
A-	
B+	Motor coil B
B-	

2. MKS SERVO42C-OC



Signal	
V+	12V~24V
Gnd	GND
Com	3.3V-5.0V, Just the signal high level voltage
En	Enable signal (3.3V-5.0V)
Stp	Pulse signal (3.3V-5.0V)
Dir	Direction signal (3.3V-5.0V)
Usart(TTL)	
3V3	floating
G	GND
Tx	Serial output data (TTL) SERVO42C → PC
Rx	Serial input data (TTL) SERVO42C ← PC
Motor	
A+	Motor coil A
A-	
B+	Motor coil B
B-	

3. MKS SERVO42C-485



V+	12V~24V
Gnd	GND
A	RS485-A (±15kV ESD-protected)
B	RS485-B (±15kV ESD-protected)
Motor	
A+	Motor coil A
A-	
B+	Motor coil B
B-	

1.2 Key Operation

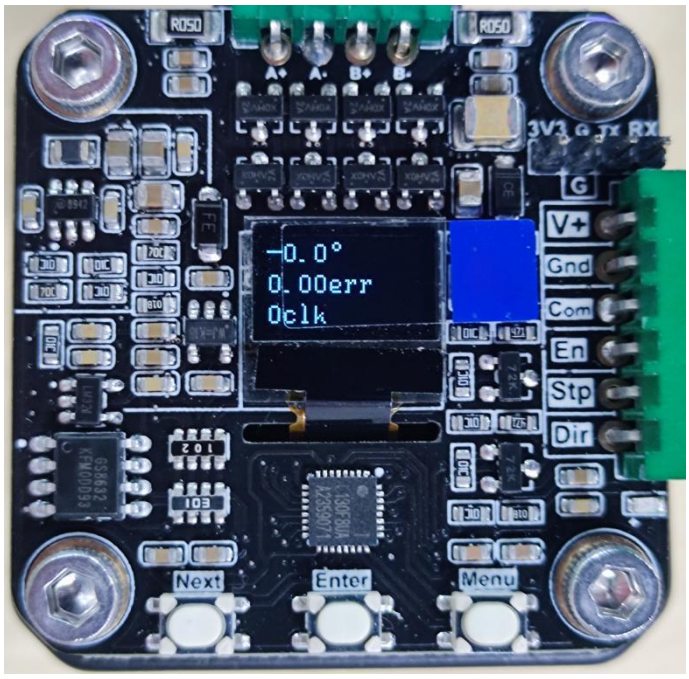
Key	Function
Next	move down
Enter	Confirm
Menu	Enter/exit parameter setting menu

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).
2. 0.00err - the err of the motor shaft angle.
3. 0clk - the pulses have been received.



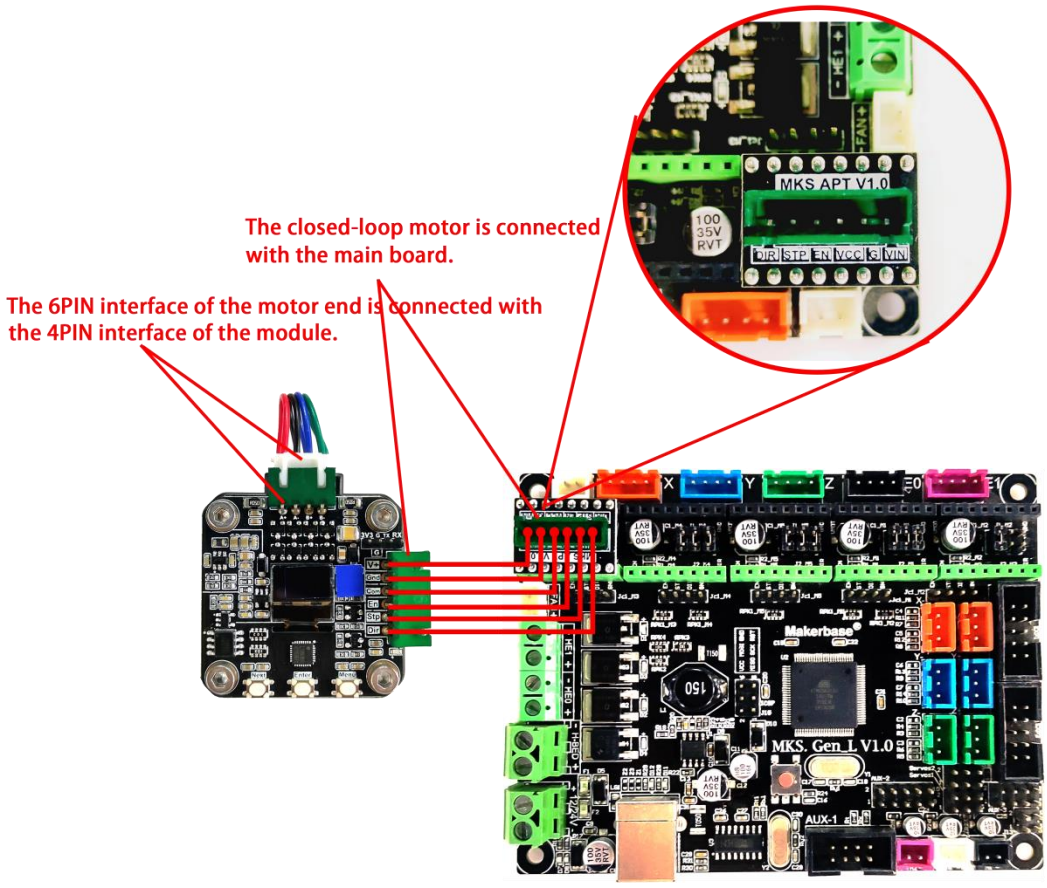
1.4 Work mode

Work mode	Signal input	Encoder	Current
CR_OPEN	EN,STP,DIR	NO	Fixed
CR_vFOC	EN,STP,DIR	YES	adaptive
CR_UART	Serial	YES	adaptive

Note: The default work mode is CR_vFOC

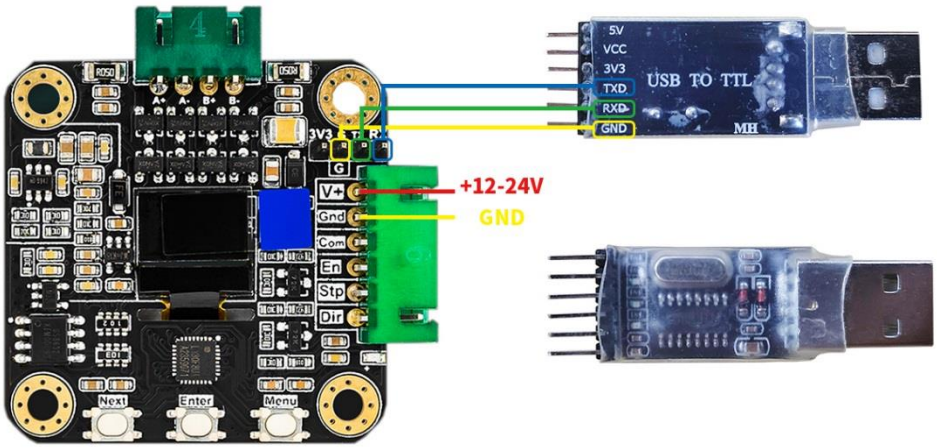
Part2. Wire

2.1 Pulse interface wire

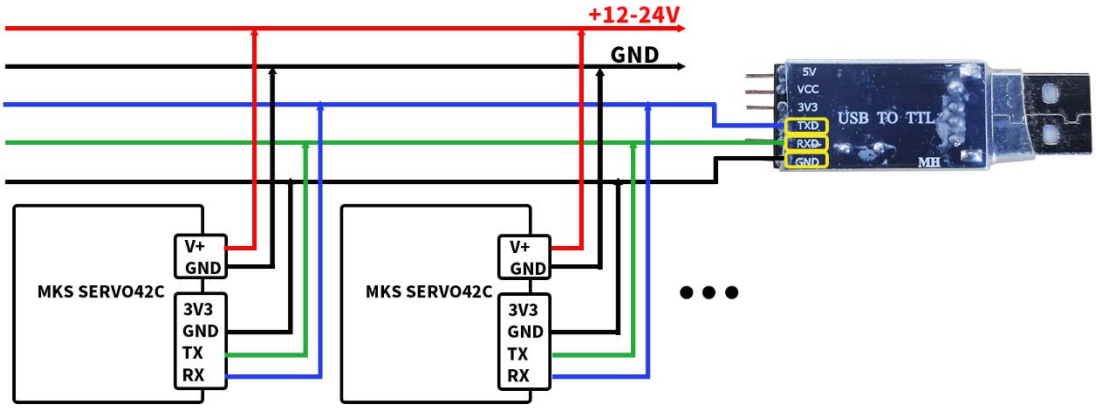


2.2 Uart (TTL) wire

1. Signal wire



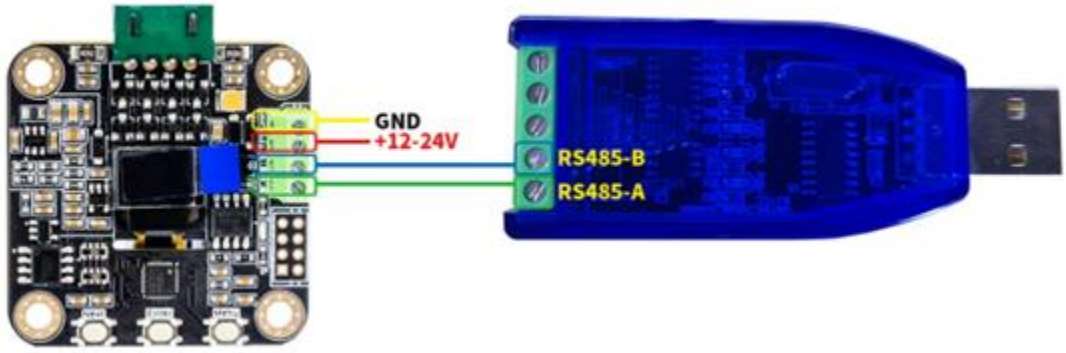
2. Multiple wire



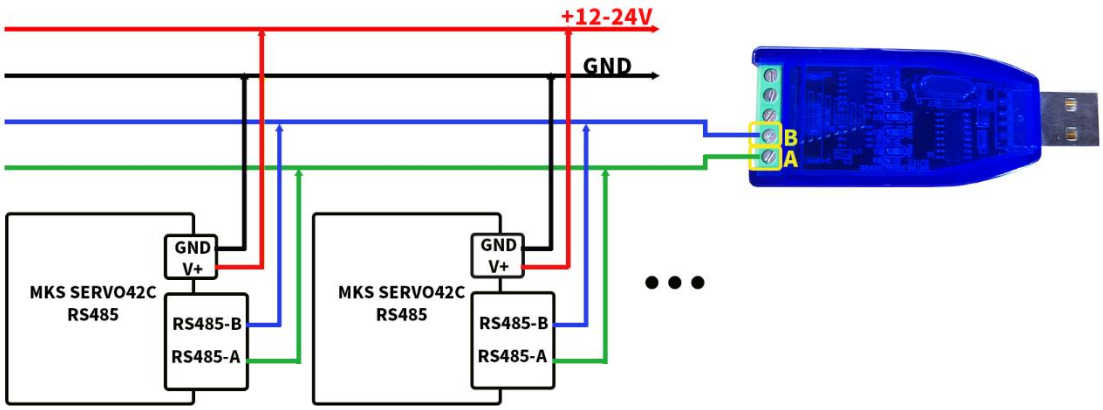
2.3 RS485 wire

Note: only for the MKS SERVO42C-485

1. Single wire



2. Multiple wire



Part3. Menu description

1. **CAL** : Calibrate the motor.
   Make sure the MotType is Configure correctly.(default 1.8 °)

2. **MotType** :Select stepper motor type.
   0.9 ° : The motor is a 0.9 degree stepper motor
   1.8 ° : The motor is a 1.8 degree stepper motor
   (Default: 1.8 °)
   (note: when MotType is changed ,the motor must be calibrated again)

3. **Mode** : Work mode selection.
   CR_OPEN : Open mode, the motor run without encoder
   CR_vFOC : FOC mode, pulse(En,Stp,Dir) interface.
   CR_UART : FOC mode, serial interface.
   (Default: CR_vFOC)

4. **Ma** :Set the operating current in CR_OPEN mode.
   Attention: The CR_vFOC and CR_UART mode will automatically adjust the current according to the load.
   (Default: 1200)

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 .

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)

8. **AutoSDD** :Set the oled screen sleep mode.
   Note : If this option is enabled, the OLED screen will automatically turn off without any key operation within seven seconds, and the display can be waked up by pressing any button.
   (Default: Disable)

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

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. **UartBaud** : Set the baud rate of serial.
   9600,
   19200,
   25000,
   38400,
   57600,
   115200.
   (Default: 38400)
   Note: The baud rate must be configured Before using serial communication.

12. UartAddr :Set the ID of the motor for serial command.
   0xe0
   0xe1
   .
   .
   .
   0xe9
   (Default: 0xe0)
   Note: If connected to multiple motors, the ID of each motor must be different.

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

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

15. **0_Speed** : Set the speed of go back to zero point.
0 : fastest.
...
4 : slowest.

16. **0_Dir** : Set the direction of go back to zero point.
CW : Clockwise.
CCW : Counterclockwise.
(Default: CW)
Note: For NearMode, the setting of 0_Dir should be consistent with the actual running direction of the motor, otherwise it will fail to return to zero.

17. Goto 0: go back to zero point.
(“0_Mode” must not be Disable and “Set 0” has been done.)

18. ACC:Set the acceleration of the motor.
Disable
286
…
1042
(Default: Disable)

19. **Restore** : Reload the default parameters.
After restored the default parameters, it needs to restart the motor.

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

Part4. Serial data format

Downlink package(PC → SERVO42C)			
Slave addr	Function	Data	CRC Code
addr	code		tCHK
Uplink package (PC ← SERVO42C)			
Slave addr		Data	CRC Code
addr			rCHK

1. The slave address(addr) range is e0~e9. (default is e0).
2. The function code (code) executes the corresponding command.
   for example, 0x80 executes the calibration command.
3. The CRC Code is CHECKSUM 8bit
   For example: command "E0 30 tCHK"
   tCHK = (0xE0 + 0x30) & 0xFF = 0x110 & 0xFF = 0x10

Part5. Serial command description

Note: Please set the serial slave address first.(default:0xe0)

The default address for the following chapters is 0xe0.

5.1 Read parameter command

1. command1 : e0 30 CRC

read the encoder value (the motor should be calibrated).

Uplink package (PC ← SERVO42C)			
Slave addr	Data		CRC
e0	carry	value	tCHK
	int32_t	uint16_t	

carry: the carry vaule of the encoder.

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

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

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

Note: The encoder value is updated regardless of whether the motor is enabled or not.

Example:

Send e0 30 10

return e0 00 00 00 00 40 00 20

2. Command2 : e0 33 CRC

Read the number of pulses received.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	pulses(int32_t)	tCHK

Example:

Send e0 33 13

return e0 00 00 01 00 e1 (256 pulses)

3. Command3 : e0 39 CRC

read the error of the motor shaft angle.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	error(int16_t)	tCHK

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.

Example:

Send e0 39 19

return e0 00 B7 97 (error 1 °)

4. Command4 : e0 3A CRC

read the En pins status.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	enable(uint8_t)	tCHK

enable =1 Enabled

enable =2 Disabled

Example:

Send e0 3a 1a

return e0 01 e1 (enable)

5. command6 : e0 3D CRC

Release the motor shaft locked-rotor protection state.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	tCHK

status =1 release success.

status =0 release fail.

Example:

Send e0 3d 1d

return e0 01 e1 (success)

6. Command7 : e0 3E CRC

Read the motor shaft protection state.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	tCHK

status =1 protected.

status =2 no protected.

Example:

Send e0 3e 1e

return e0 02 e2 (unblocked)

5.2 Set parameters command

1. Calibrate the encoder

（Same as the "Cal" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	80	00	tCHK

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Calibrated success.
status =2 Calibrating fail.
Note : The motor must be unloaded.
Example:
Send e0 80 00 60
return e0 01 e1 (successful)

2. Set the Motor Type

（Same as the "MotType" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	81	Type (0~1)	tCHK

Type = 0 0.9 degree motor
Type = 1 1.8 degree motor

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 81 01 62 (1.8 degree motor)
return e0 01 e1 (successful)

3. Set the work mode

（Same as the "Mode" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	82	mode (0~2)	tCHK

mode = 0 CR_OPEN
mode = 1 CR_vFOC
mode = 2 CR_UART

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 82 01 63 (CR_vFOC)
return e0 01 e1 (successful)

4. Set the current

（Same as the "Ma" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	83	ma (00~0F)	tCHK

The current = ma x 200 （mA）

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 83 06 69 (1200ma)
return e0 01 e1 (successful)

5. Set subdivision

（Same as the "MStep" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	84	micstep(00~FF)	tCHK

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

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 84 1a 7e (26 subdivision)
return e0 01 e1 (successful)

6. Set the active of the En pin

（Same as the "En" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	85	enable(00~02)	tCHK

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

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 85 00 65 (active low)
return e0 01 e1 (successful)

7. Set the direction of motor rotation

（Same as the "Dir" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	86	dir(00~01)	tCHK

dir = 00 CW
dir = 01 CCW

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 86 00 66 (CW)
return e0 01 e1 (successful)

8. Set automatic turn off the screen

（Same as the "AutoSDD" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	87	enable(00~01)	tCHK

enable = 01 enabled
enable = 00 disabled

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 87 00 67 (Disable)
return e0 01 e1 (successful)

9. Set the motor shaft locked-rotor protection function

（Same as the "Protect" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	88	enable(00~01)	tCHK

enable = 01 enabled protection
enable = 00 disabled protection

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 88 00 68 (Disable)
return e0 01 e1 (successful)

10. Set the subdivision interpolation function

（Same as the "Mplyer" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	89	enable(00~01)	tCHK

enable = 01 enabled interpolation function.
enable = 00 disabled interpolation function.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 89 00 69 (Disable)
return e0 01 e1 (successful)

11. Set the baud rate

（Same as the "UartBaud" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	8A	baud(01~06)	tCHK

baud = 01 9600.
baud = 02 19200.
baud = 03 25000.
baud = 04 38400.
baud = 05 57600.
baud = 06 115200.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 8a 04 6e (38400)
return e0 01 e1 (successful)

12. Set the slave address

（Same as the "UautAddr" option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	8B	addr(00~09)	tCHK

Slave address = addr + 0xe0

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 8b 02 6d (the slave address is e2)
return e0 01 e1 (successful)

13. Restore the default parameter

（Same as the "Restore" option on screen）

Downlink package (PC → SERVO42C)		
Slave addr	Function	CRC
e0	3F	tCHK

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 restore success.
status =0 restore fail.
Example:
Send e0 3f 1f
return e0 01 e1 (successful)

Note: After restoring the default parameters, you need to power on again and set the serial baud rate before serial communication.

5.3 zeroMode parameters command

1. Set the mode of zeroMode

（Same as the " 0_Mode " option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	90	mode(00~02)	tCHK

mode = 00 Disable
mode = 01 DirMode
mode = 02 NearMode

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 90 01 71 (DirMode)
return e0 01 e1 (successful)

2. Set the zero of zeroMode

（Same as the ″ Set 0 ″ option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	91	00	tCHK

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 91 00 71
return e0 01 e1 (successful)
Note: The mode of ″0_Mode″ needs to be set first.

3. Set the speed of zeroMode

（Same as the ″ 0_Speed ″ option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	92	speed(00~04)	tCHK

(speed = 0~4, the smaller the value, the faster the speed)

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 92 02 74
return e0 01 e1 (successful)
Note: The mode of ″0_Mode″ needs to be set first.

4. Set the dir of zeroMode

（Same as the " 0_Dir " option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	93	dir(00~01)	tCHK

dir = 00 CW
dir = 01 CCW

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 93 00 73 (CW)
return e0 01 e1 (successful)
Note: For NearMode, the setting of 0_Dir should be consistent with the actual running direction of the motor, otherwise it will fail to return to zero.

5. Return to zero

（Same as the " Goto 0 " option on screen）

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	94	00	tCHK

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 94 00 74
return e0 01 e1 (successful)

5.4 Set PID/ACC/Torque command

1. Set the position Kp parameter

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	a1	Kp(uint16_t)	tCHK

(Default Kp is 0x650).

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 a1 01 20 a2 (Kp = 0x120)
return e0 01 e1 (successful)

2. Set the position Ki parameter

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	a2	Ki(uint16_t)	tCHK

(Default Ki is 1).

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 a2 00 02 84 (Ki = 0x02)
return e0 01 e1 (successful)

3. Set the position Kd parameter

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	a3	Kd(uint16_t)	tCHK

(Default Kd is 0x650).

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.

status =0 Set fail.

Example:

Send e0 a3 02 50 d5 (Kd = 0x250)

return e0 01 e1 (successful)

4. Set the acceleration (ACC) parameter

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	a4	ACC(uint16_t)	tCHK

(Default ACC is 0x11e).

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.

status =0 Set fail.

Example:

Send e0 a4 00 80 04 (ACC = 0x80)

return e0 01 e1 (successful)

Important Note: If the ACC parameter is set too large, it may damage the driver board, please set the parameters carefully!

5. Set the maximum torque (MaxT) parameter

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	a5	MaxT(0~0x4B0)	tCHK

(Default MaxT is 0x4B0).

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status =1 Set success.
status =0 Set fail.
Example:
Send e0 a5 02 58 df (MaxT = 0x258)
return e0 01 e1 (successful)

5.5 Serial control command

Note: This chapter needs to set the working mode to "CR_UART".

1. Set the En pin status in CR_UART mode.

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	F3	en (00~01)	tCHK

en = 00 disable.
en = 01 enable.

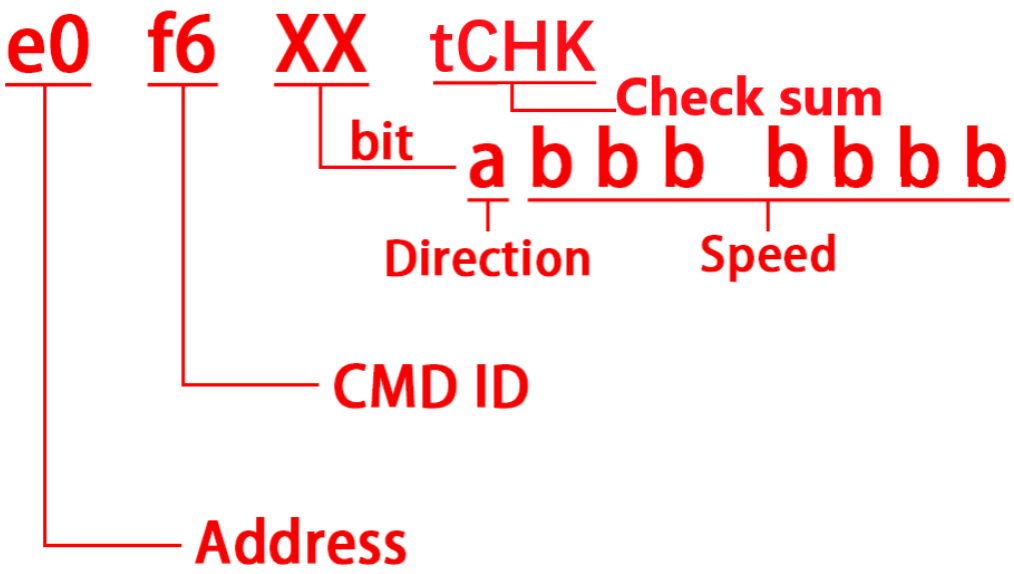
Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status = 1 set success.
status = 0 set fail.
Example:
Send e0 f3 01 d4 (enable)
return e0 01 e1 (successful)

2. run the motor forward / reverse in a Constant speed.

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	F6	VAL(00~FF)	tCHK

Direction : The highest 1bit of VAL.
Speed : The lowest 7bit of VAL.
for example:



The Vrpm calculation formula is:
Vrpm = (Speed × 30000)/(Mstep × 200)(RPM) (1.8 degree motor)
Vrpm = (Speed × 30000)/(Mstep × 400)(RPM) (0.9 degree motor)
For example:
1.8 degree motor, Mstep = 150, Speed=1
Vrpm = (1 × 30000)/(150 × 200) = 1
Part of the speed is as follows:

1.8 degree motor	Vrpm (RPM)	
Speed	Mstep=16	Mstep=150
1	9.375	1
2	18.75	2
3	28.125	3
4	37.5	4
…	…	…
8	75	8
…	…	…
0x10	150	16
0x20	300	32
0x40	600	64
0x7F	1200	127

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
01	status(uint8_t)	rCHK

status = 1 run success.
status = 0 run fail.
Such as:
Send “e0 f6 01 d7”, The motor rotates forward at the speed of 1;
Send “e0 f6 81 d7”, The motor reverses at the speed of 1;
Send “e0 f6 5a 30”, The motor rotates forward at the speed of 90;
Send “e0 f6 da b0”, The motor reverses at the speed of 90;
Example:
Send e0 f6 10 e6 (speed of 16)
return e0 01 e1 (successful)
Note: the Vrpm no great than 2000RPM.

3. stop the motor

Downlink package (PC → SERVO42C)		
Slave addr	Function	CRC
e0	F7	tCHK

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status = 1 stop the motor success.
status = 0 stop the motor fail.
Example:
Send e0 f7 d7
return e0 01 e1 (successful)

4. Save/Clear the status set in function(F6)

Downlink package (PC → SERVO42C)			
Slave addr	Function	Data	CRC
e0	FF	state	tCHK

state = C8 Save.
state = CA Clean.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(uint8_t)	rCHK

status = 1 success.
status = 0 fail.

note1: The motor can rotates clockwise or counterclockwise at a constant speed when powered on.
note2: After the save is successful, the driver board will be disable, and it needs to be re-enabled.

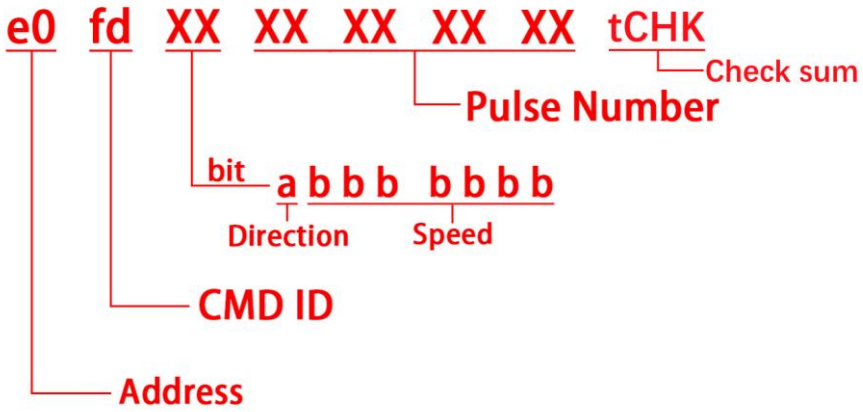
For example:

Step1: Send e0 f6 10 e6
return e0 01 e1
Step2: Send e0 ff c8 a7
return e0 01 e1
the motor will always rotates clockwise in speed(16) when power on.

5. Run the motor by serial command

Downlink package (PC → SERVO42C)				
Slave addr	Function	Data		CRC
e0	FD	VAL	pulses	tCHK
		uint8_t	uint32_t	

VAL: is the direction and speed.
pulses: is the number of pulses.



The speed (Vrpm) calculation formula is the same as above.

Uplink package (PC ← SERVO42C)		
Slave addr	Data	CRC
e0	status(0~2)	rCHK

status = 0 run fail.
status = 1 run starting….
status = 2 run complete.
Note: the Vrpm no great than 2000RPM.

Part6. Serial example

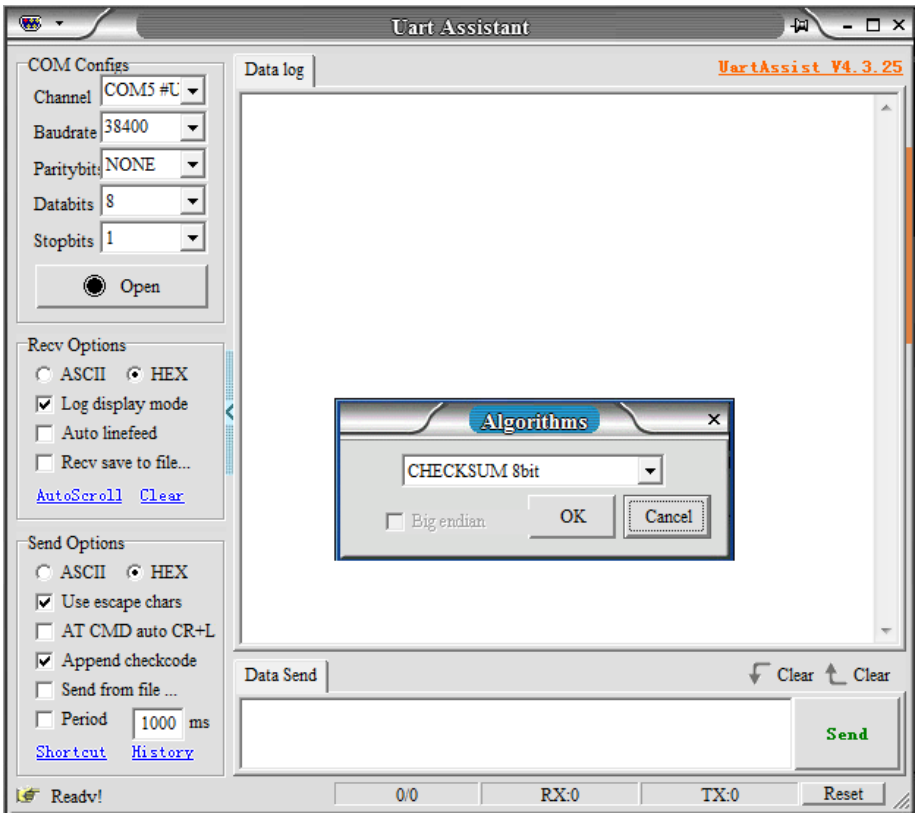
6.1 Config the SERVO42C

1. Menu → Mode → CR_UART.
2. Menu → UartBaud → 38400.
3. Menu → UartAddr → 0xe0.

6.2 Config the Uart Assistant

1. Select the Channel; (such as COM5).
2. Select the Baudrate; (such as 38400, Must be equal to motor baudrate).
3. Recv Options: select “HEX”.
4. Send Options: select “HEX”.
5. Append checkcode: select “CHECKSUM-8”.

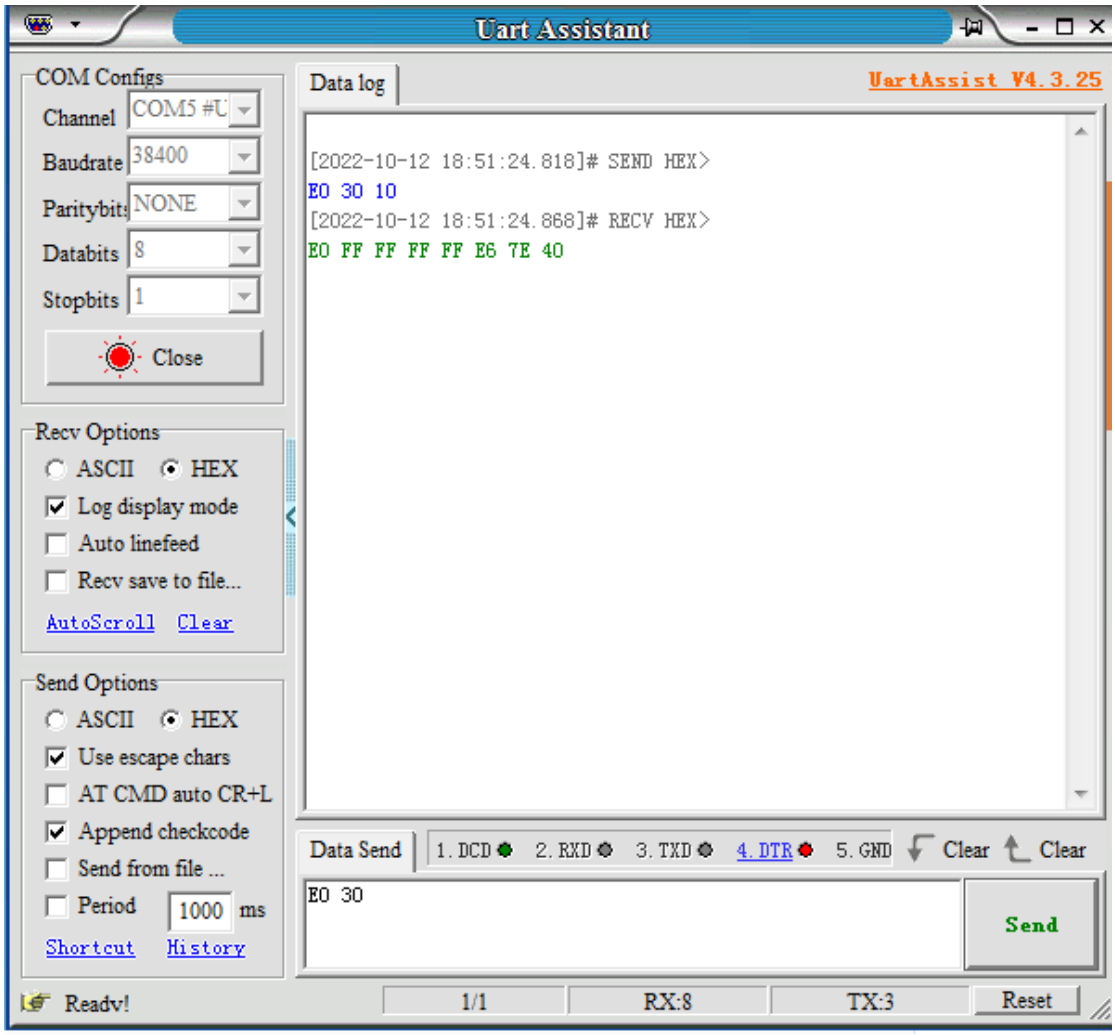
Such as below:



6.3 Read the encoder value

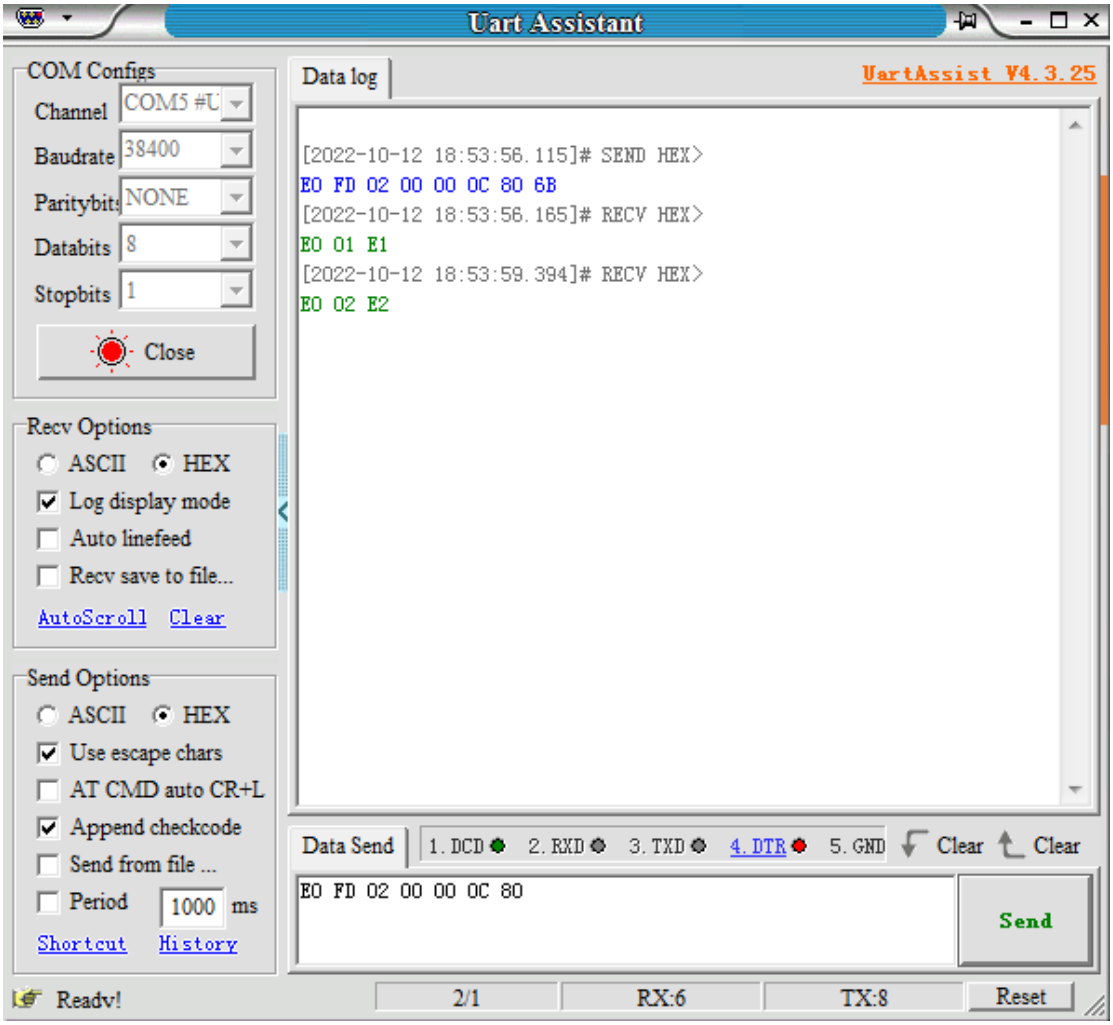
send “E0 30”

return “E0 FF FF FF FF E6 7E 40”



6.4 Run the motor

Note : Please configure the working mode to “CR_UART”.
Send “E0 FD 02 00 00 0C 80”
Return1 “E0 01 E1” run starting….
Return2 “E0 02 E2” run complete.
The motor will run one revolution at speed 2. (subdivision 16)



Part7. FAQ

7.1 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).

7.2 FAQ

No	Question	Solution
1	Not Cal	Calibrate the motor.
2	Waiting V+ Power!	The input voltage is 12V-24V ?
3	Phase Line Error!	The motor line sequence is wrong.
4	Reverse Lookup Error!	Check magnet and motor shaft.
5	Magnet Loss!	The magnet is not installed.
6	Magnet Error!	Check magnet and motor shaft
7	Motor Type Error!	Check motor type and motor shaft.
8	Offset Current Error!	Hardware error.
9	Wrong Protect!	Locked-rotor protection

Part8. Schematic

Please download 《MKS SERVO42C V1.0 Schematic.pdf》 in
https://github.com/makerbase-mks/MKS-SERVO42C

Part9. contact us

https://makerbase.aliexpress.com/
https://www.youtube.com/channel/UC2i5I1tc0XRJ2ZJiRxwpCUQ
https://github.com/makerbase-mks