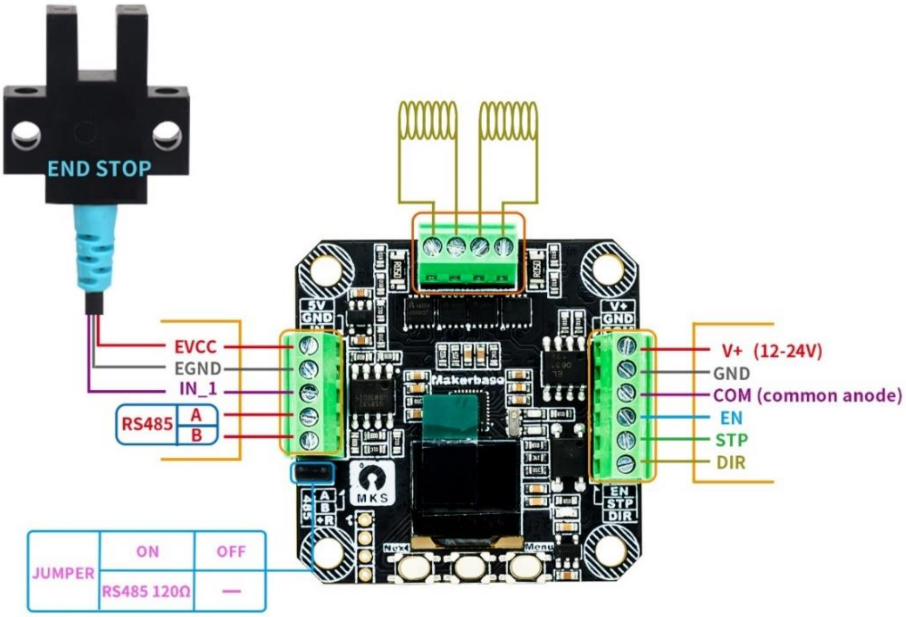
MKS SERVO42D/57D_RS485 V1.0.5
USER MANUAL

MKS SERVO42D/57D_RS485 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.Add long data package.	V1.0.2	May-2023
	2.Support for Modbus-RTU communication protocol.		
	3.Slave does not answer if broadcast address or group address or long data package is used.		
	4. 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 "94" 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).		

Part1. Feature

1.1 Interface

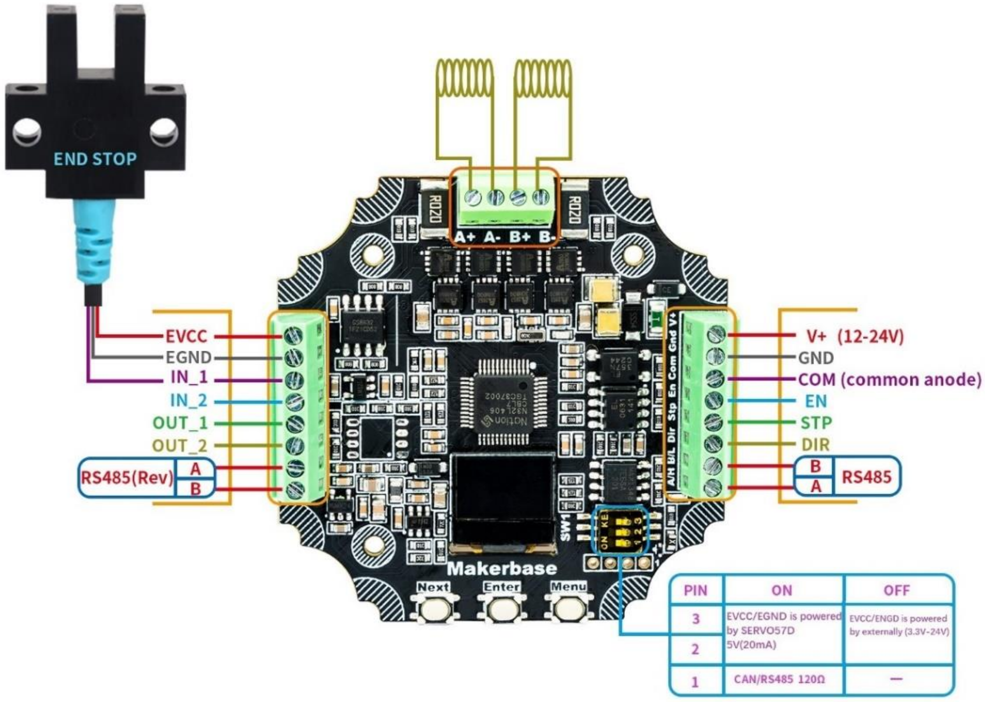
1. SERVO42D_RS485 Interface



JUMPER	ON	OFF
	RS485 120Ω	—

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

2. SERVO57D_RS485 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		

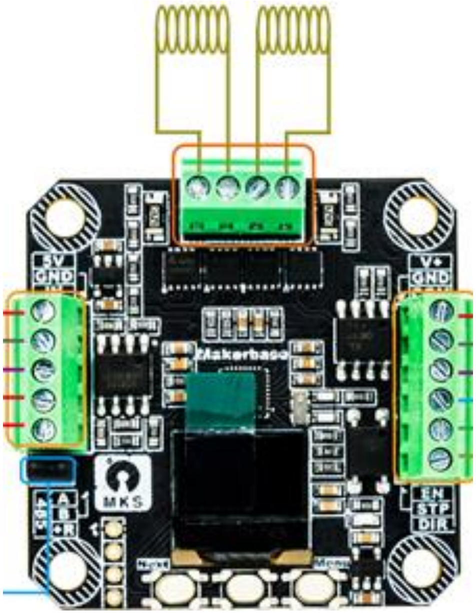
Note1: The default work mode is CR_vFOC.
Note2: It can work without encoder in “OPEN” work mode.

Part2. Wire

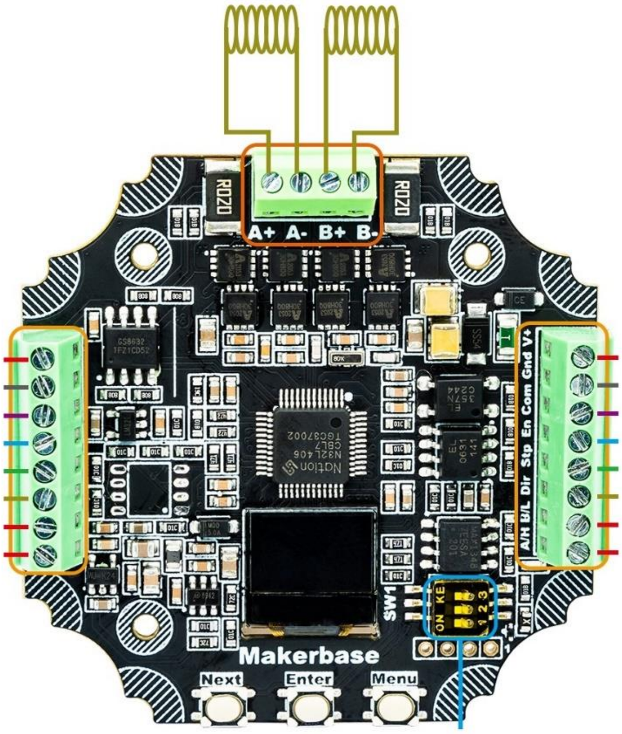
2.1 Motor wire

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

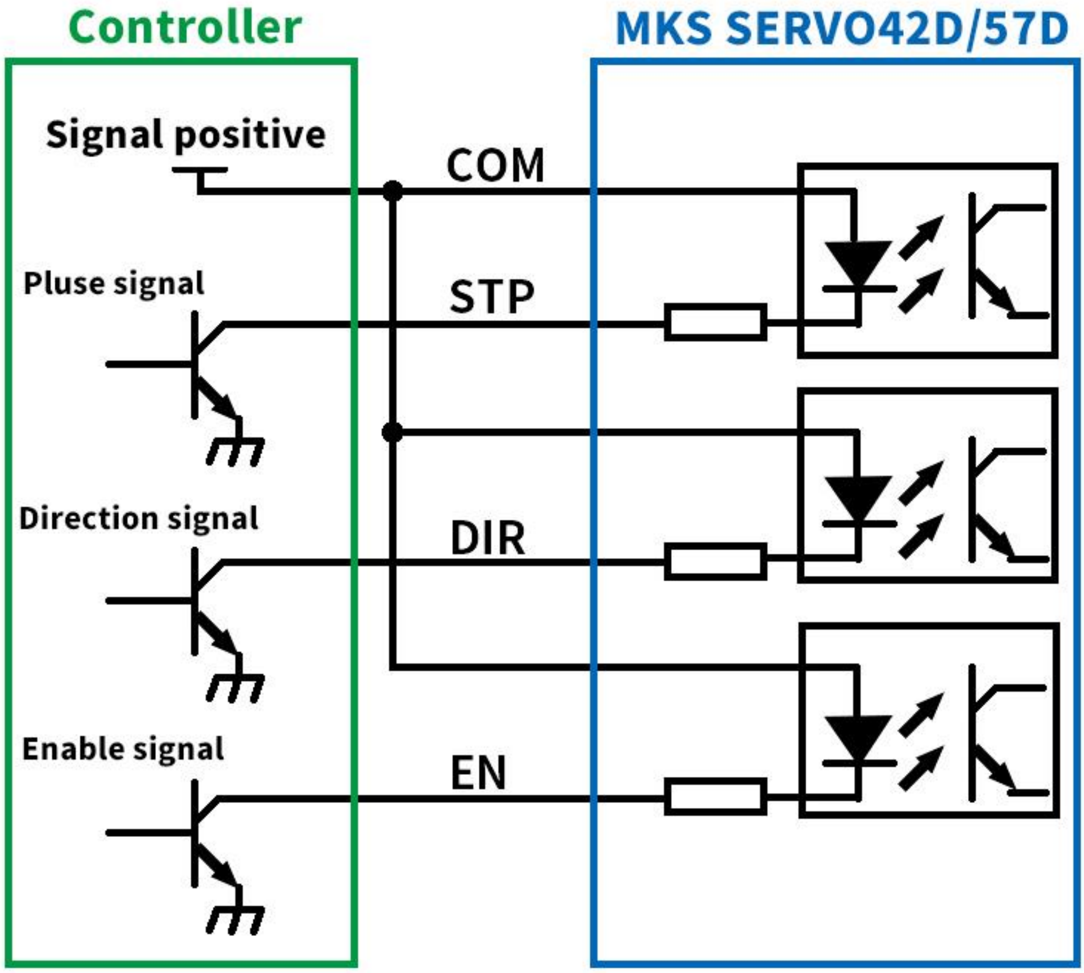
1. SERVO42D_RS485 motor wire



2. SERVO57D_RS485 motor wire



2.2 Pulse interface wire

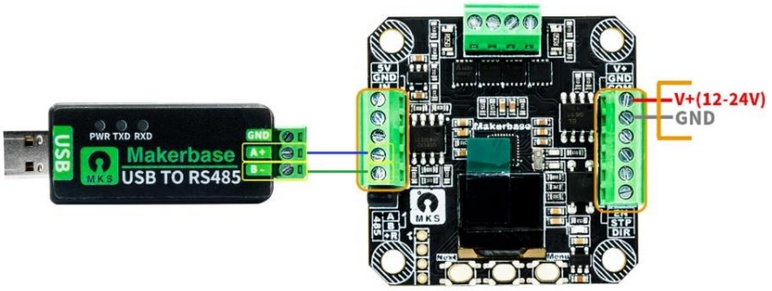


Note: if the (STP/DIR/EN)signal high level is 3.3V,the COM must be 3.3V
if the (STP/DIR/EN)signal high level is 5.0V,the COM must be 5.0V
…
and so on.

2.3 RS485 wire

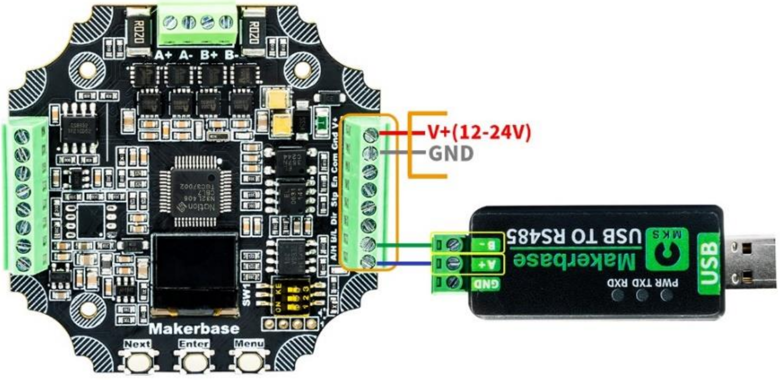
Note: Tip: In order to reduce bus interference, the host-gnd and the motor-gnd must connected togeter, and RS485 signals are transmitted using shielded twisted pairs.

1. SERVO42D_RS485 Single-slave



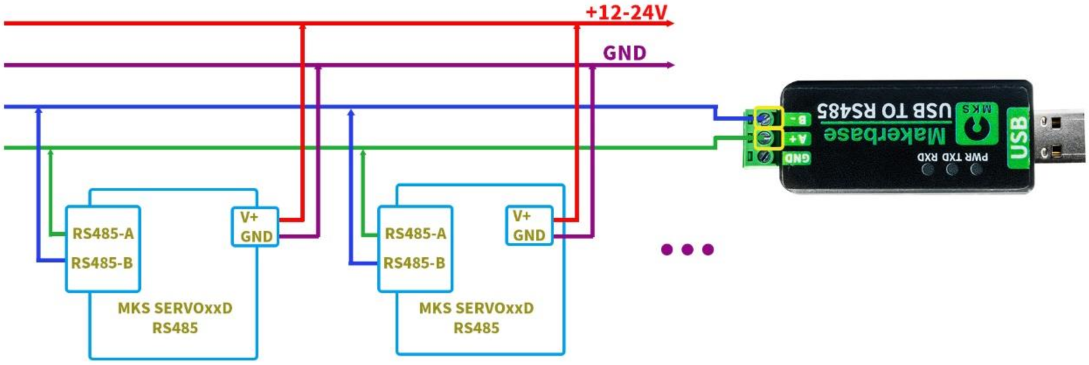
Note: Single slave communication does not need 120Ω Terminal.

2. SERVO57D_RS485 Single-slave



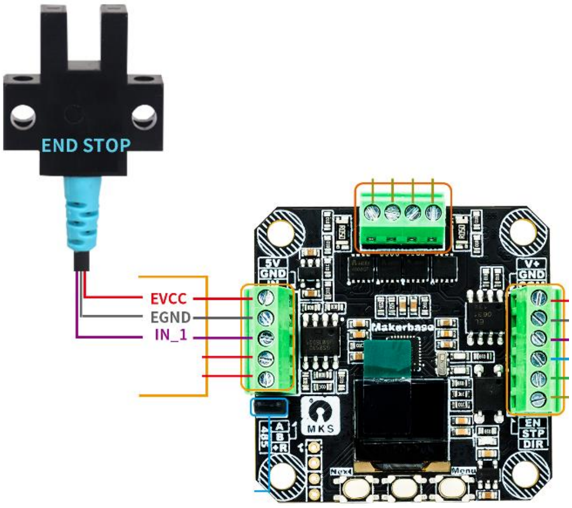
Note: Single slave communication does not need 120Ω Terminal.

3. Multiple-slave



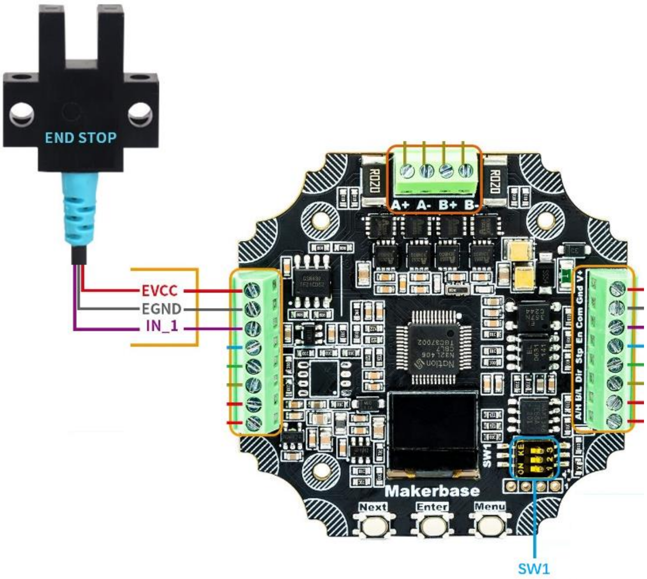
2.4 End stop wire

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	RS485 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 . It will be added to the last options.

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: 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.
   9600/19200/25000/38400/57600/115200/256000.
   (Default: 38400)

12. **UartAddr** : Set the the slave address of seria.
    01
    …
    15
    16
    (Default: 01)
    Note: The addresses greater than 16 need to be set by serial command. After it is set, it will be added to this option.

13. **UartRSP** : 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 command "F1".

14. **Mb_RTU** : Choose whether to use MODBUS-RTU communication protocol.
    Disable: disable MODBUS-RTU communication protocol
    Enable: enable MODBUS-RTU communication protocol
    (Default: Disable)

15. **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)

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

17. **0_Speed** : Set the speed of go back to zero point.
    0 ： slowest.
    ...
    4 ： fastest.

18. **0_Dir** : Set the direction of go back to zero point.
    CW ： Clockwise.
    CCW ： Counterclockwise.
    (Default: CW)

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

20. **Hm_Dir** ： Set the direction of go home.

CW ： Clockwise rotation is positive

CCW ： Counterclockwise rotation is positive

(Default: CW)

21. **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.

22. **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.

23. **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.

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

Disable: disable endstop-limit

Enable: enable endstop-limit

(Default: Disable)

Note1: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″)

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.

25. **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.

26. **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.

27. **About** : Show version parameters.

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

Part4. Serial data format

Note: For MODBUS-RTU protocol commands, see Part 8.

Downlink package(PC → SERVO42D/57D)						
Head	Slave addr	Function	Data			Check code
FA	addr	code				CRC
Uplink package (PC ← SERVO42D/57D)						
Head	Slave addr	Function	Data			Check code
FB	addr	code				CRC

1. Downlink package Head is "FA", uplink package Head is "FB".
2. The slave address(addr) range is 00~255. (default is 01).
   00 is the broadcast address;
   01~10 can be set in the UartAddr option of the display menu;
   greater than 10 need to be set by serial commands.
3. The function code (code) executes the corresponding command.
   for example, 0x80 executes the calibration command.
4. The Check code is CHECKSUM 8bit
   For example: command "FA 01 80 00 CRC"
   CRC = (0xFA + 0x01 + 0x80 + 0x00) & 0xFF = 0x17B & 0xFF = 0x7B

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

Part5. Serial command description

Note1: Please set the serial slave address first.(default:01)

The default address for the following chapters is 01.

Note2: For MODBUS-RTU protocol commands, see Part 8.

5.1 Read parameter command

1. command1 : FA 01 30 CRC

read the encoder value(carry).

Uplink package (PC ← SERVO42D/57D)					
Head	Slave addr	Function	Data		CRC
FB	01	30	carry	value	CRC
			int32_t	uint16_t	

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.

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

2. Command2 : FA 01 31 CRC

read the encoder value(addition).

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	value	CRC
FB	01	31	(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 : FA 01 32 CRC

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	value	CRC
FB	01	31	speed(int16_t)	CRC

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

4. Command4 : FA 01 33 CRC

Read the number of pulses received.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	33	pulses(int32_t)	CRC

5. Command6 : FA 01 34 CRC

read the IO Ports status.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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. Command5 : FA 01 39 CRC

read the error of the motor shaft angle.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	39	error(int32_t)	CRC

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

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

7. Command6 : FA 01 3A CRC

read the En pins status.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	3A	enable(uint8_t)	CRC

enable =1 Enabled

enable =0 Disabled

8. Command7 : FA 01 3B CRC

Read the go back to zero status when power on.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	3B	status(uint8_t)	CRC

status =0 going to zero.

status =1 go back to zero success.

status =2 go back to zero fail.

9. Command8 : FA 01 3D CRC

Release the motor shaft locked-rotor protection state.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	3D	status(uint8_t)	CRC

status =1 release success.

status =0 release fail.

10. Command9 : FA 01 3E CRC

Read the motor shaft protection state.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	80	00	CRC

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	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 package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	83	Current(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 package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	9B	holdMa(01~08)	CRC

hold = 00 10%.

hold = 01 20%.

…

hold = 08 90%.

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	84	micstep(00~FF)	CRC

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

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	85	enable(00~02)	CRC

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	86	dir(00~01)	CRC

dir = 00 CW
dir = 01 CCW

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	86	status(uint8_t)	CRC

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

8. Set auto turn off the screen function

(Same as the "AutoSDD" option on screen)

Downlink package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	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 package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	88	enable(00~01)	CRC

enable = 01 enabled protection

enable = 00 disabled protection

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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.

10. Set the subdivision interpolation function

(Same as the "Mplyer" option on screen)

Downlink package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	89	enable(00~01)	CRC

enable = 01 enabled interpolation function.

enable = 00 disabled interpolation function.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	89	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

11. Set the baud rate

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

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	8A	baud(01~07)	CRC

baud = 01	9600.
baud = 02	19200.
baud = 03	25000.
baud = 04	38400.
baud = 05	57600.
baud = 06	115200.
baud = 07	256000.

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	8A	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

12. Set the slave address

(Same as the "UautAddr" option on screen)

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	8B	addr(00~FF)	CRC

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

Note2: 0 is the broadcast address

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	8B	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

13. Set the slave respond and active

Downlink package（PC → SERVO42D/57D）					
Head	Slave addr	Function	Data	Data	CRC
FA	01	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 package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 02 or 03 after the motor finishes running or touches the limit stop.

14. Set MODBUS-RTU communication protocol

(Same as the "MB_RTU" option on screen)

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	8E	enable(00~01)	CRC

enable = 01　enabled Modbus-RTU
enable = 00　disabled Modbus-RTU

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	8E	status(uint8_t)	CRC

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

15. Set the key lock or unlock

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	8F	enable(00~01)	CRC

enable = 01 lock the key

enable = 00 unlock the key

Uplink package（PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	8F	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

16. Set the group address

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	8D	addr(01~FF)	CRC

Uplink package（PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	8D	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

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

	Broadcast addr	Slave addr	Group addr
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 FA 01 FD 01 2C 64 00 00 0C 80 15, motor 1 will rotate a turn

send FA 00 FD 01 2C 64 00 00 0C 80 14, motor1-6 will rotate a turn

send FA 50 FD 01 2C 64 00 00 0C 80 64, motor1-3 will rotate a turn

send FA 51 FD 01 2C 64 00 00 0C 80 65, motor4-6 will rotate a turn

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

5.3 Set Home command

1. Set the parameter of home

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

Downlink package (PC → SERVO42D/57D)							
byte1	byte2	byte3	byte 4	byte 5	byte 6-7	byte 8	byte 9
Head	Slave addr	Function	level	dir	speed	enable	Check
FA	01	90	HmTrig	HmDir	HmSpeed	EndLimit	CRC

HmTrig the effective level of the end stop

0: Low 1: High

HmDir the direction of go home

0: CW 1: CCW

HmSpeed the speed of go home

0~3000 (RPM)

EndLimit

0: disable endstop-limit

1: enable endstop-limit

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	91	NULL	CRC

Note: 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 package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	92	NULL	CRC

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	92	status(uint8_t)	CRC

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

4. Set the parameter of "noLimit" go home

Downlink package (PC → SERVO42D/57D)						
byte1	byte2	byte3	byte 4-7	byte 8	byte 9-10	byte 11
Head	Slave addr	Function	Reverse Angle	hm-Mode	Hm_ma	Check
FA	01	94	retValue	mode	ma	CRC

mode	0: used Limit switch for go home
	1: no Limit switch for go home
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 package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package (PC → SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FA	01	9E	enable(00~01)	CRC

enable = 01 enable remap limit port

enable = 00 disable remap limit port

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	9E	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

5.4 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 package (PC → SERVO42D/57D)							
byte1	byte2	byte3	byte 4	byte 5	byte 6	byte 7	byte 8
Head	Slave addr	Function	0_Mode	Set 0	0_Speed	0_Dir	Check
FA	01	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

speed:

0 ~ 4 (0:slowest 4:fastest)

dir:

0: CW

1: CCW

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	9A	status(uint8_t)	CRC

status =0 set fail.

status =1 set success.

5.5 Restore the default parameter

（Same as the ″Restore″ option on screen）

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	3F	NULL	CRC

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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, and power on the motor, the default parameter will be restored.

5.6 Restart the motor

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	41	NULL	CRC

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	41	status(uint8_t)	CRC

status =1 restart success.

status =0 restart fail.

5.7 Long Data Package

Long data packets, that is, a packet of data contains up to 5 commands, and the slave judges which command to execute based on the address.

Long data packets format:

Head	0xFC				1byte
	byte 1	byte 2	…	byte 10	
command 1	slaveAddr1	code	…		10 byte
command 2	slaveAddr 2	code	…		10 byte
command 3	slaveAddr 3	code	…		10 byte
command 4	slaveAddr 4	code	…		10 byte
command 5	slaveAddr 5	code	…		10 byte
checksum	CRC				1 byte

Note:

1.The length of the long data packet is 52 bytes in total.

2.The length of each command X is 10 bytes, when it is less than 10 bytes, add 0 to supplement.

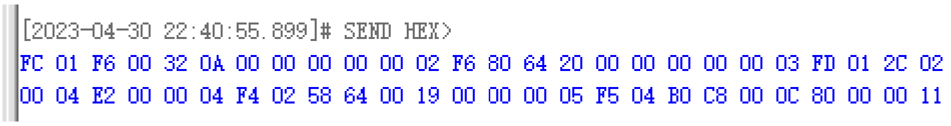
3.Command X is the corresponding ordinary command, remove the frame header (FA) and checksum.

4.If the slave addresses of command X and command Y (X<Y) are the same, only command X is executed.

5.Slave does not answer.

For example, sending the following long data packet can control 5 motors to perform different actions (16 subdivisions)

```
FC
01 F6 00 32 0A 00 00 00 00 00
02 F6 80 64 20 00 00 00 00 00
03 FD 01 2C 02 00 04 E2 00 00
04 F4 02 58 64 00 19 00 00 00
05 F5 04 B0 C8 00 0C 80 00 00
11
```



```
[2023-04-30 22:40:55.899]# SEND HEX>
FC 01 F6 00 32 0A 00 00 00 00 00 02 F6 80 64 20 00 00 00 00 00 03 FD 01 2C 02
00 04 E2 00 00 04 F4 02 58 64 00 19 00 00 00 05 F5 04 B0 C8 00 0C 80 00 00 11
```

Motor 1 rotates forward continuously in speed mode (speed=0x32, acc=0x0A)

Motor 2 reverses continuously in speed mode (speed=0x64, acc=0x20)

Motor 3 rotates forward 100 times in position mode 1 (speed=0x12C, acc=0x02)

Motor 4 rotates forward 100 times in position mode 2 (speed=0x258, acc=0x64)

Motor 5 runs to coordinate 0xC8000 in position mode 3 (speed=0x4B0, acc=0xC8)

Part6. Run the motor by serial command

Note1: This chapter needs to set the working mode to serial mode. (SR_OPEN/SR_CLOSE/SR_VFOC)

Note2: For MODBUS-RTU protocol commands, see Part 8.

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:

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

For example: acc = 236, speed = 3000

T(ms)	speed (RPM)
0	0
1	1
2	2
3	3
…	…

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

② decelerates

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

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

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

6.2 Query/Enable the motor command

1. Query the motor status

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	F1	—	CRC

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 = 5	motor is Cal…

2. Enable the motor

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	F3	en（00~01）	CRC

en = 00	disable.
en = 01	enable.

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	F3	status(uint8_t)	CRC

status = 1 set success.

status = 0 set fail.

6.3 Emergency stop the motor

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	F7	—	CRC

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 About Multiple Motors Control

1. Broadcast address, all motors execute the same command.
2. Group address, group A motors execute command a, group B motors execute command b.
3. Long data packets, motors can execute different commands.

6.5 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 package（PC → SERVO42D/57D）								
BYTE1	BYTE2	BYTE3	BYTE4			BYTE5	BYTE6	BYTE7
Head	Slave addr	Function	dir	Rev	speed		acc	CRC
FA	addr	F6	b7	b6-b4	b3-b0	b7-b0	acc	CRC
			dir	--	speed			

Byte 4: The highest bit indicates the direction, the lower 4 bits and byte 5 together indicate the speed

Byte 5: The lower 4 bits of byte 5 and byte 4 together indicate speed

The parameter description is as follows:

addr: slave address, the value range is 0-255

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 “FA 01 F6 01 40 02 34”,

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

Send “FA 01 F6 81 40 02 B4”,

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

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	F6	status(uint8_t)	CRC

status = 1 run success.

status = 0 run fail.

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

2. Stop the motor in speed mode

Downlink package（PC → SERVO42D/57D）								
BYTE1	BYTE2	BYTE3	BYTE4			BYTE5	BYTE6	BYTE7
Head	Slave addr	Function	dir	Rev	speed		acc	CRC
FA	addr	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 FA 01 F6 00 00 02 F3

Stop the motor with deceleration acc=2

② Immediate stop command (acc = 0)

for example:

Send FA 01 F6 00 00 00 F1

Stop the motor immediately

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

Uplink package（PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package” can be disable by Menu “UartRSP” or Command “8C”.

3. Save/Clean the parameter in speed mode

Downlink package（PC → SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FA	01	FF	state	CRC

state = C8 Save.
state = CA Clean.

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	FF	status(uint8_t)	CRC

status = 1 start.
status = 0 fail.
status = 2 success.

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

6.6 Position mode1: relative motion by pulses

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

1. Run the motor in position mode1

Downlink package（PC → SERVO42D/57D）									
BYTE1	BYTE2	BYTE3	BYTE4			BYTE5	BYTE6	BYTE7-10	BYTE11
Head	Slave addr	Function	dir	Rev	speed		acc	pulses	CRC
FA	addr	FD	b7	b6-b4	b3-b0	b7-b0	acc	pulses	CRC
			dir	--	speed				

Byte 4: The highest bit indicates the direction, the lower 4 bits and byte 5 together indicate the speed

Byte 5: The lower 4 bits of byte 5 and byte 4 together indicate speed

The parameter description is as follows:

addr: slave address, the value range is 0-255

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

speed: the speed, the value range is 0-3000 (RPM)

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

pulses: the motor run steps, the value range is 0 – 0xFFFFFFFF

for example:

Send FA 01 FD 01 40 02 00 00 FA 00 35,

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

Send FA 01 FD 81 40 02 00 00 FA 00 b5,

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

Uplink package（PC ← SERVO42D/57D）				
Head	Slave addr	Function	Data	CRC
FB	01	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 package” can be disable by Menu “UartRSP” or Command “8C”.

2. Stop the motor in position mode1

Downlink package（PC → SERVO42D/57D)									
BYTE1	BYTE2	BYTE3	BYTE4			BYTE5	BYTE6	BYTE7-10	BYTE11
Head	Slave addr	Function	dir	Rev	speed		acc	pulses	CRC
FA	addr	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 FA 01 FD 00 00 02 00 00 00 00 FA

Stop the motor with deceleration acc=2

② Immediate stop command (acc = 0)

for example:

Send FA 01 FD 00 00 00 00 00 00 00 F8

Stop the motor immediately

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

Uplink package（PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	FD	status(uint8_t)	CRC

status = 0 stop the motor fail.

status = 1 stop the motor starting….

status = 2 stop the motor complete.

Note1: the “Uplink package” can be disable by Menu “UartRSP” or Command “8C”.

6.7 Position mode2: absolute motion by pulses

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

1. Run the motor in position mode2

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	absolute axis	Check
FA	addr	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, int32_t

For example:
If the current axis is any value
Send FA 01 FE 02 58 02 00 00 40 00 95
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 FA 01 FE 02 58 02 FF FF C0 00 13
The motor will move to -0x4000 (speed = 600(RPM),acc =2)
After move the pulses is -0x4000.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package” can be disable by Menu “UartRSP” or Command “8C”.

2. Stop the motor in position mode2

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	absPulses	Check
FA	addr	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 FA 01 FE 00 00 04 00 00 00 00 FD

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send FA 01 FE 00 00 00 00 00 00 00 F9

Stop the motor immediately

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	FE	status(uint8_t)	CRC

status = 0 stop fail.

status = 1 stop starting….

status = 2 stop complete.

status = 3 end limit stoped.

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

6.8 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 mode2

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	Relative axis	Check
FA	addr	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, int32_t

For example:
If the current axis is 0x8000.(read by code “31”)
Send FA 01 F4 02 58 02 00 00 40 00 8B
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 FA 01 F4 02 58 02 FF FF C0 00 03
The motor will relative move -0x4000 (speed = 600(RPM),acc =2)
After move the axis is 0x4000.(0x8000-0x4000=0x4000)

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package” can be disable by Menu “UartRSP” or Command “8C”.

2. Stop the motor in position mode3

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	Relative axis	Check
FA	addr	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 FA 01 F4 00 00 04 00 00 00 00 F3

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send FA 01 F4 00 00 00 00 00 00 00 EF

Stop the motor immediately

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	F4	status(uint8_t)	CRC

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

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

6.9 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

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	absolute axis	Check
FA	addr	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
relAxis: the absolute axis, int32_t

For example:
If the current axis is any value
Send FA 01 F5 02 58 02 00 00 40 00 8C
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 FA 01 F5 02 58 02 FF FF C0 00 0A
The motor will move to -0x4000 (speed = 600(RPM),acc =2)
After move the axis is -0x4000.

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	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 package” can be disable by Menu “UartRSP” or Command “8C”.

2. Stop the motor in position mode4

byte1	byte2	byte3	byte 4-5	byte 6	byte 7-10	byte 11
Head	Slave addr	Function	speed	acc	absolute axis	Check
FA	addr	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 FA 01 F5 00 00 04 00 00 00 00 F4

Stop the motor with deceleration acc=4

② Immediate stop command (acc = 0)

for example:

Send FA 01 F5 00 00 00 00 00 00 00 F0

Stop the motor immediately

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

Uplink package (PC ← SERVO42D/57D)				
Head	Slave addr	Function	Data	CRC
FB	01	F5	status(uint8_t)	CRC

status = 0　stop fail.

status = 1　stop starting….

status = 2　stop complete.

status = 3　end limit stoped.

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

Part7. Serial example

Note: For MODBUS-RTU example, see Part 8.

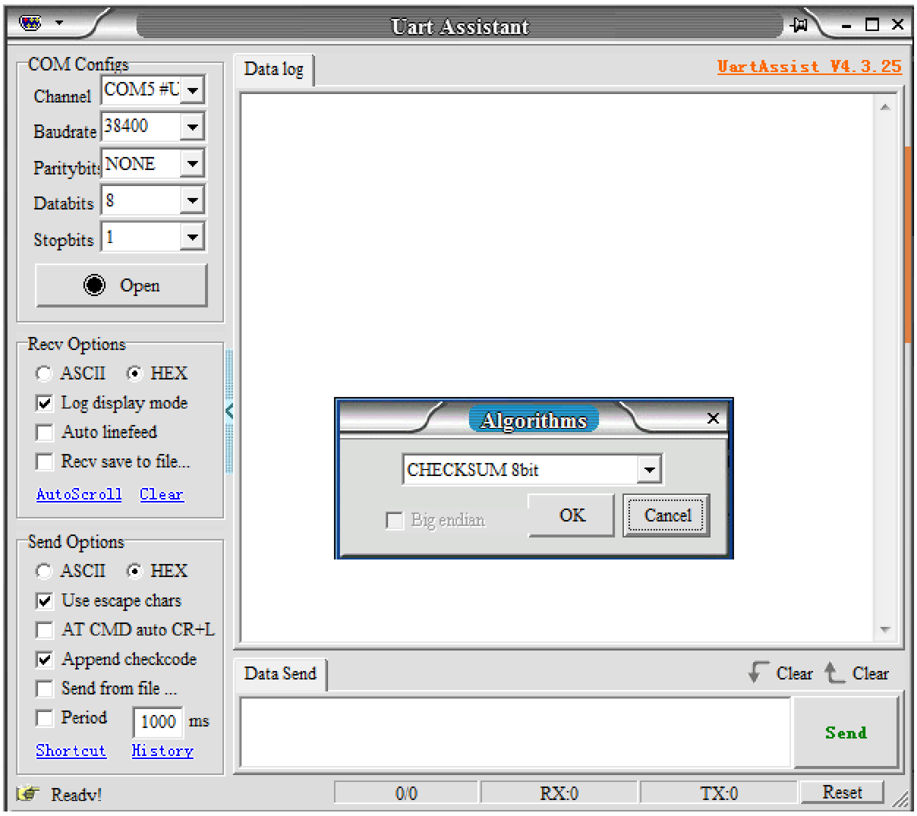
7.1 Config the SERVO42D/57D

1. Menu → Mode → SR_vFOC.
2. Menu → UartBaud → 38400.
3. Menu → UartAddr → 01.

7.2 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:



COM Configs
Channel: COM5 #U
Baudrate: 38400
Paritybits: NONE
Databits: 8
Stopbits: 1

Recv Options: ○ ASCII ● HEX; ☑ Log display mode; □ Auto linefeed; □ Recv save to file...

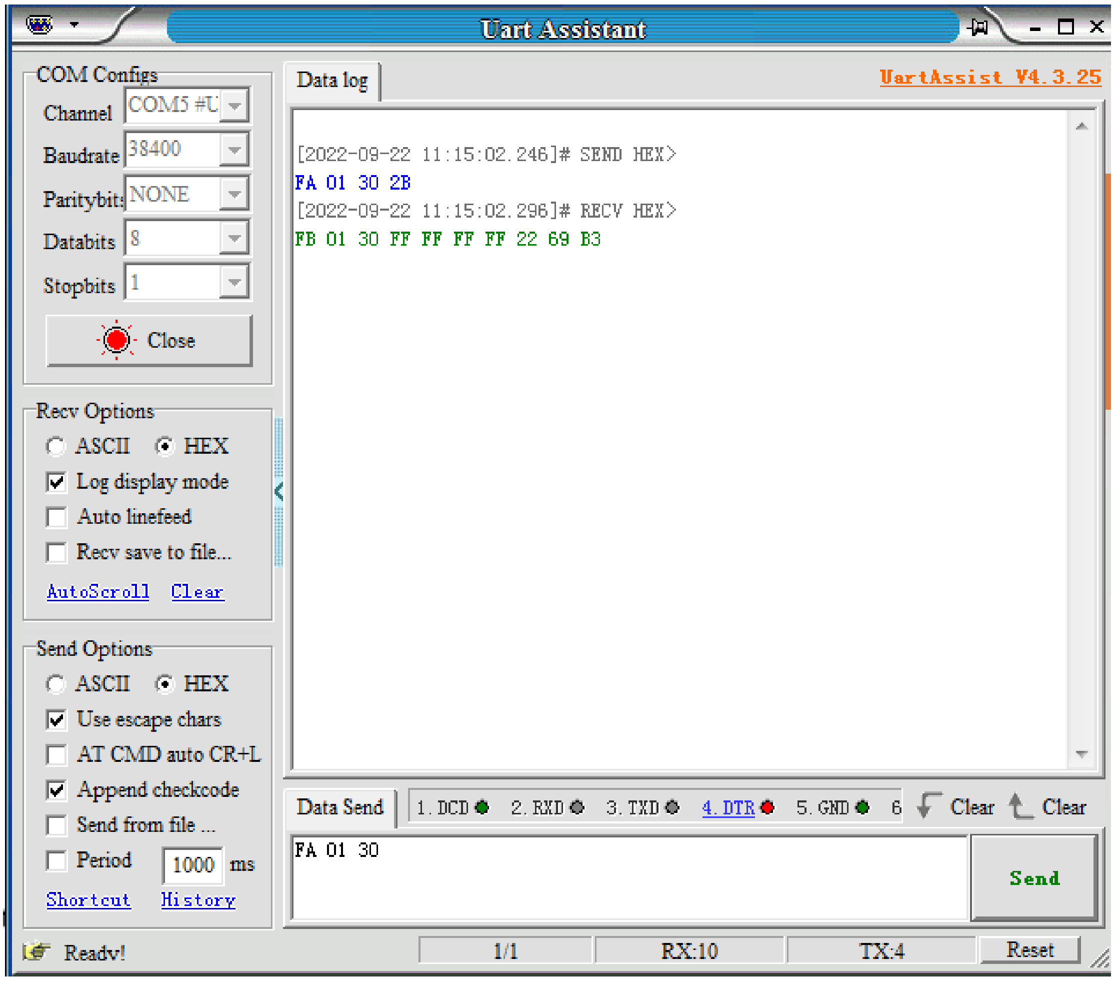
Send Options: ○ ASCII ● HEX; ☑ Use escape chars; □ AT CMD auto CR+L; ☑ Append checkcode; □ Send from file ...; □ Period 1000 ms

Algorithms: CHECKSUM 8bit; □ Big endian

7.3 Read the encoder value

send “FA 01 30 2B”

return “FB 01 30 FF FF FF FF 22 69 B3”



Channel COM5 #U
Baudrate 38400
Paritybits NONE
Databits 8
Stopbits 1

○ ASCII ● HEX
☑ Log display mode
□ Auto linefeed
□ Recv save to file...

○ ASCII ● HEX
☑ Use escape chars
□ AT CMD auto CR+L
☑ Append checkcode
□ Send from file ...
□ Period 1000 ms

```
[2022-09-22 11:15:02.246]# SEND HEX>
FA 01 30 2B
[2022-09-22 11:15:02.296]# RECV HEX>
FB 01 30 FF FF FF FF 22 69 B3
```

```
FA 01 30
```

Ready! 1/1 RX:10 TX:4

7.4 Run the motor in speed mode

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

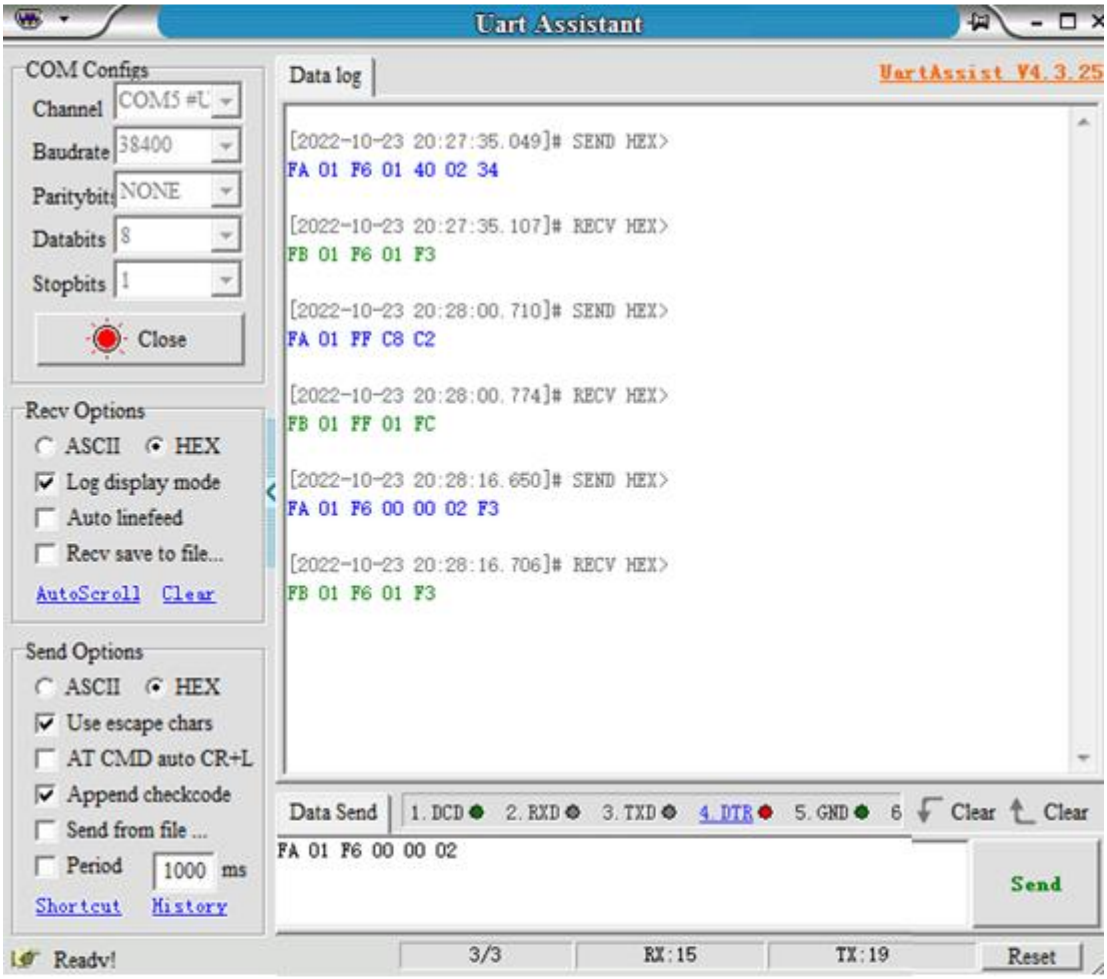
1. Send FA 01 F6 01 40 02 , the motor will rotate at "speed = 320RPM, acc=2";
Return FB 01 F6 01 F3, the motor run in speed mode successful;

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

3. Send FA 01 F6 00 00 02 to stop the motor;
Return FB 01 F6 01 F3, the motor stops successfully;

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

The example command of speed mode is shown in the following figure:



Channel COM5 #U
Baudrate 38400
Paritybits NONE
Databits 8
Stopbits 1

○ ASCII ● HEX
☑ Log display mode
□ Auto linefeed
□ Recv save to file...

○ ASCII ● HEX
☑ Use escape chars
□ AT CMD auto CR+L
☑ Append checkcode
□ Send from file ...
□ Period 1000 ms

```
[2022-10-23 20:27:35.049]# SEND HEX>
FA 01 F6 01 40 02 34

[2022-10-23 20:27:35.107]# RECV HEX>
FB 01 F6 01 F3

[2022-10-23 20:28:00.710]# SEND HEX>
FA 01 FF C8 C2

[2022-10-23 20:28:00.774]# RECV HEX>
FB 01 FF 01 FC

[2022-10-23 20:28:16.650]# SEND HEX>
FA 01 F6 00 00 02 F3

[2022-10-23 20:28:16.706]# RECV HEX>
FB 01 F6 01 F3
```

FA 01 F6 00 00 02

3/3 RX:15 TX:19

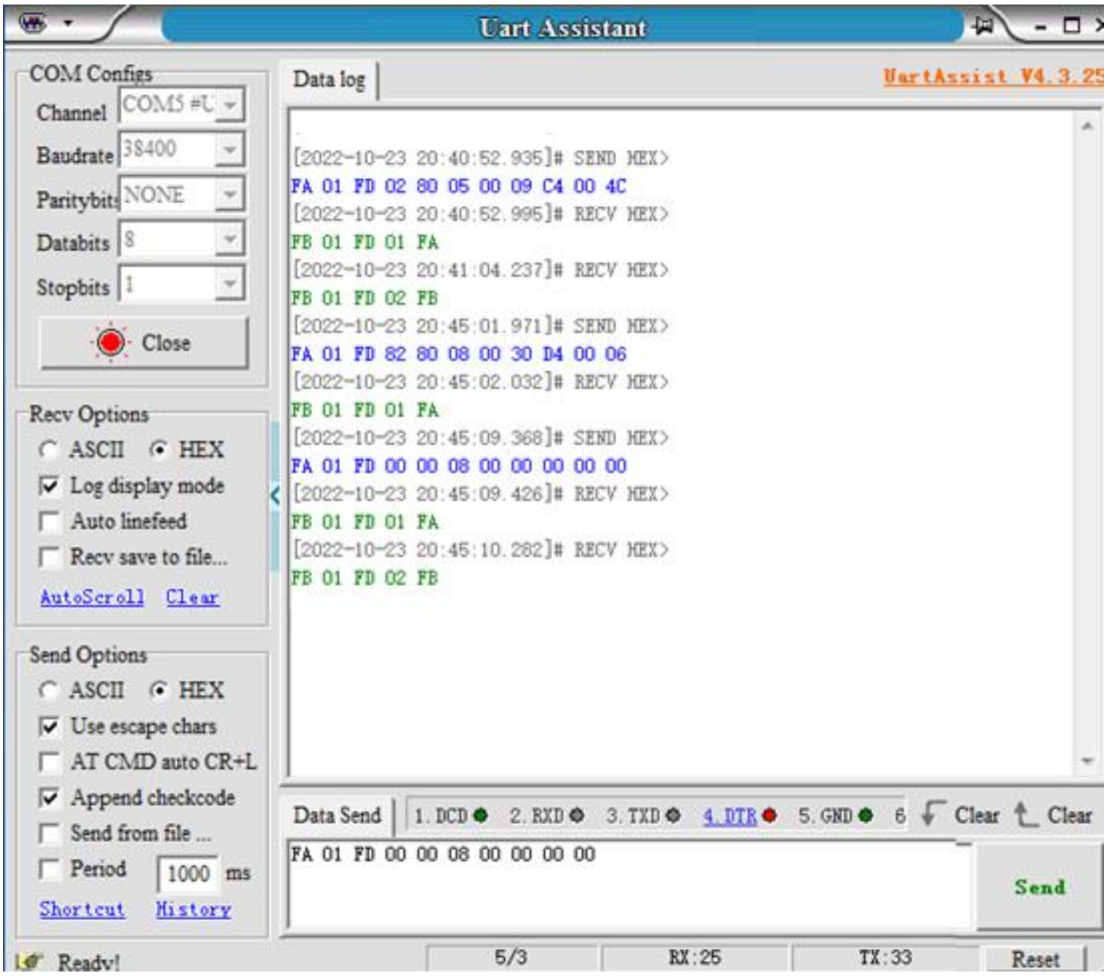
7.5 Run the motor in position model

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

1. Send FA 01 FD 02 80 05 00 09 C4 00, the motor will rotate forward 200 circles (16 subdivisions) with "speed = 640RPM, acc = 5";
Return FB 01 FD 01 FA, the motor starts to run;
Return FB 01 FD 02 FB, the motor is run completed;

2. Send FA 01 FD 82 80 08 00 30 D4 00, the motor to reverse 1000 circles with "speed = 640RPM, acc = 8" (16 subdivisions);
Return FB 01 FD 01 FA, the motor starts to run;
While the motor is running:
Send FA 01 FD 00 00 08 00 00 00 00, the motor to stop with acc=8;
Return FB 01 FD 01 FA, the motor starting to stop;
Return FB 01 FD 02 FB, the motor has stopped;

The example command of position control mode is shown in the following figure:



Channel COM5 #U
Baudrate 38400
Paritybits NONE
Databits 8
Stopbits 1

○ ASCII ● HEX
☑ Log display mode
□ Auto linefeed
□ Recv save to file...

○ ASCII ● HEX
☑ Use escape chars
□ AT CMD auto CR+L
☑ Append checkcode
□ Send from file ...
□ Period 1000 ms

```
[2022-10-23 20:40:52.935]# SEND HEX>
FA 01 FD 02 80 05 00 09 C4 00 4C
[2022-10-23 20:40:52.995]# RECV HEX>
FB 01 FD 01 FA
[2022-10-23 20:41:04.237]# RECV HEX>
FB 01 FD 02 FB
[2022-10-23 20:45:01.971]# SEND HEX>
FA 01 FD 82 80 08 00 30 D4 00 06
[2022-10-23 20:45:02.032]# RECV HEX>
FB 01 FD 01 FA
[2022-10-23 20:45:09.368]# SEND HEX>
FA 01 FD 00 00 08 00 00 00 00 00
[2022-10-23 20:45:09.426]# RECV HEX>
FB 01 FD 01 FA
[2022-10-23 20:45:10.282]# RECV HEX>
FB 01 FD 02 FB
```

```
FA 01 FD 00 00 08 00 00 00 00
```

5/3 RX:25 TX:33 Reset

Part8. MODBUS-RTU command description

Note: It need to enable MODBUS-RTU by menu or serial command.

8.1 Read parameter command

1. Read the encoder value(carry)

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	30H	00H	03H	B0H	04H

Response						
SlaveAddr	Function	Bytes	DATA		CRC16	
			carry	value	Hi	Lo
01H	04H	06H	int32_t	uint16_t		

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.

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

See "00_F4(30)read encoder value (carry).mbp" for example.

2. Read the encoder value(addition)

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	31H	00H	03H	E1H	C4H

Response					
SlaveAddr	Function	Bytes	value	CRC16	
				Hi	Lo
01H	04H	06H	(int48_t)		

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.

See “01_F4(31)read encoder value(addition).mbp” for example.

3. Read the real-time speed of the motor

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	32H	00H	01H	E1H	C4H

Response					
SlaveAddr	Function	Bytes	speed	CRC16	
				Hi	Lo
01H	04H	02H	(int16_t)		

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

if it run CW, the speed < 0 (RPM)

See “02_F4(32)Read the real-time speed.mbp” for example.

4. Read the number of pulses

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	33H	00H	02H	81H	C4H

Response					
SlaveAddr	Function	Bytes	pulses	CRC16	
				Hi	Lo
01H	04H	04H	(uint32_t)		

See “03_F4(33)Read the number of pulses.mbp” for example.

5. Read the IO Ports status

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	34H	00H	01H	70H	04H

Response						
SlaveAddr	Function	Bytes	Reserved	status	CRC16	
					Hi	Lo
01H	04H	02H	00H	(uint8_t)		

status							
Bit7	…	…	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. Read the error of angle

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	39H	00H	02H	A1H	C6H

Response					
SlaveAddr	Function	Bytes	errors	CRC16	
				Hi	Lo
01H	04H	04H	(int32_t)		

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.

See “04_F4(39)Read the error of angle.mbp” for example.

7. Read the En pins status

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	3AH	00H	01H	11H	C7H

Response						
SlaveAddr	Function	Bytes	Reserved	enable	CRC16	
					Hi	Lo
01H	04H	02H	00H	(uint8_t)		

enable =1 Enabled

enable =0 Disabled

See “05_F4(3A)Read the En pins status.mbp” for example.

8. Read the go back to zero status

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	3BH	00H	01H	40H	07H

Response						
SlaveAddr	Function	Bytes	Reserved	status	CRC16	
					Hi	Lo
01H	04H	02H	00H	(uint8_t)		

status =0 going to zero.

status =1 go back to zero success.

status =2 go back to zero fail.

See “06_F4(3B)Read the go back to zero status.mbp” for example.

9. Read the motor shaft protection status

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	3EH	00H	01H	50H	06H

Response						
SlaveAddr	Function	Bytes	Reserved	status	CRC16	
					Hi	Lo
01H	04H	02H	00H	(uint8_t)		

status =1 protected.

status =0 no protected.

See “07_F4(3E)Read the motor shaft protection status.mbp” for example.

10. Read the motor status

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	F1H	00H	01H	60H	39H

Response						
SlaveAddr	Function	Bytes	Reserved	status	CRC16	
					Hi	Lo
01H	04H	02H	00H	(uint8_t)		

status = 0 read 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…

See “09_F4(F1)Read the motor status.mbp” for example.

8.2 Write parameter command

1. Release the motor shaft locked-rotor protection status

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	3DH	00H	01H	D9H	C6H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	3DH	00H	01H	D9H	C6H

See “09_F4(F1)Read the motor status.mbp” for example.

2. Restore the default parameter

(Same as the "Restore" option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	3FH	00H	01H	78H	06H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	3FH	00H	01H	78H	06H

See “11_F6(3F)Restore the default parameter.mbp” for example.

3. Restart the motor

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	41H	00H	01H	18H	1EH

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	41H	00H	01H	18H	1EH

4. Calibrate the motor

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	80H	00H	01H	49H	E2H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	80H	00H	01H	49H	E2H

See “12_F6(80)Calibrate the motor.mbp” for example.

Note : The motor must be unloaded.

5. Set the work mode

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	82H	00H	mode		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	82H	00H	mode		

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

mode = 4 SR_CLOSE
mode = 5 SR_vFOC
See “13_F6(82)Set the work mode.mbp” for example.

6. Set the work current

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	83H	Current			

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	83H	Current			

Note:the new current will show in the screen of Ma option.
SERVO42D/28D/35D: Maximum Current =3000mA
SERVO57D: Maximum Current =5200mA
See “14_F6(83)Set the work current.mbp” for example.

7. Set the holding current percentage

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	9BH	00H	holdMa		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	9BH	00H	holdMa		

holdMa = 00 10%
holdMa = 01 20%
…
holdMa = 08 90%
Note: Only for OPEN mode and CLOSE mode, vFOC mode is invalid.

8. Set subdivision

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	84H	micstep			

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	84H	micstep			

See “15_F6(84)Set subdivision.mbp” for example.

9. Set the active of the En pin

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	85H	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	85H	00H	enable		

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

See “16_F6(85)Set the active of the En pin.mbp” for example.

10. Set the direction of motor rotation

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	86H	00H	dir		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	86H	00H	dir		

dir = 00 CW
dir = 01 CCW

See “17_F6(86)Set the direction of motor rotation.mbp” for example.

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

11. Set auto turn off the screen function

(Same as the ″ **AutoSDD** ″ option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	87H	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	87H	00H	enable		

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.

See “18_F6(87)Set auto turn off the screen.mbp” for example.

12. Set the motor shaft locked-rotor protection

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

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	88H	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	88H	00H	enable		

enable = 01 enabled protection

enable = 00 disabled protection

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

See “19_F6(88)Set the motor locked-rotor protection.mbp” for example.

13. Set the subdivision interpolation function

(Same as the ″ **MPlyer** ″ option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	89H	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	89H	00H	enable		

enable = 01   enabled interpolation function.

enable = 00   disabled interpolation function.

See “20_F6(89)Set the subdivision interpolation.mbp” for example.

14. Set the baud rate

(Same as the ″ **UartBaud** ″ option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8AH	00H	baud		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8AH	00H	baud		

baud = 01	9600.
baud = 02	19200.
baud = 03	25000.
baud = 04	38400.
baud = 05	57600.
baud = 06	115200.
baud = 07	256000.

See “21_F6(8A)Set the baud rate.mbp” for example.

15. Set the slave address

(Same as the ″ **UartAddr** ″ option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8BH	00H	addr		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8BH	00H	addr		

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

Note2: 0 is the broadcast address

See “22_F6(8B)Set the slave address.mbp” for example.

16. Set MODBUS-RTU communication protocol

(Same as the ″ **MB_RTU** ″ option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8EH	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8EH	00H	enable		

enable = 01 enabled MODBUS-RTU communication protocol.

enable = 00 disabled MODBUS-RTU communication protocol.

See “23_F6(8E)Set MODBUS-RTU.mbp” for example.

17. Set the key locked or unlocked

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8FH	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	8FH	00H	enable		

enable = 01 locked the key.

enable = 00 unlocked the key.

18. Set Currnet Axis to zero

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	92H	00H	01H	E9H	E7H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	92H	00H	01H	E9H	E7H

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

See “25_F6(92) Set Currnet Axis to zero.mbp” for example.

19. Set serial mode motor enable

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	F3H	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	F3H	00H	enable		

enable = 01 enabled the motor.

enable = 00 disabled the motor.

See “26_F6(F3)Set serial mode motor enable.mbp” for example.

Note : This command is only for “SR_OPEN/SR_CLOSE/SR_VFOC” mode

20. Set the parameter of home

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

Request													
Slave addr	Function	Starting Address		Quantity of Registers		Bytes	Trig level	Home dir	Home speed		enable	CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo		Hi	Lo
01H	10H	00H	90H	00H	03H	05H	hmTrig	hmDir	HmSpeed		EndLimit		

hmTrig the effective level of the end stop

0: Low

1: High

hmDir the direction of go home

0: CW

1: CCW

hmSpeed the speed of go home

0~3000 (RPM)

EndLimit

0: disable endstop-limit

1: enable endstop-limit

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

Response							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	90H	00H	03H	80H	25H

See “28_F16(90)Set the parameter of home.mbp” for example.

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.

21. 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.

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	9EH	00H	enable		

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	9EH	00H	enable		

enable = 01 enable remap limit port

enable = 00 disable remap limit port

22. Set the parameter of 0_Mode

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

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

Request												
Slave addr	Function	Starting Address		Quantity of Registers		Bytes	0_Mode	Set 0	0_Speed	0_Dir	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10 H	00H	9AH	00H	02H	04H	mode	enable	speed	dir		

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

speed:

0 ~ 4 (0:slowest 4:fastest)

dir:

0: CW

1: CCW

Response							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	9AH	00H	02H	61H	E7H

8.3 Motor running command

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

8.3.1 Emergency stop the motor

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	F7H	00H	01H	F9H	F8H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	F7H	00H	01H	F9H	F8H

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

8.3.2 Go home

(Same as the “**GoHome**” option on screen)

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	91H	00H	01H	19H	E7H

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	91H	00H	01H	19H	E7H

See “24_F6(91) Go Home.mbp” for example.

Note: 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.

8.3.3 Speed mode command

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

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

1. Run the motor in speed mode

Request												
Slave Addr	Function	Starting Address		Quantity of Registers		Bytes	direction	acceleration	speed		CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	04H	dir	acc	speed			

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

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

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

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	A1H	FAH

See “29_F16(F6) Run the motor in speed mode.mbp” for example.

2. Stop the motor in speed mode

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

Request												
Slave Addr	Function	Starting Address		Quantity of Registers		Bytes	direction	acceleration	speed		CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	04H	00H	acc	00H			

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F6H	00H	02H	A1H	FAH

See “29_F16(F6) Run the motor in speed mode.mbp” for example.

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

3. Save/Clean the parameter in speed mode

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	FFH	00H	flag		

flag = C8H   save the parameter
flag = CAH   clean the parameter

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	FFH	00H	flag		

See “27_F6(FF) Save or Clean the parameter in speed mode.mbp” for example.

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

8.3.4 Position mode1: relative motion by pulses

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

1. Run the motor in position mode1

Request												
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	direction	acceleration	speed	pulses	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	FDH	00H	04H	08H	dir	acc	speed	pulses		

dir	(uint8_t)	the value range is 0/1 （CCW/CW）
acc	(uint8_t)	the acceleration, the value range is 0 - 255
speed	(uint16_t)	the speed, the value range is 0 – 3000 (RPM)
pulses	(uint32_t)	the steps, the value range is 0 – 0xFFFFFFFF

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	FDH	00H	04H	50H	3AH

See “30_F16(FD) Position mode1 relative motion by pulses.mbp” for example.

2. Stop the motor in position mode1

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

Request												
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	direction	acceleration	speed	pulses	CRC16	
		Hi	Lo	Hi	Lo						Hi	Lo
01H	10H	00H	FDH	00H	04H	08H	00H	acc	00H	00H		

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	FDH	00H	04H	50H	3AH

See “30_F16(FD) Position mode1 relative motion by pulses.mbp” for example.

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

8.3.5 Position mode2: absolute motion by pulses

Note1: It needs to set the working mode to serial mode.
(SR_OPEN/SR_CLOSE/SR_VFOC)

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

1. Run the motor in position mode2

Request											
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	absolute axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	FEH	00H	04H	08H	acc	speed	absPulses		

acc (uint16_t) the acceleration, the value range is 0 - 255
speed (uint16_t) the speed, the value range is 0 - 3000 (RPM)
absPulses(int32_t) the Pulses, int32_t

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	FEH	00H	04H		

2. Stop the motor in position mode2

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

Request											
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	absolute axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	FEH	00H	04H	08H	acc	00H	00H		

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	FEH	00H	04H		

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

8.3.6 Position mode3: relative motion by axis

Note1: It needs to set the working mode to serial mode.
(SR_OPEN/SR_CLOSE/SR_VFOC)

Note2: the axis is the encoder value(addition).It can be read by command “31”.

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

1. Run the motor in position mode3

Request											
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	Relative axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F4H	00H	04H	08H	acc	speed	relAxis		

acc (uint16_t) the acceleration, the value range is 0 - 255
speed (uint16_t) the speed, the value range is 0 - 3000 (RPM)
relAxis(int32_t) the steps, int32_t

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F4H	00H	04H	80H	38H

See “31_F16(F4) Position mode2 relative motion by axis.mbp” for example.

2. Stop the motor in position mode3

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

Request											
SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	Relative axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F4H	00H	04H	08H	acc	00H	00H		

Response							
SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F4H	00H	04H	80H	38H

See “31_F16(F4) Position mode2 relative motion by axis.mbp” for example.

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

8.3.7 Position mode4: absolute motion by axis

Note1: It needs to set the working mode to serial mode.
(SR_OPEN/SR_CLOSE/SR_VFOC)

Note2: the axis is the encoder value(addition). It can be read by command “31”.

Note3: 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

Request

SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	absolute axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F5H	00H	04H	08H	acc	speed	absAxis		

acc (uint16_t) the acceleration, the value range is 0 - 255
speed (uint16_t) the speed, the value range is 0 - 3000 (RPM)
absAxis(int32_t) the steps, int32_t

Response

SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F5H	00H	04H	D1H	F8H

See “32_F16(F5) Position mode3 absolute motion by axis.mbp” for example.

2. Stop the motor in position mode4

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

Request

SlaveAddr	Function	Starting Address		Quantity of Registers		Bytes	acceleration	speed	absolute axis	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	F5H	00H	04H	08H	acc	00H	00H		

Response

SlaveAddr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	F5H	00H	04H	D1H	F8H

See “32_F16(F5) Position mode3 absolute motion by axis.mbp” for example.

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

Part9. FAQ

9.1 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.

9.2 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.

Part10. Schematic

Please download 《MKS SERVO42D/57D V1.0 Schematic.pdf》 in
https://github.com/makerbase-motor/MKS-SERVO42D
https://github.com/makerbase-motor/MKS-SERVO57D

Part11. contact us

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