MKS SERVO42E/57E_RS485 V1.0.1

USER MANUAL

MKS SERVO42E/57E_RS485 Manual Release			
manual	discription	firmware	date
V1.0.0	First release	V1.0.0	Dec-2024
V1.0.1	1. Fix some bug; 2. Add IAP firmware upgrade function;	V1.0.1	Feb-2025

Part1. Product Overview

1.1 Introduction

MKS SERVO42E/57E_RS485 closed-loop stepper motor is a product independently developed by the Maker Base to meet market demand and in accordance with industrial standards. It has a pulse interface and an RS485 interface, a built-in efficient FOC vector algorithm, and a high-precision encoder. Through position feedback, it effectively prevents the motor from losing steps. It is suitable for small robotic arms, 3D printers, engraving machines, writers, automation products, and e-sports applications.

1.2 Features

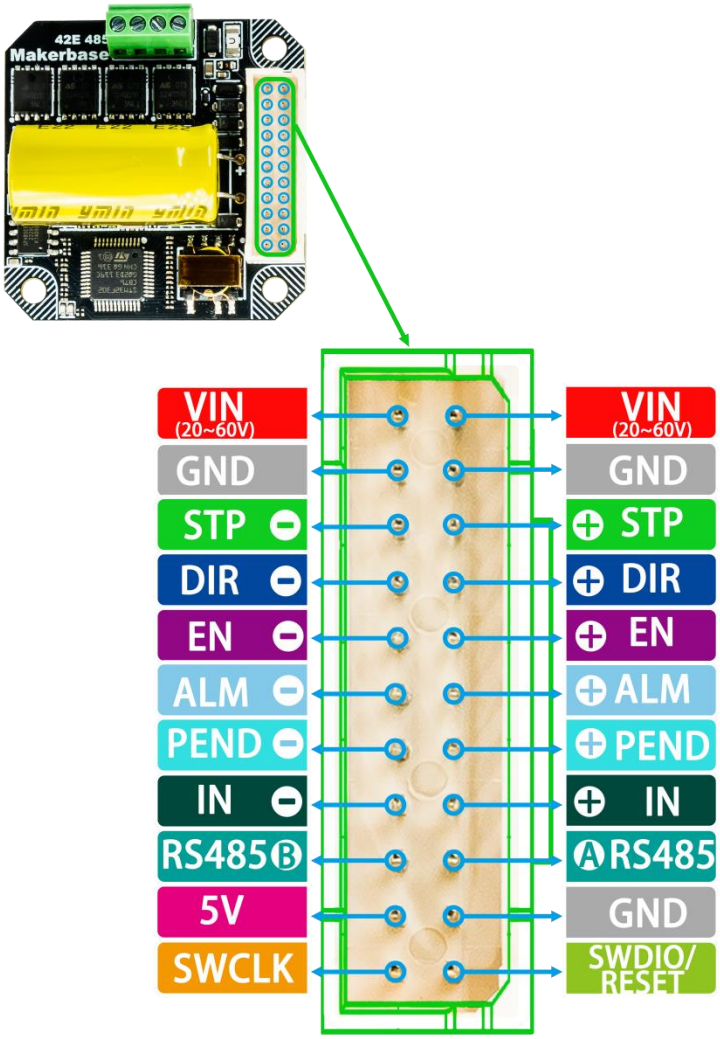
1. Support 6 working modes: pulse interface (pulse-pulse, pulse-direction mode), serial interface (open loop, Closed-loop mode);
2. High-performance FOC vector control algorithm, torque, speed, position three-loop control;
3. Support curve acceleration and deceleration, motor start and stop more smoothly;
4. Support relative position and absolute position control mode;
5. Built-in 256-step subdivision interpolation algorithm, motor operation is ultra-quiet and ultra-low vibration;
6. Maximum input pulse frequency 300KHz, maximum speed 3000RPM;
7. Motor angle information is updated in real time (motor enabled or disabled);
8. Support MODBUS-RTU communication protocol;
9. Built-in stall protection function;
10. Quickly restore factory configuration function;
11. High-speed performance is stable, running smoothly, no jitter, and emergency stop is possible;
12. Integrated aluminum alloy shell, effective heat dissipation, motor continuous high current operation is more stable;
13. Provide host computer (open source), STM32/Arduion usage routines
14. Support left and right limit functions.
15. Onboard industrial-grade high-precision magnetic encoder;
16. Onboard high-power MOSFET, 100V/25A;
17. Onboard RS-485 interface, 256 slave addresses, support broadcast address and group address;
18. Industrial-grade selection design, stable and reliable;
19. Working voltage 20V~60V;

1.3 Parameters

Parameters		
Motherboard model	MKS SERVO42E V1.0	MKS SERVO57E V1.0
MCU	STM32F302RBT6Cortex-M4)	STM32F302RBT6Cortex-M4)
MOSFET	ASS240D10Q1M	
Magnetic encoder	MT6816	
RS485 transceiver	SP485EEN	
Operating Voltage	20-60V	
Working current	0-3000mA	0-5200mA
Closed loop feedback frequency	Torque loop 20KHz	
	Speedloop 10KHz	
	Position loop 10KHz	
Maximum speed	3000RPM+	
Segmentation support	2/4/8/16/32/64/128/5/10/20/25/40/50/100/200/256	
interface Wiring Type	Common anode, common cathode, differential, double pulse, pulse + direction	
Input interface voltage	3.3V-24V	
Max Pulse frequency	300KHz	
RS485 Interface rate	9600/19200/…/115200/25600	
RS485 Interface address	1 broadcast address, 255 slave addresses	

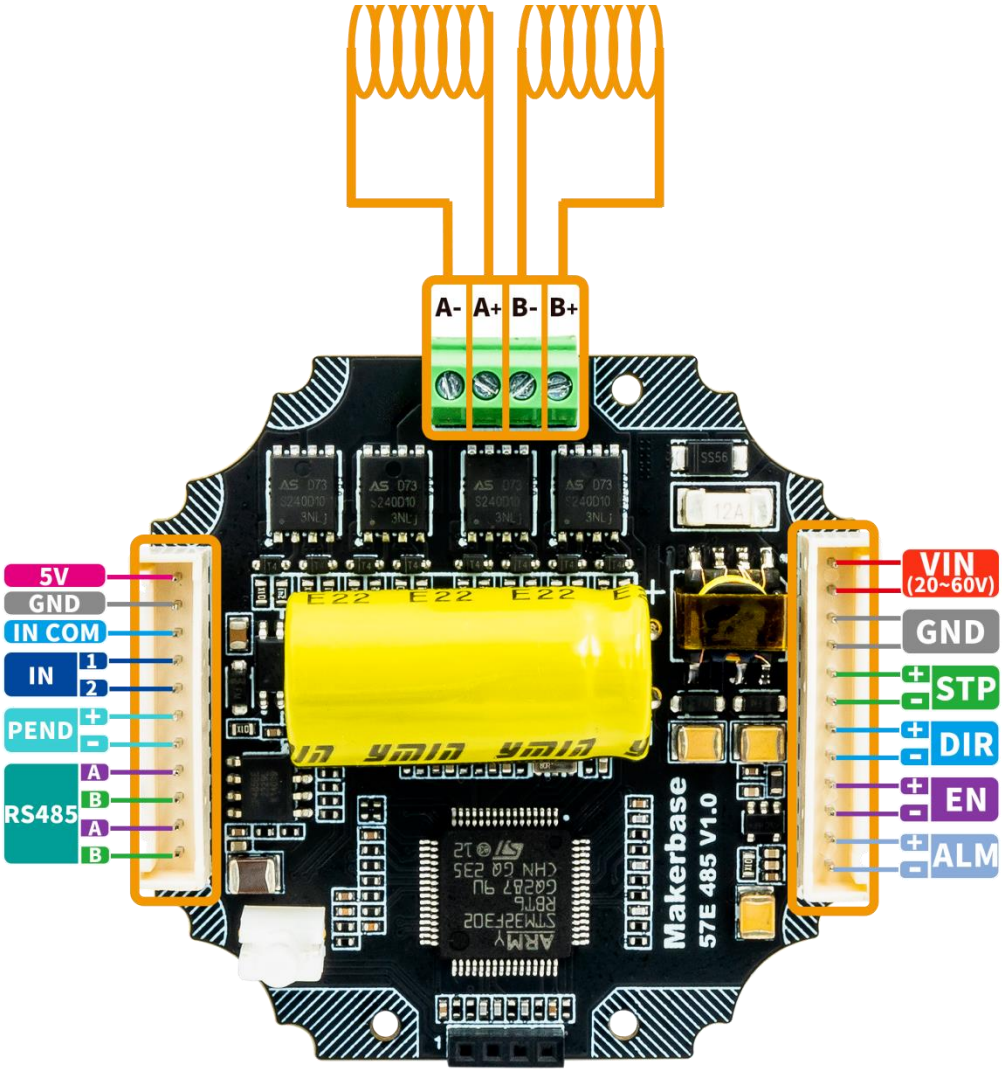
1.4 Interface Description

① Description of the SERVO42D_RS485 interface



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

② Description of the SERVO57D_RS485 interface



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

1.5 Port funktionsbeskrivelse

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

1.6 Status LED

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

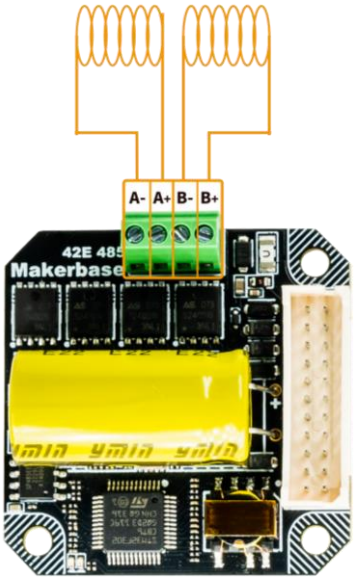
Tip: Please make sure the motor phase line is connected correctly, otherwise there will be a tracking error alarm after receiving the pulse (5 red and 1 green)

Part2. Motor wire

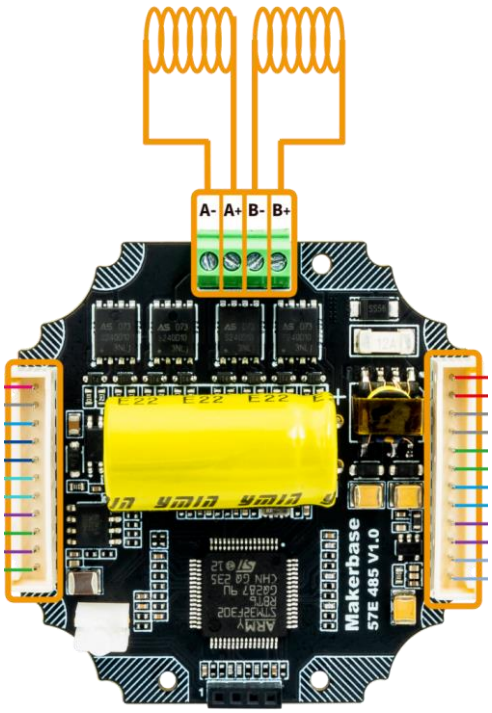
Note1: The motor internal resistance should be less than 10 ohms.

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

① Connection method of SERVO42E_RS485



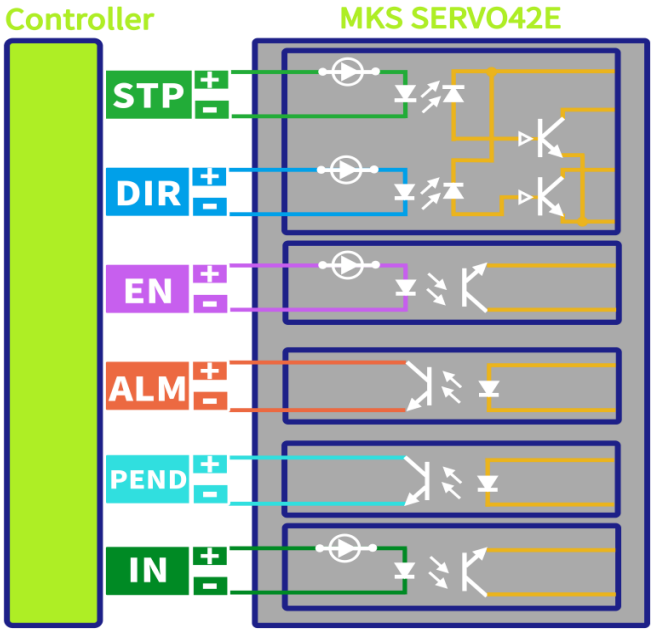
② Connection method of SERVO57E_RS485



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

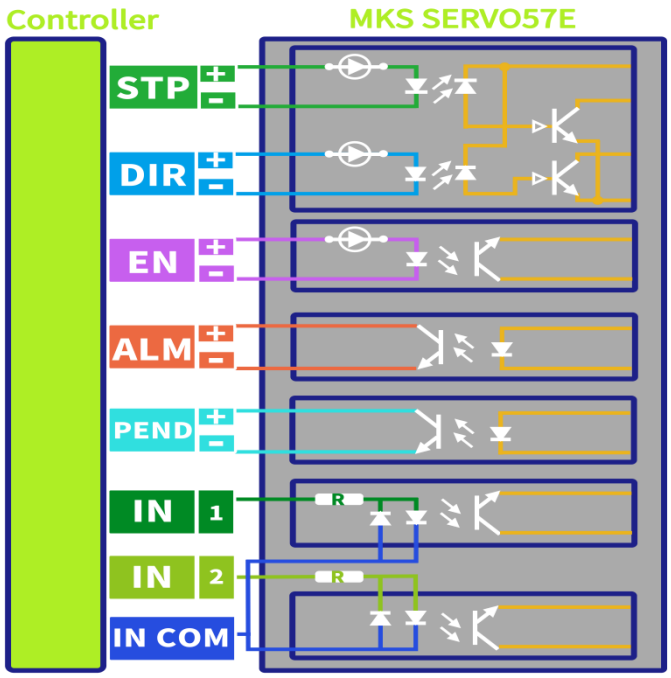
2.1 Pulse interface wire

① Connection method of SERVO42E_RS485



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

② Connection method of SERVO57E_RS485



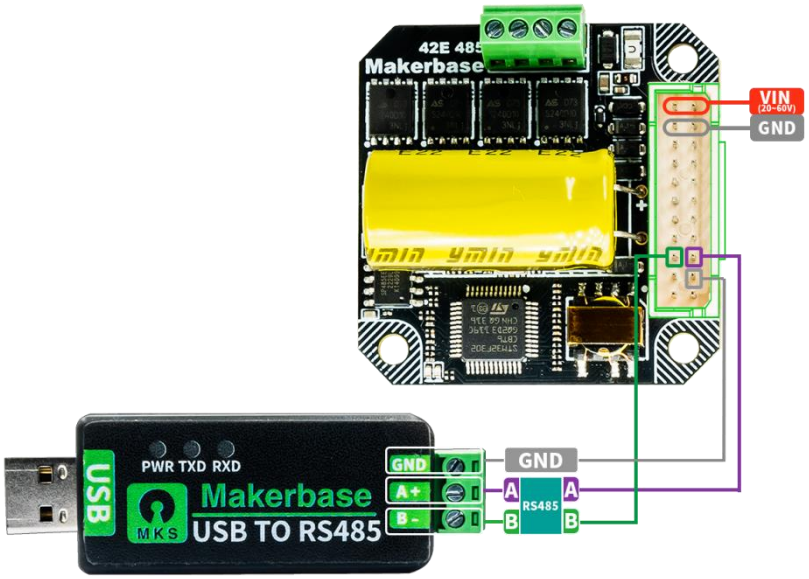
Note: All input ports have built-in 10mA current limiting, and can directly input 3.3V-24V signals without the need for external current limiting resistors

2.2 RS485 wire

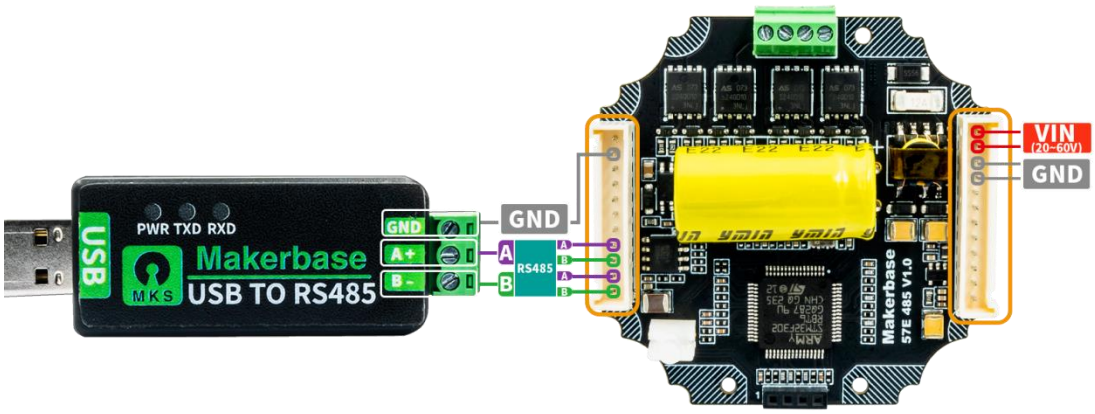
1. SERVO57E_RS485 Single-slave

NOTE: 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 Connect cables to the MKS SERVO42E RS485 single machine

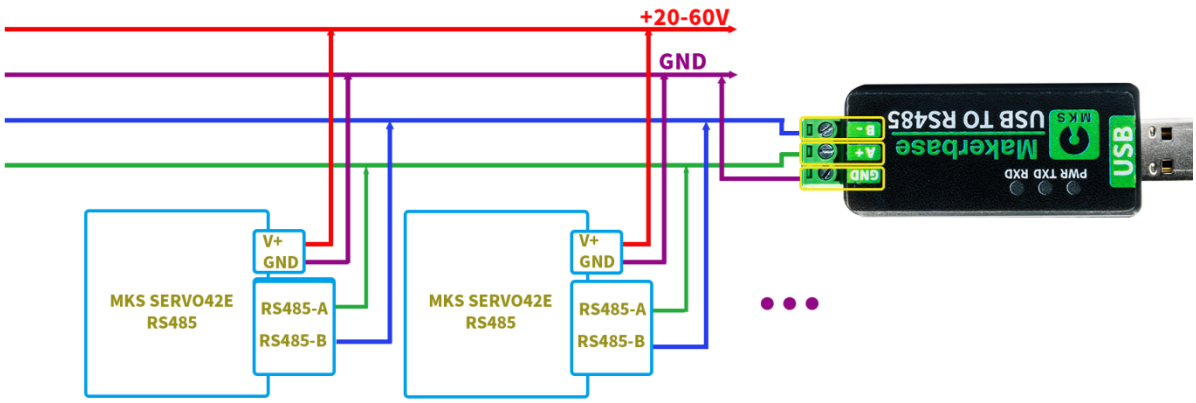


② Connect cables to the MKS SERVO57E RS485 single machine

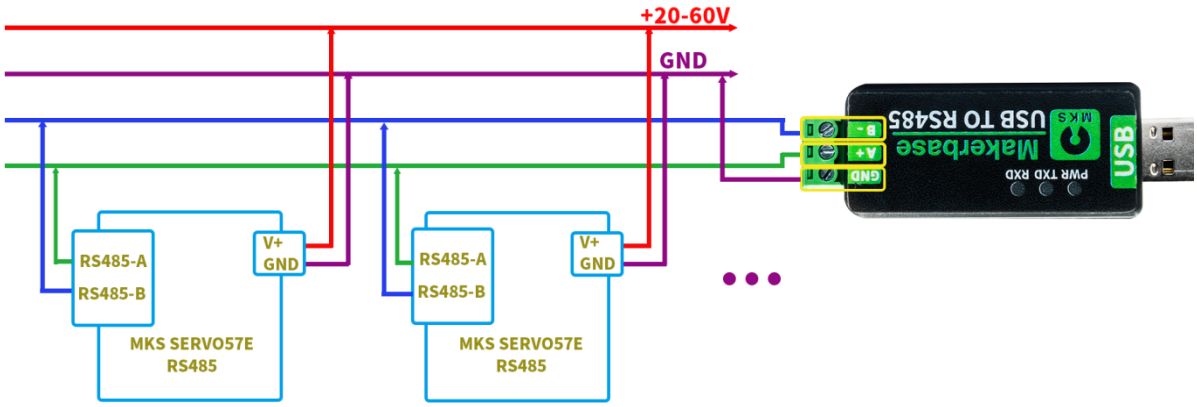


2. Multiple-slave

① Connection method of SERVO42E_RS485



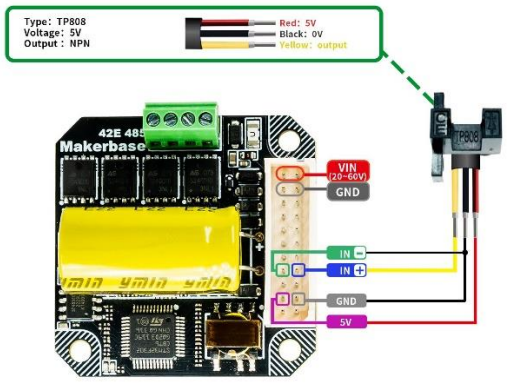
② Connection method of SERVO57E_RS485



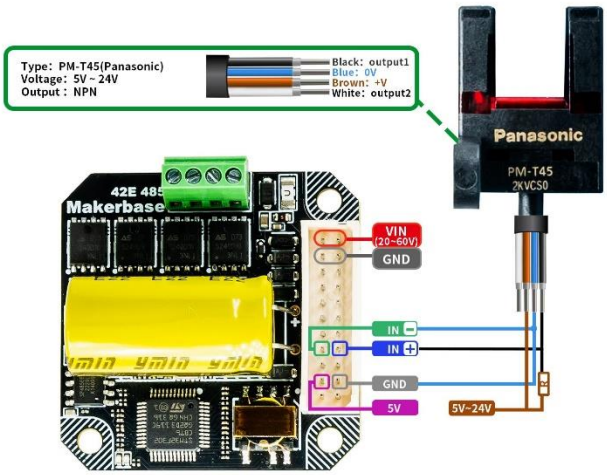
2.3 End stop wire

① SERVO42E_RS485

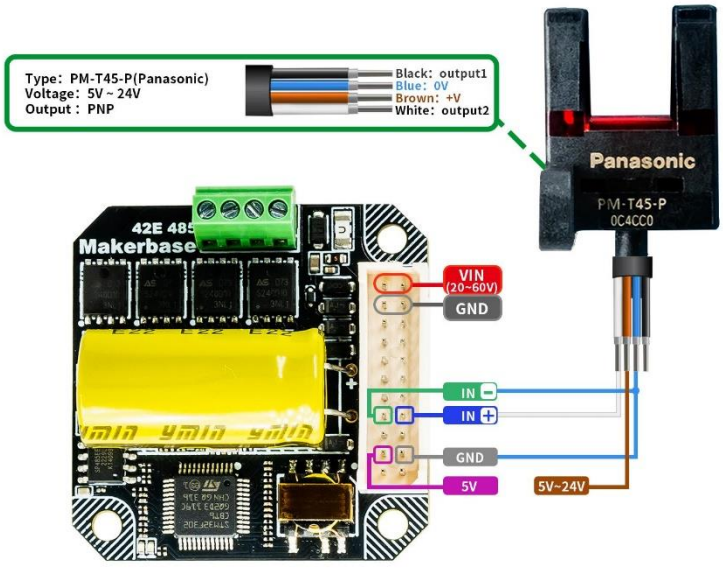
a) TP808 wire



b) PM-T45 wire

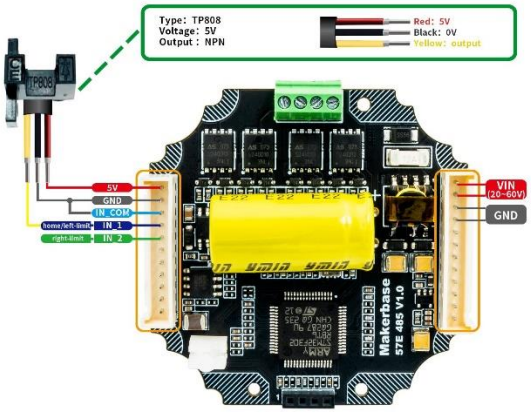


c) PM-T45-P wire

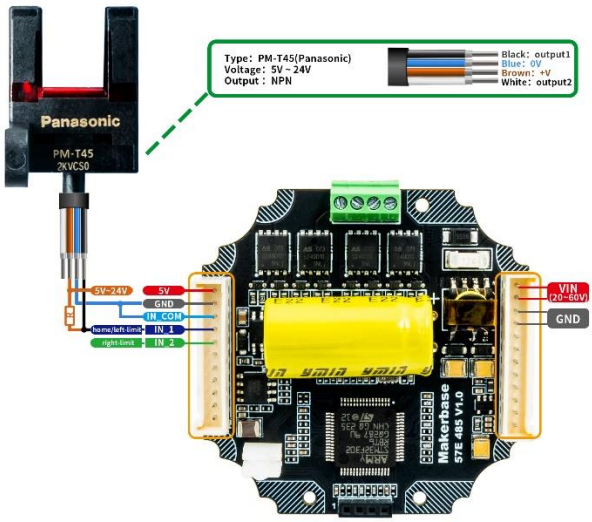


② SERVO57E_RS485

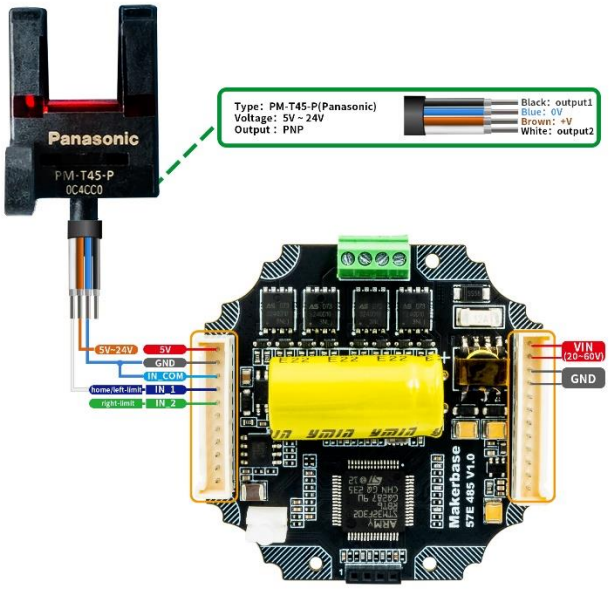
a) TP808 wire



b) PM-T45 wire



c) PM-T45-P wire



Part3. Serial data format

Note: For MODBUS-RTU protocol commands, see Part 7.

Downlink package(PC → SERVO42E/57E)						
Head	Slave addr	Function	Data			Check code
FA	addr	code				CRC
Uplink package (PC ← SERVO42E/57E)						
Head	Slave addr	Function	Data			Check code
FB	addr	code				CRC

1. The Data is Big-endian.
2. Downlink package Head is “FA”, uplink package Head is “FB”.
3. The slave address(addr) range is 00~255.（default is 01）.
   00 is the broadcast address;
4. The function code (code) executes the corresponding command.
   for example, 0x80 executes the calibration command.
5. The Check code is CHECKSUM 8bit
   For example: command “FA 01 80 00 CRC”
   CRC = (0xFA + 0x01 + 0x80 + 0x00) & 0xFF = 0x17B & 0xFF = 0x7B
6. When the host computer sends a command, the timing between the bytes of a single command (FA ... CRC) must be continuous, and there must not be more than one byte delay, otherwise multiple idle line will be detected and the motor may fail to receive the command..

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

Part4. 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 7.

4.1 Read parameter command

1. command1 : FA 01 30 CRC

read the encoder value(carry).

Uplink package (PC ← SERVO42E/57E)					
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 ← SERVO42E/57E)				
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 ← SERVO42E/57E)				
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 ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	33	pulses(int32_t)	CRC

5. Command5 : FA 01 34 CRC

read the IO Ports status.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	34	status(uint8_t)	CRC

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

PEND 1: Already in place 0: Not in place

ALM 1: No alarm 0: Alarmed

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

6. Command6 : FA 01 39 CRC

read the error of the motor shaft angle.

Uplink package (PC ← SERVO42E/57E)				
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. Command7 : FA 01 3A CRC

read the En pins status.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	3A	enable(uint8_t)	CRC

enable =1 Enabled

enable =0 Disabled

8. Command8 : FA 01 3D CRC

Release the motor shaft locked-rotor protection state.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	3D	status(uint8_t)	CRC

status =1 release success.

status =0 release fail.

Note: The stall state can also be released through the "EN level invalid mode"

9. Command9 : FA 01 3E CRC

Read the motor shaft protection state.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	3E	status(uint8_t)	CRC

status =1 protected.

status =0 no protected.

4.2 Set parameters command

1.Calibrate the encoder

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	80	00	CRC

Uplink package (PC ← SERVO42E/57E)				
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 calibration only determines the relationship between the motor direction and the encoder, that is, when the motor rotates clockwise, the encoder value increases or decreases. If the motor phase line is wired according to the factory default connection, no calibration is required.

2.Set the work mode

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	82	mode (uint8_t)	CRC

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

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

mode = 02 Pulse+Pulse Closed-loop mode

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

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

mode = 05 RS485 bus Closed-loop mode

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

Note 2: X=0 with encoder, the motor shaft need a magnet, the driver board is installed at the back, and the encoder value can be read.

X=1 without encoder, the motor shaft without magnet, the driver board can be installed arbitrarily, and the encoder value cannot be read.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	82	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

3.Set the working current

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	83	Current(uint16_t)	CRC

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	83	status(uint8_t)	CRC

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

4.Set subdivision(Default 16 subdivisions)

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	84	micstep(00~FF)	CRC

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	84	status(uint8_t)	CRC

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

5.Set the active of the En pin

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	85	enable(00~02)	CRC

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	85	status(uint8_t)	CRC

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

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

Note2:Only valid for pulse control mode.

6.Set the direction of motor rotation

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	86	dir(00~01)	CRC

dir = 00 CW (default)

dir = 01 CCW

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	86	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

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

7. Set pulse delay

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	87	delay	CRC

delay = 00 0ms delay = 01 4ms

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	87	delay	CRC

status =1 Set success.

status =0 Set fail.

8.Set the motor shaft locked-rotor protection function

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	88	enable(00~01)	CRC

enable = 01 enabled protection(default value)

enable = 00 disabled protection

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	88	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

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

9. Set the Stalling tolerant value

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	89	value（0-0x7FFF）	CRC

The default value is 0x64.
When the error exceeds the value, the stall protection is triggered and the motor loosens its shaft.
value = 0x64 corresponds to an angle of 180 degrees
value = 0xC8 corresponds to an angle of 360 degrees
and so on...

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	89	status(uint8_t)	CRC

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

10. Set the baud rate

Downlink package (PC → SERVO42E/57E)				
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(default value).
baud = 05 57600.
baud = 06 115200.
baud = 07 256000.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	8A	status(uint8_t)	CRC

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

11. Set the slave address

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	8B	addr(00~FF)	CRC

Note1: 00 is the broadcast address

Note2: 01 is the default address

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	8B	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

12. Set the slave respond and active

Downlink package (PC → SERVO42E/57E)					
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 ← SERVO42E/57E)				
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.

13. Set MODBUS-RTU communication protocol

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	8E	enable(00~01)	CRC

enable = 01 enabled Modbus-RTU

enable = 00(default value) disabled Modbus-RTU

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	8E	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

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

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	8F	enable(00~01)	CRC

enable = 01 lock the axis(default value)

enable = 00 unlock axis

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	8F	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

15. Set the group address

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	8D	addr(01~FF)	CRC

Uplink package (PC ← SERVO42E/57E)				
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.

4.3 Write IO port command

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	36	status(uint8_t)	CRC

status							
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
alm_mask		pend_mask		ALM	PEND	0	0

alm_mask 0: Do not write to ALM IO port (default)
1: Write the ALM value to the ALM IO port
2: ALM IO port value remains unchanged
pend_mask 0: Do not write to the PEND IO port (defaults)
1: Write the PEND value to the PEND IO port
2: The PEND IO port value remains unchanged
ALM ALM port write value (0/1)
PEND PEND port write value (0/1)

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	36	status(uint8_t)	CRC

status =1 write success.
status =0 write fail.

Note1: ALM writes 1, the corresponding optocoupler is opened.
Note2: PEND writes 1, the corresponding optocoupler is closed.

4.4 Set Home command

1. Set the parameter of home

Downlink package (PC → SERVO42E/57E)							
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 (default value) 1: High

HmDir the direction of go home

0: CW(default value) 1: CCW

HmSpeed the speed of go home

0~3000 (RPM) default value = 60

EndLimit

0: disable endstop-limit(default value)

1: enable endstop-limit

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	90	status(uint8_t)	CRC

status =1 Set success.

status =0 Set fail.

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

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

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

2. Go home

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	91	NULL	CRC

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

Uplink package (PC ← SERVO42E/57E)				
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 → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	92	NULL	CRC

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	92	status(uint8_t)	CRC

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

4. Set the unlimited switch to return to zero current

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	93	nullHmMa	CRC

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	93	status(uint8_t)	CRC

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

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

5. Set the parameter of "noLimit" go home

Downlink package (PC → SERVO42E/57E)						
byte1	byte2	byte3	byte 4-7	byte 8	byte 9	byte 10
Head	Slave addr	Function	Reverse Angle	hm-Mode	Hm_Trig	Check
FA	01	94	retValue	mode	trig	CRC

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

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

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

for example:

retValue = 0x4000 (it will return 360 degree)

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

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	94	status(uint8_t)	CRC

status =0 set fail.

status =1 set success.

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

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

6. Set limit port remap

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

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	9E	reMap	CRC

reMap = 0 Right limit mapping is off (default value)
reMap = 1 Right limit mapping is on

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	9E	status(uint8_t)	CRC

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

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

4.5 Setting the pulse division output command

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

Downlink package（PC → SERVO42E/57E）					
byte1	byte2	byte3	byte 4	byte 5-8	byte 9
Head	Slave addr	Function	Start level	division period	Check
FA	01	9F	divLevel	divPeriod	CRC

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

When divPeriod < 100, there is no frequency division output
When divPeriod >= 100, the PEND port flips once for every divPeriod pulse cycle.
For example, if 16 subdivisions are set and divPeriod = 3200, the PEND port flips once for every motor rotation.
Note: To cancel this function, set divPeriod = 0.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	9F	status(uint8_t)	CRC

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

4.6 Restore default parameters and reset instructions

1. Restore default parameters

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	3F	NULL	CRC

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	3F	status(uint8_t)	CRC

status =1 restore success.

status =0 restore fail.

Note: After restored the parameters, It will reboot.

2. Reset the motor

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	41	NULL	CRC

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	41	status(uint8_t)	CRC

status =1 restart success.

status =0 restart fail.

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

4.7 Read version information

Downlink package (PC → SERVO42E/57E)			
Head	Slave addr	Function	CRC
FA	01	40	CRC

byte1	byte 2	byte 3	byte 4			byte 5-7	byte 8
head	addr	cmd	b7	b5-b4	b3-b0	firmVer	CRC
			D/E	cal	hardVer		
FB	addr	40	series	cal	hardVer	firmVer[3]	CRC

series = 1 E series stepper motor
series = 0 D series stepper motor
cal = 1 When the motor rotates clockwise, the encoder value increases
cal = 2 When the motor rotates clockwise, the encoder value decreases

Firmware version: firmVer[0] = 1 firmVer[1] = 0 firmVer[2] = 0
Corresponding version V1.0.0

The hardware versions correspond to the following

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

4.8 Read/Write User ID

1. Write User ID

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	User ID	CRC
FA	01	42	ID(uint32_t)	CRC

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	42	status(uint8_t)	CRC

status =1 Write success.

status =0 Write fail.

2. Read User ID

Downlink package (PC → SERVO42E/57E)			
Head	Slave addr	Function	CRC
FA	01	42	CRC

Downlink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	User ID	CRC
FB	01	42	ID(uint32_t)	CRC

4.9 Firmware upgrade by IAP

There are 2 upgrade modes:

IAP mode 1:

Ground the SWD port of the motor, then power on and enter mode 1. Factory parameters will be restored after the upgrade.

IAP mode 2:

Send control instructions and enter IAP mode 2. User parameters will be retained after the upgrade.

Note:

For IAP upgrade operation instructions, see "MKS SERVO42&57E IAP upgrade operation instructions.pdf"

For IAP upgrade operation video, see "MKS SERVO42&57E IAP upgrade operation video.mp4"

IAP mode 2 instructions are as follows

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	CMD	CRC
FA	01	50	cmd	CRC

cmd = 01 Enter boot mode

cmd = 02 Enter silent state

cmd = 03 Exit silent state

Control word cmd description:

When there is only one motor on the bus, directly send cmd = 01 command to enter boot mode;

When there are multiple motors on the bus, to avoid data interference, you can do the following:

a. First broadcast cmd = 02 command (FA 00 50 02 CRC) to make all motors enter the silent state;

b. Then send cmd = 01 command to the motor to be upgraded to enter the boot mode to upgrade the firmware;

c. After the upgrade is completed, broadcast cmd = 03 command (FA 00 50 03 CRC) to make all motors exit the silent state.

Note: In the silent state, the motor does not respond to commands other than 50.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	50	status(uint8_t)	CRC

status =1 Write success.

status =0 Write fail.

Note: After successful setup, the motor automatically restarts and enters IAP mode 2, waiting to receive the upgrade file(*.bin).

4.10 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)

4.11 Read/Write all parameters commands

3. Write all configuration parameters

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Byte4~Byte37	CRC
FA	01	46	parameters	CRC

Note: The parameters are shown in the table below.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	46	status(uint8_t)	CRC

status =1 write success.
status =0 write fail.

4. Read all configuration parameters

Downlink package (PC → SERVO42E/57E)			
Head	Slave addr	Function	CRC
FA	01	47	42

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	parameters	CRC
FB	01	47	Byte4~Byte37	CRC

Note1: The parameters are shown in the table below.
Note2: If read failed,it return FB 01 47 FF 42

Write commands (46)			
byte	Frame	Single cmd	default(HEX)
1	FA		
2	01		
3	46		
4	Mode	82	3
5	Holding current		Reserved (FF)
6	Work current	83	0C / 06
7			80 / 40
8	Subdivision	84	10
9	En	85	0
10	Dir	86	0
11	Pulse Delay	87	0
12	Protect	88	0
13	Baud rate	8A	4
14	Slave address	8B	1

Read commands (47)	
byte	Frame
1	FB
2	01
3	47
4	Mode
5	Reserved (FF)
6	Work current
7	
8	Subdivision
9	En
10	Dir
11	Pulse Delay
12	Protect
13	Baud rate
14	Slave address

15	Group address	8D	0	15	Group address
16	Respond	8C	1	16	Respond
17			1	17	
18	MODBUS	8E	0	18	MODBUS
19	Limit remap	9E		19	Limit remap
20	Axis lock	8F	1	20	Axis lock
21	Reserved		Reserved(FF)	21	Reserved(FF)
22	HmTrig	90	0	22	HmTrig
23	HmDir		0	23	HmDir
24	HmSpeed		0	24	HmSpeed
25			3C	25	
26	EndLimit		0	26	EndLimit
27	Reserved		Reserved(FF)	27	Reserved(FF)
28	retValue	94	0	28	retValue
29			0	29	
30			20	30	
31			0	31	
32	Hm-mode		0	32	Hm-mode
33	hmTrig		0	33	hmTrig
34	nullHmMa	93	02/01	34	nullHmMa
35			58/2C	35	
36	tolerance value	89	00	36	tolerance value
37			64	37	
38	CRC			38	CRC

Note：xx/xx corresponds to 57E/42E

5. Read all status parameters

Downlink package (PC → SERVO42E/57E)			
Head	Slave addr	Function	CRC
FA	01	48	43

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	parameters	CRC
FB	01	48	Byte4~Byte30	CRC

Note1: The parameters are shown in the table below.

Note2: If read failed,it return FB 01 48 FF 43

Read status commands (48)		
Bytes	Frame	Single commad
1	FB	
2	1	
3	48	
4	motor status	F1
5	En status	3A
6	Protect status	3E
7	IO status	34
8	encoder value	31
9		
10		
11		
12		
13		
14	speed	32
15		
16	pulses	33
17		
18		
19		
20	error	39
21		
22		
23		
24	CRC	

Part5. Run the motor by serial command

Note1: This command is only valid in bus mode.
Note2: For MODBUS-RTU protocol commands, see Part 7.

5.1 Description the parameters of speed and acceleration

1.speed

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

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

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

2. acceleration

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

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

① accelerates

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

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

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

5.2 Query/Enable the motor command

1. Query the motor status

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	F1	—	CRC

Uplink package (PC ← SERVO42E/57E)				
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 = 6 motor is Cal…

Note 1: This instruction is only valid in bus control mode.
Note 2: This instruction can only query the motor calibration operation in pulse control mode.

2. Enable the motor

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	F3	en (00~01)	CRC

en = 00 disable.
en = 01 enable.

Uplink package (PC ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	F3	status(uint8_t)	CRC

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

5.3 Emergency stop the motor

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	F7	—	CRC

Uplink package (PC ← SERVO42E/57E)				
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!

5.4 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.

5.5 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 → SERVO42E/57E)								
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 ← SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FB	01	F6	status(uint8_t)	CRC

status = 1 run success.

status = 0 run fail.

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

2. Stop the motor in speed mode

Downlink package (PC → SERVO42E/57E)								
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 ← SERVO42E/57E)				
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: This instruction is only valid in bus control mode.

3. Save/Clean the parameter in speed mode

Downlink package (PC → SERVO42E/57E)				
Head	Slave addr	Function	Data	CRC
FA	01	FF	state	CRC

state = C8 Save.
state = CA Clean.

Uplink package (PC ← SERVO42E/57E)				
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.

5.6 Position model: 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 model

Downlink package (PC → SERVO42E/57E)									
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 ← SERVO42E/57E)				
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 → SERVO42E/57E)								
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 ← SERVO42E/57E)				
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.

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

5.7 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 ← SERVO42E/57E)				
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: This instruction is only valid in bus control mode.

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 ← SERVO42E/57E)				
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: This instruction is only valid in bus control mode.

5.8 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 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	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 ← SERVO42E/57E)				
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 instruction "8C" can be used to set whether to return to the running state.

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 ← SERVO42E/57E)				
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 Command “8C”.

5.9 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 ← SERVO42E/57E)				
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 instruction "8C" can be used to set whether to return to the running state.

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 ← SERVO42E/57E)				
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 instruction "8C" can be used to set whether to return to the running state.

Part6. Serial example

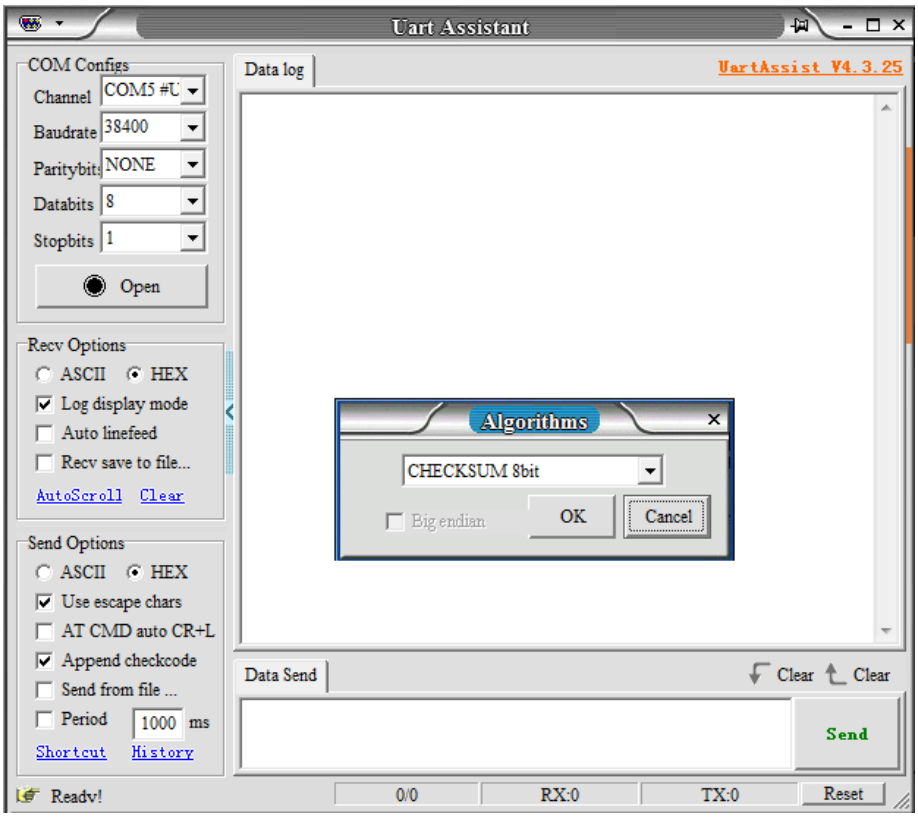
6.1 Config the SERVO42E/57E

1. Set the working mode: 05 RS485 bus closed loop FOC mode
2. Setting the baud rate: 04 38400 (default value).
3. Set the slave address:01 (default value).

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".

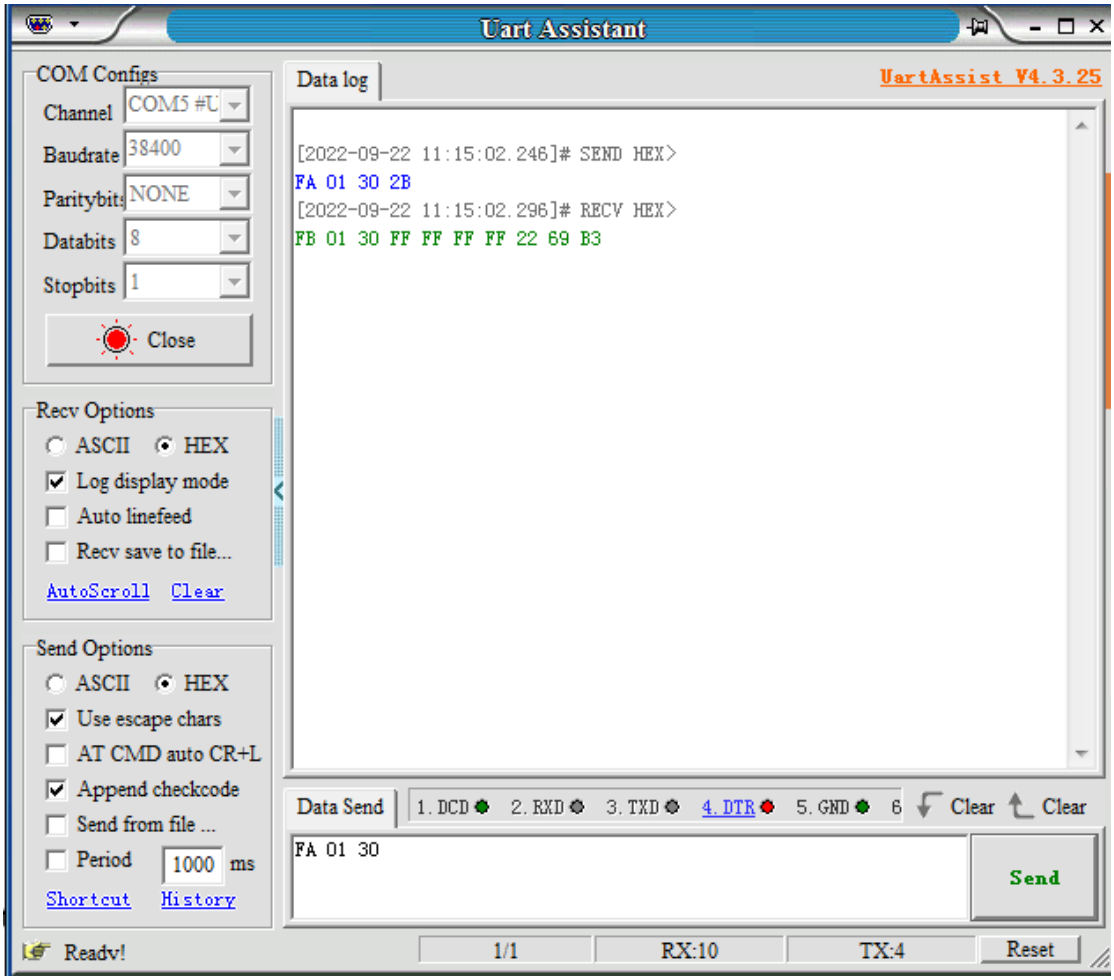
Such as below:



6.3 Read the encoder value

send “FA 01 30 2B”

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



6.4 Run the motor in speed mode

Note : The control mode needs to be set to bus mode.

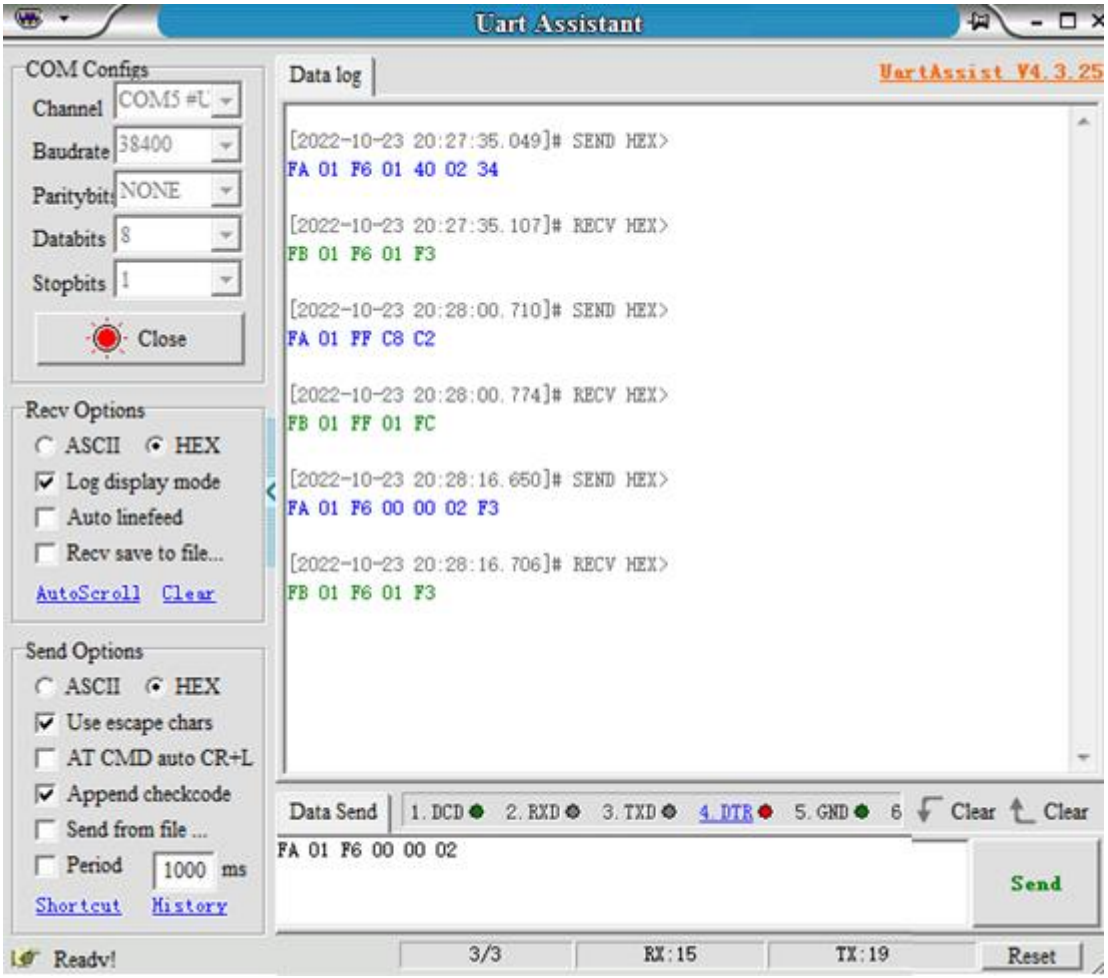
1. Send FA 01 F6 01 40 02 , the motor will rotate at "speed = 320RPM, acc=2";
Return FB 01 F6 01 F3, the motor run in speed mode successful;

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:



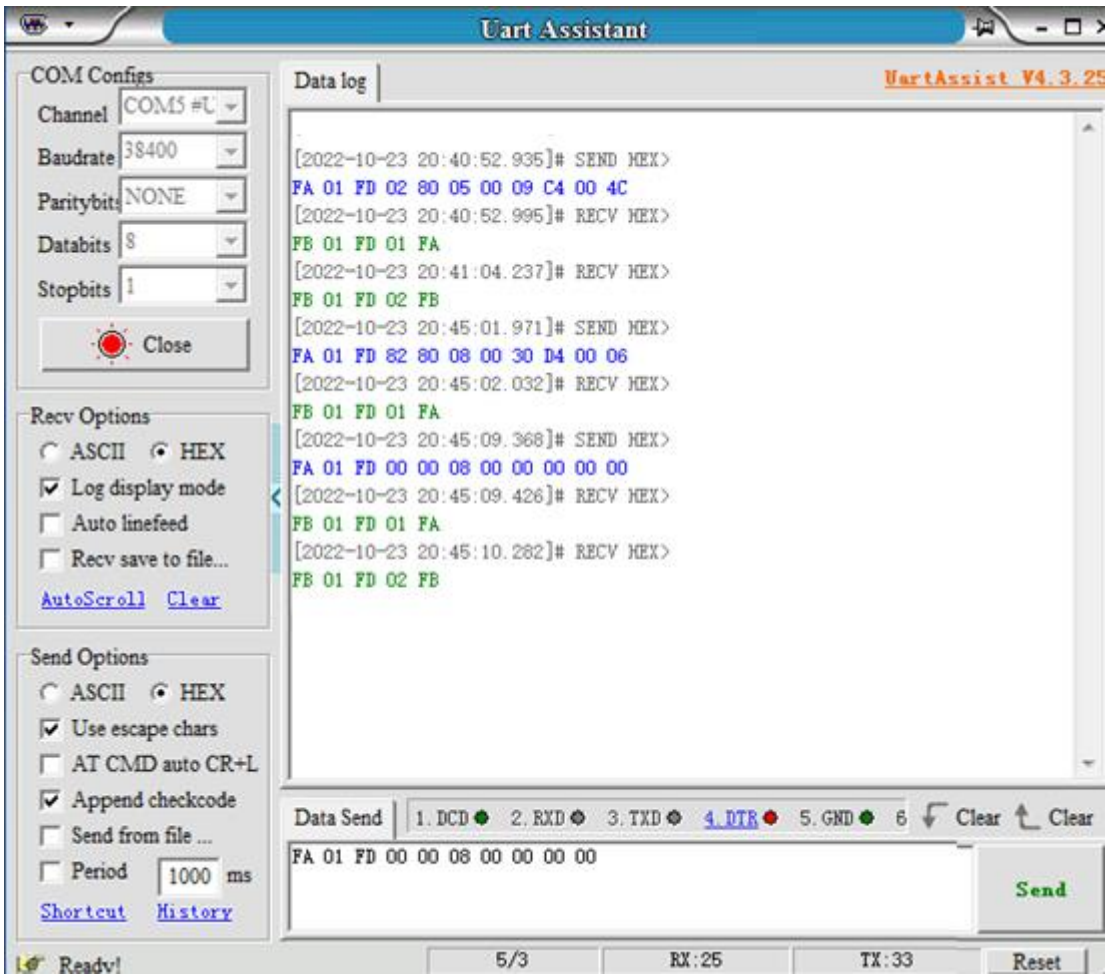
6.5 Run the motor in position model

Note : The control mode needs to be set to bus mode.

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:



Part7. MODBUS-RTU command description

Note1: It need to enable MODBUS-RTU by menu or serial command.

Note2: the addresses 1046H, 1147H, and 1248H to write or read all parameters , refer to Section 7.3.

7.1 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.

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.

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)

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)		

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				ALM	PEND	IN_2	IN_1

PEND 1: Already in place 0: Not in place

ALM 1: No alarm 0: Alarmed

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

6. 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.

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

8. 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.

9. Read version information

Request							
SlaveAddr	Function	Starting Address		Quantity of Reg		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	00H	40H	00H	02H	70H	1FH

Response					
SlaveAddr	Function	Bytes	Version Information	CRC16	
				Hi	Lo
01H	04H	04H	version(uint32_t)		

Byte0			Byte1-3
b7	b5-b4	b3-b0	fimware
series	cal	hardVer	firmVer[3]

series = 1 E series stepper motor
series = 0 D series stepper motor
cal = 1 When the motor rotates clockwise, the encoder value increases
cal = 2 When the motor rotates clockwise, the encoder value decreases

Firmware version: firmVer[0] = 1 firmVer[1] = 0 firmVer[2] = 0
Corresponding version V1.0.0

The hardware versions correspond to the following

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

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…

7.2 Write parameter command

Note: If write fails, the function code 06H response frame register data is 0xFFFF.The function code 10H response frame register quantity is 0.

1. Write the IO port

Request												
Slave Addr	Function	Address		Quantity		Bytes	REG1		REG2		CRC16	
		Hi	Lo	Hi	Lo		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	36H	00H	02H	04H	alm_mask	ALM	pend_mask	PEND		

alm_mask 0: Do not write to ALM IO port (ALM default alarm signal)
1: Write the ALM value to the ALM IO port
2: ALM IO port value remains unchanged

pend_mask 0: Do not write to the PEND IO port (PEND defaults to the in-place signal)
1: Write the PEND value to the PEND IO port
2: The PEND IO port value remains unchanged

ALM ALM port write value (0/1)

PEND PEND port write value (0/1)

Response							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	36H	00H	02H	A1H	C6H

Note1: ALM writes 1, the corresponding optocoupler is opened.
Note2: PEND writes 1, the corresponding optocoupler is closed.

2. 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

Note: The stall state can also be released through the "EN level invalid mode"

3. Restore the default parameter

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

Note: After restored the parameters, It will reboot.

4. 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

5. Calibrate the motor

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

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

6. Set the work mode

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

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

mode = 02 Pulse+Pulse Closed-loop mode

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

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

mode = 05 RS485 bus Closed-loop mode

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

Note 2: X=0 with encoder, the motor shaft need a magnet, the driver board is installed at the back, and the encoder value can be read.

X=1 without encoder, the motor shaft without magnet, the driver board can be installed arbitrarily, and the encoder value cannot be read.

7. Set the work current

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			

SERVO42E: Maximum Current =3000mA (default 1600mA)

SERVO57E: Maximum Current =5200mA (default 3200mA)

8. Set subdivision

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			

The value range of micstep (decimal) is as follows:

0, 2, 4, 8, 16, 32, 64, 128,

5, 10, 20, 25, 40, 50, 100, 200

Note: 0 corresponds to 256 subdivisions. (default value is 16)

9. Set the active of the En pin

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) (default value)

enable = 01 active high (H)

enable = 02 active always (Hold)

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

Note2:Only valid for pulse control mode.

10. Set the direction of motor rotation

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(default value)
dir = 01 CCW
Note:This instruction can also change the bus mode to control the running direction of the motor

11. Set pulse delay

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

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

time = 00 0ms time = 01 4ms
time = 02 20ms(default value) time = 03 40ms

12. Set the motor shaft locked-rotor protection

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 (default value)
enable = 00 disabled protection

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

13. Set the stall tolerance value

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

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

The default value is 0x64.

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

value = 0x64 corresponds to an angle of 180 degrees

value = 0xC8 corresponds to an angle of 360 degrees

and so on...

14. Set the baud rate

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.

15. Set the slave address

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 address range is 00~0xFF, 00 is the broadcast address, 01 is the default address.

16. Set MODBUS-RTU communication protocol

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.

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

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 axis(default value).
enable = 00 unlocked the axis.

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.

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.

Note : This instruction is only valid in bus control mode

20. Set the parameter of home

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(default value)

1: High

hmDir the direction of go home

0: CW(default value)

1: CCW

hmSpeed the speed of go home

0~3000 (RPM)

EndLimit

0: disable endstop-limit(default value)

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

Note 2: When using the limit function for the first time or changing the limit parameters, it is necessary to execute a limit reset ("91" command).

Note 3:The limit function is invalid in pulse control mode.

21. Set the parameter of “noLimit” go home

Slave addr	Function	Starting Address		Quantity of Registers		Bytes	Reverse Angle	Zero return mode	Zero return trigger	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	00H	94H	00H	03H	06H	retValue(uint32_t)	HmMode (uint8_t)	hmTrig (uint8_t)		

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

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

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

for example:

retValue = 0x4000 (it will return 360 degree)

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

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	94H	00H	03	C1H	E4H

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

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

22. Setting the pulse division output command

Slave addr	Function	Starting Address		Quantity of Registers		Bytes	Start level	division period	CRC16	
		Hi	Lo	Hi	Lo				Hi	Lo
01H	10H	00H	9FH	00H	03H	06H	divLevel (uint8_t)	divPeriod (uint32_t)		

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

divPeriod division period (default 0)

When divPeriod < 100, there is no frequency division output

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

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

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

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	00H	9FH	00H	03	B0H	26H

7.3 Firmware upgrade by IAP

There are 2 upgrade modes:

IAP mode 1:

Ground the SWD port of the motor, then power on and enter mode 1. Factory parameters will be restored after the upgrade.

IAP mode 2:

Send control instructions and enter IAP mode 2. User parameters will be retained after the upgrade.

Note:

For IAP upgrade operation instructions, see "MKS SERVO42&57E IAP upgrade operation instructions.pdf"

For IAP upgrade operation video, see "MKS SERVO42&57E IAP upgrade operation video.mp4"

IAP mode 2 instructions are as follows

Request							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	50H	00H	cmd	48H	1BH

cmd = 01 Enter boot mode

cmd = 02 Enter silent state

cmd = 03 Exit silent state

Control word cmd description:

When there is only one motor on the bus, directly set cmd = 01 to enter boot mode;

When there are multiple motors on the bus, to avoid data interference, you can do the following:

a. First send cmd = 02 command to other motors that are not upgraded to enter silent state;

b. Then send cmd = 01 command to the motor to be upgraded to enter boot mode and upgrade the firmware;

c. After the upgrade is completed, send cmd = 03 command to other motors to exit silent state.

Note: In silent state, the motor does not respond to commands other than 50H.

Response							
SlaveAddr	Function	Register Address		Write Data		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	06H	00H	50H	00H	01H	48H	1BH

Note: After successful setup, the motor automatically restarts and enters IAP mode 2, waiting to receive the upgrade file (*.bin).

7.4 Read/Write all parameters commands

1. Write all configuration parameters

Request											
Slave addr	Function	Starting Address		Quantity of Registers		Bytes	reg1	…	reg19	CRC16	
		Hi	Lo	Hi	Lo					Hi	Lo
01H	10H	10H	46H	00H	11H	20H					

Note:The definitions of reg 1…reg17 are shown in the following table. < Configuration parameters table>

The default register parameters are as follows:

03 FF 0C 80 10 00 00 02 01 04 01 00 01 01 01 00 01 FF 00 00 00 3C 00 FF 00 00 20 00 00 00 02 58 00 64

Response							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	10H	10H	46H	00H	11H	E5H	10H

2. Read all configuration parameters

Request							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	11H	47H	00H	11H	85H	2FH

Response							
Slave addr	Function	Bytes	reg1	…	reg16	CRC16	
						Hi	Lo
01H	04H	20H					

Note:The definitions of reg 1…reg17 are shown in the following table. < Configuration parameters table>

Configuration parameters table

Write parameters (1046H)				
byte		Frame	default(HEX)	Single cmd
1		Slave addr	01	
2		Function	10	
3		Starting Address	00	
4			46	
5			00	

Read parameters(1147H)		
byte		返回数据格式
1	Slave addr	01
2	Function	04

6		Quantity of Registers	10	
7		Bytes	20	
8	REG1	Mode	02	82
9		Hold current	Reserve(FF)	
10	REG 2	Work current	0C / 06	83
11			80 / 40	
12	REG 3	Subdivision	10	84
13		En	0	85
14	REG 4	Dir	0	86
15		Pulse Delay	2	87
16	REG 5	Protect	0	88
17		Baud rate	4	8A
18	REG 6	Slave address	1	8B
19		Group address	0	8D
20	REG 7	Respond	1	8C
21			1	
22	REG 8	MODBUS	0	8E
23		Limit remap	0	9E
24	REG 9	Bus lock shaft	1	8F
25		Reserve(FF)	Reserve(FF)	
26	REG 10	HmTrig	0	90
27		HmDir	0	
28	REG 11	HmSpeed	0	
29			3C	
30	REG 12	EndLimit	0	
31		Reserve(FF)	Reserve(FF)	
32	REG 13	retValue	0	94
33			0	
34	REG 14		20	
35			0	
36	REG 15	Hm-mode	0	
37		Hm-Trig	0	
38	REG 16	Hm_ma	02/ 01	93
39			58 / 90	
40	REG 17	tolerance value	00	89
41			64	
42	CRC16			

3	Bytes	20
4	REG 1	Mode
5		Hold current
6	REG 2	Work current
7		
8	REG 3	Subdivision
9		En
10	REG 4	Dir
11		Pulse Delay
12	REG 5	Protect
13		Baud rate
14	REG 6	Slave address
15		Group address
16	REG 7	Respond
17		
18	REG 8	MODBUS
19		Limit remap
20	REG 9	Bus lock shaft
21		Reserve(FF)
22	REG 10	HmTrig
23		HmDir
24	REG 11	HmSpeed
25		
26	REG 12	EndLimit
27		Reserve
28	REG 13	retValue
29		
30	REG 14	
31		
32	REG 15	Hm-mode
33		Hm-Trig
34	REG 16	Hm_ma
35		
36	REG 17	tolerance value
37		
38	CRC16	

3. Read all status parameters

Request							
Slave addr	Function	Starting Address		Quantity of Registers		CRC16	
		Hi	Lo	Hi	Lo	Hi	Lo
01H	04H	12H	48H	00H	0EH	F5H	63H

Response							
Slave addr	Function	bytes	reg1	…	reg14	CRC16	
						Hi	Lo
01H	04H	14H					

Note:The definitions of reg 1…reg10 are shown in the following table. < Status parameters table>

Status parameters table

Status parameters(1248H)			
Byte		Response(HEX)	Single commad
1	Slave addr	01	
2	Function	04	
3	bytes	14	
4	REG 1	motor status	F1
5		En status	3A
6	REG 2	Protect status	3E
7		IO status	34
8	REG 3	encoder value	31
9			
10	REG 4		
11			
12	REG 5		
13			
14	REG 6	speed	32
15			
16	REG 7	pulses	33
17			
18	REG 8		
19			
20	REG 9	error	39
21			
22	REG 10		
23			
24	CRC16		
25			

7.5 Motor running command

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

7.4.1 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!

7.4.2 Go home

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

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

7.4.3 Speed mode command

Note: This command is only valid in bus mode.

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

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

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		

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

7.4.4 Position mode1: relative motion by pulses

Note: This command is only valid in bus mode.

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

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

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

7.4.5 Position mode2: absolute motion by pulses

Note: This command is only valid in bus mode.

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!

7.4.6 Position mode3: relative motion by axis

Note1: This command is only valid in bus mode.
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

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

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

7.4.7 Position mode4: absolute motion by axis

Note: This command is only valid in bus mode.

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

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

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

Part8. FAQ

8.1 NOTE

1. Power input voltage is 20V-60V.
2. Don’t hot plug motor cable and data cable.
3. The phase lines A+, A -/B+, B - should be connected correspondingly. （A -, A+/B+, B - is incorrect）

8.2 FAQ

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

Part9. Schematic

Part10. contact us

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