DC integrated servo

YZ-AIM_v2.53

one. product features

1. A 15-bit absolute encoder with a loop pulse up to 32,768.
2. Multi-lap absolute value (battery is required).Pulse mode: recharge and automatically return to the power-off position.Communication mode: power-off recording location.
3. Multi-stage DD motor structure, large torque output.
4. Integrated servo, simplified wiring, ultra-small volume.
5. Low noise, low vibration, high-speed positioning, high reliability.
6. FOC field directional vector control, support position / speed closed loop.
7. Can work at the zero lag given the pulse state, following the zero lag.
8. The 16-bit electronic gear function.
9. modbu s R T U communication (19200,8, N, 1).
10. Position mode, supporting the pulse + direction signal
11. With blocking rotation, overcurrent protection, overpressure protection.

Two. Parameter table

Model parameters		42AIM15	42AIM10
source	voltage	24VDC±10%	24VDC±10%
	current	2.2A	1.6A
parameter of electric machine	torsion	0.48NM	0.33NM
	rated speed	1000RPM	1000RPM
	maximum speed	1500RPM	1500RPM
	power	50W	35W
return signal		Single lap 15-bit magneto-electric encoder (single lap 32768 pulse)	
cooling-down method		natural cooling	
weight			
Location control mode	Maximum input pulse frequency	500KHz	
	Pulse instruction mode	Pulse + direction, phase A + B phase	
	Electronic gear than	Set range 1~65535 to 1 to 65535	
	Location sampling frequency	2KHz	
defencive function		Block the alarm	
CI		RS485（modbusRTU 19200,8,N,1）	
service environment	ambient temperature	0~40°	
	Motor motor allowable maximum temperature	85°	
	humidity	5~95%	

Model parameters		57AIM15	57AIM15H	57AIM30	57AIM30H
source	voltage	24~36VDC	24~36VDC	24~36VDC	24~36VDC
	current	2.2A	2.2A	4.4A	4.4A
parameter of electric machine	torsion	0.48NM	0.24NM	0.96NM	0.48NM
	rated speed	1000RPM	2500RPM	1000RPM	2500RPM
	maximum speed	1500RPM	3000RPM	1500RPM	3000RPM
	power	50W	50W	100W	100W
return signal		Multi-lap absolute value encoder (single lap 32768 pulse, single lap 15-bit)			
cooling-down method		natural cooling			
weight					
Location control mode	Maximum input pulse frequency	500KHz			
	Pulse instruction mode	Pulse + direction, phase A + B phase			
	Electronic gear than	Set range 1~65535 to 1 to 65535			
	Location sampling frequency	2KHz			
defencive function		Block the alarm			
CI		RS485（modbusRTU 19200,8,N,1）			
service environment	ambient temperature	0~40°			
	Motor motor allowable maximum temperature	85°			
	humidity	5~95%			

Model parameters		60AIM25	60AIM25H
source	voltage	36VDC±10%	36VDC±10%
	current	7A	7A
parameter of electric machine	torsion	2NM	1NM
	rated speed	1000RPM	2500RPM
	maximum speed	1500RPM	3000RPM
	power	250W	250W
return signal		Single lap 15-bit magneto-electric encoder (single lap 32768 pulse)	
cooling-down method		natural cooling	
weight			
Location control mode	Maximum input pulse frequency	500KHz	
	Pulse instruction mode	Pulse + direction, phase A + B phase	
	Electronic gear than	Set range 1~65535 to 1 to 65535	
	Location sampling	2KHz	

	frequency	
defencive function		Block the alarm
CI		RS485（modbusRTU 19200,8,N,1）
service environment	ambient temperature	0~40°
	Motor motor allowable maximum temperature	85°
	humidity	5~95%

II. Drive interface

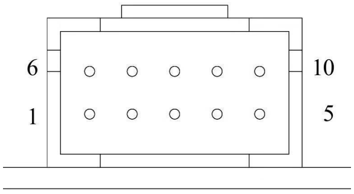
1. Power supply and control signal interface

Terminal serial number	name	function
1	+24V	DC power supply positive pole, + 24V.Positive and negative connections will directly short circuit the power supply and may also damage the drive
2	GND	DC power supply ground.Positive and negative connections will directly short circuit the power supply and may also damage the drive
3	PU+ (+5V)	Pulse control signal: the pulse rise line is effective; 3.3~5V at PU-high level, and 0~0.5V at low level. For a reliable response to the pulse signal, the pulse width should be greater than 1.2s.Series resistance is required when using + 12V or + 24V.
4	PU- (PU)	
5	DIR+ (+5V)	Direction signal: high / low level signal, To ensure the direction signal shall precede the pulse signal At least 5s were established.DIR-3.3-5 V at high levels and 0~0.5V at low levels.
6	DIR- (DIR)	

Terminal serial number: Facing the terminal, the first is on the left.

The AIM series uses a differential interface circuit can apply to the differential signal, single-end common Yin and co-Yang interface, built-in high-speed photoelectric coupler, allowing to receive a long-line drive, collector open circuit and PNP output circuit signals.

2. Communication and output interface



Terminal serial number: facing the terminal, the left row from left to right, and the top row from left to right is 6 7 8 9 10 from left to right.

Terminal serial number	name	function
1	NC	
2	485A	485 Positive end of the communication
3	485B	485 Negative communication terminal
4	EN+	Positive end of the enabling signal. 3.3~5V, if a voltage of 24V requires a string of 2K resistance.
5	EN-	Positive end of the enabling signal.
6	COM	The output signal is in common with the 485 power supply.
7	WR	Alarm signal output, internal is photocoupled NPN output. Normal is high resistance, lead to COM when the alarm.
8	RDY/PF	Servo is ready for the signal / in-place signal. After the automatic operation of the signal (on), when the following error is less than 0.5° Signal (on), following error greater than 0.5° no signal (high resistance).
9	ZO	Encoder zero-point output. There is a zero-point signal photocoupling NPN output guide communication signal.
10	485_5V	485 Communication is 5V power supply, which requires external power supply. (This power supply is powered via the controller)

3. Status indication and alarm

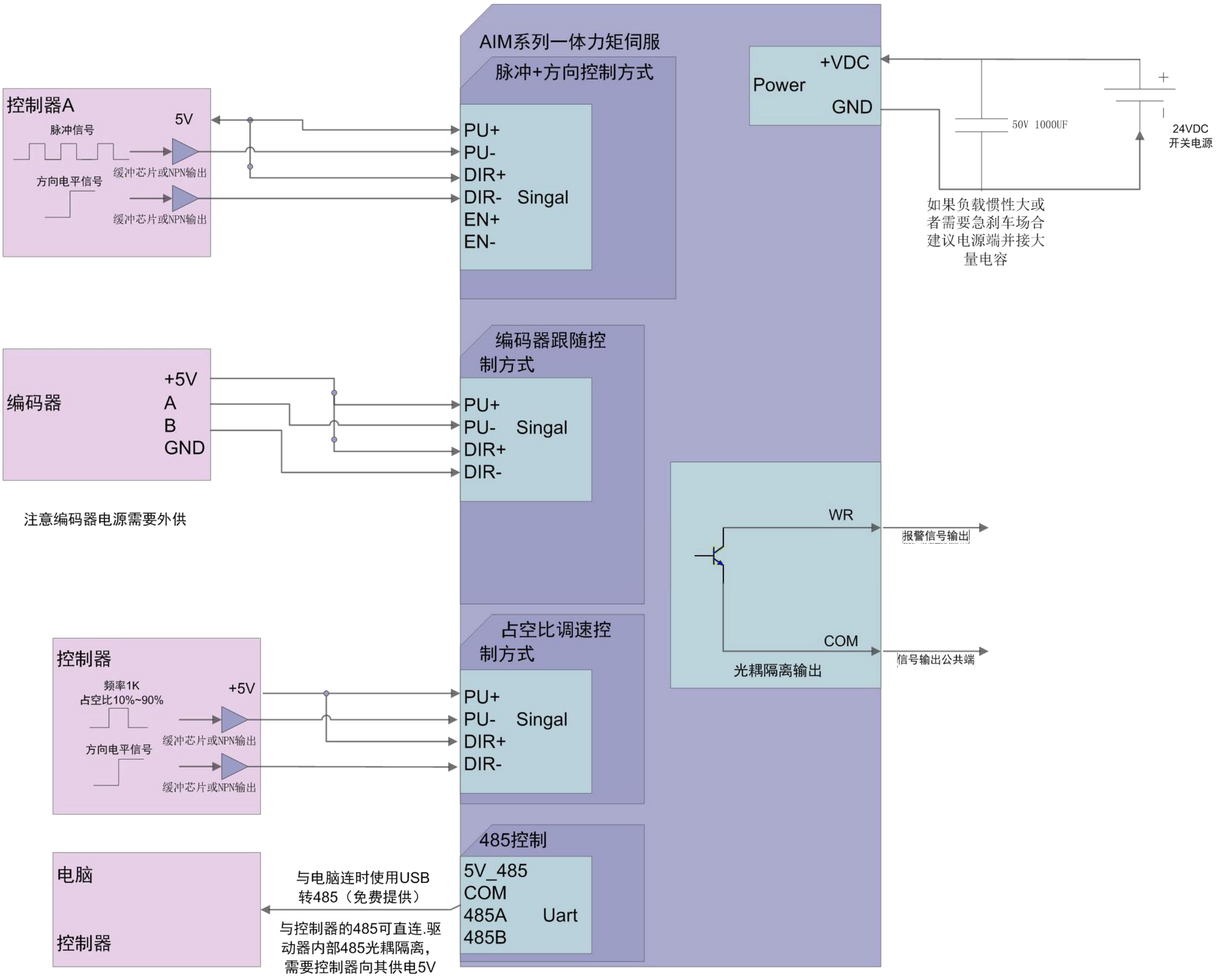
The red light is turned on once after turning on to check that the LED is working properly. Then the green light turns and the red light turns out to normal. If you encounter an alarm state, you can judge the cause by red, or you can read the alarm code through modbus.

Alarm code	blinking red lamp	The reason for the alarm	Alarm processing
0x10	A long flash	Battery off alarm	Only prompt, do not stop time
0x12	A long flash and a 2 short flash	Overflow alarm	halt
0x14	A long flash and a 4 short flash	Block the alarm	halt
0x15	A long flash and a 5 short flash	Overpressure alarm	Stop, large inertia load deceleration will generate electricity, which may be an overpressure alarm. Need to be in Add a discharge module at the power supply, or increase the capacitance and energy storage at the power source.

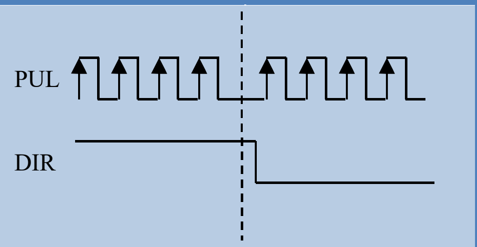
Note: Block turn alarm, block turn time can be set, specific according to the register description.

three. Drive wiring diagram and control mode

1. Typical drive wiring diagram



2.Command pulse + directional position control mode

Pulse	CCW CW	Parameter
Pulse		Instruction

If desired, 3200, the pulse in a ring

Electronic gear is set to 32768 (Number of pulse per encoder) ratio 3200 (number of pulses to set): 256 to 25

If 8192 pulse is required (default parameters)

Electronic gear is set to 32768 (Number of pulse per encoder) ratio 8192 (Number of one-ring pulses to be set) is about divided after points: 32768 to 8192

The approximate score is: 4 to 1

Note: can be about points as far as possible, the electronic gear molecule is, 32768, the value is too large, will affect the following performance

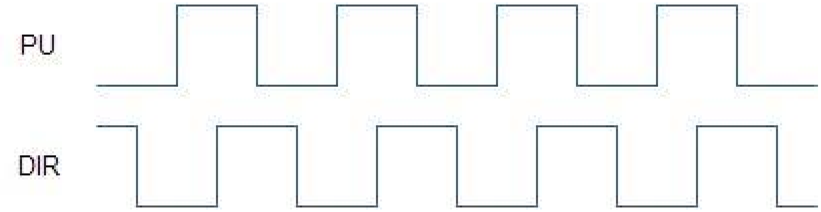
Command pulse frequency = (speed / 60 requiring motor operation) * Number of pulses in one lap, e. g. Number of pulses requiring 1000RPM is 8192

impulse frequency = 1000/60 * 8192= 136533HZ

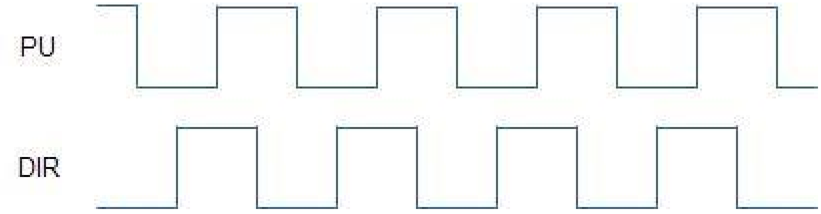
3. Orthogonal instruction pulse position control mode

By setting the special feature (0x19 address) to 2, the encoder follows the mode.This mode can be used to follow the encoder, such as an axis connecting the encoder, connecting the encoder output to the drive (wiring mode such as the drive typical wiring diagram), the drive can control the servo motor, press the input encoder signal, along with the controlled encoder.The ratio of the control encoder and the rotation angle of the motor can be set by adjusting the electronic gear.

Positive turn pulse:



Reverse pulse:



Direction of motor rotation: The PU rises positively along the advanced DIR.The PU rise along the lagging DIR rise along is reversed.

four. parameter testing

Depending on the load attached to the motor, the parameters need to be adjusted to achieve the best effect.

1. Internal acceleration and deceleration curve

Select whether to use an internal acceleration and deceleration curve depending on the controller output

signal.**Use the internal acceleration curve:**

When the motor acceleration is less than 60000, the drive enables the internal acceleration and deceleration curve, and the specific acceleration is the same size as the same value set.

Use occasion: use the internal acceleration curve, will produce a lag pulse phenomenon, some occasions that do not need to follow in real time, you can use the internal acceleration curve.Some controllers, pulse directly to the frequency of the corresponding speed, without acceleration and deceleration, the use of internal acceleration and deceleration curve, can reduce the difficulty of controller programming difficulty.

No internal acceleration curve:

When the motor acceleration is greater than or equal to 60,000, the drive is invalid according to the acceleration and deceleration permit of the external pulse.

Use occasion: for example, the pulse output by the controller is added and reduced, which does not need the acceleration curve inside the drive. If used at this time, it will lag behind the actual pulse.

2. Silk rod load

First, introduce the lower torque, first use 400W motor, 1.3NM.The load is a 5mm screw wire rod, which is when the motor shaft rotates and the load moves by 5mm,

Load-equivalency arm = 5mm / 3.14 = 1.592 mm

The thrust that the motor provides is

Ththrust = 1.3NM / (1.592mm * 0.001) = 816 N

The weight that pushes the load is about 80KG, which is vertical, and the flat push can be slightly larger.

Due to the short distance, the parameters of the drive (acceleration can be larger, such as 20000, position ring KP, such as 3000).The servo motor is best suitable for this load.

3. Belt wheel load

The servo motor is actually not very suitable for this load.Because the pulley is generally relatively large in diameter, such as the diameter of 30mm.The motor turns around and the load moving distance is 30mm * = 94.2, much times larger than the 5mm said above.

The thrust that the motor provides is

Through the belt drive, the thrust is = 1.3NM / (30mm * 0.001) = 43.3 N

The weight that pushes the load is about 4.3KG.Therefore, the servo motor is in fact not suitable for connecting the synchronous wheels, because the synchronous wheel turns a circle of load movement distance is too long, the force arm is long.If the servo motor is to be used in this occasion, you can choose the direct small synchronous wheel as possible or the small synchronous wheel through the motor shaft, the load end of the large synchronous wheel, so as to slow down several times, you can achieve a good effect.This occasion drive parameter (with smaller acceleration settings, such as 5000,), so that the parameters aim to reduce acceleration and deceleration due to large load equivalent inertia.

4. Disk load

This load servo can not directly drive, generally need to connect the reducer.For example, a 200mm diameter weight of 10KG disk.The radius is 100mm, and the weight equivalent radius is 50mm.The arm is very large.If the server is to receive this load, compare the reducer to connect the load.

If the disk is not particularly heavy, some positioning precision and rigidity can be sacrificed to control.Specific method, the motor acceleration is set to a relatively small, for example 1000 the left and right sides.Speed KI is set to 2000 to cancel the integral action.Position KP to 1000. Changing these parameters with general disk load can be available.

5. Automatically find the single-circle origin function

The automatic origin finding function is selected by changing the parameters of the register address 0x19 (special function).If you need to automatically find the origin, set the method such as Next:

The modbus enables you to send the 1

Special function (address 0x19) send 10~32768 (32768 corresponding to 360° for the motor) parameter save send 1

Find the source automatically after reagain.Because it is an absolute value encoder, you can automatically find a circle in any position.(The DIR polarity is either 1 or Person 0 can set the direction of finding the origin)

6. Automatically find the mechanical origin function

The automatic find mechanical origin function is selected by changing the parameters of the register address 0x19 (special function).If you need to automatically find the mechanical origin, the setting method is as follows:

The modbus enables you to send the 1

Special feature (address 0x19) Send 1 (the mechanical origin is automatically found immediately)

Parameter save Send 1(The need to automatically find the mechanical origin can be achieved by saving this parameter)

After recharging, it automatically reverses to the motor block, and then the motor reverses 36° as the origin.(The DIR polarity is 1 or 0 can set the direction of finding the origin)

7. Automatic origin-finding function (origin switch)

The automatic find mechanical origin function is selected by changing the parameters of the register address 0x19 (special function).If you need to automatically find the origin, set it as follows:

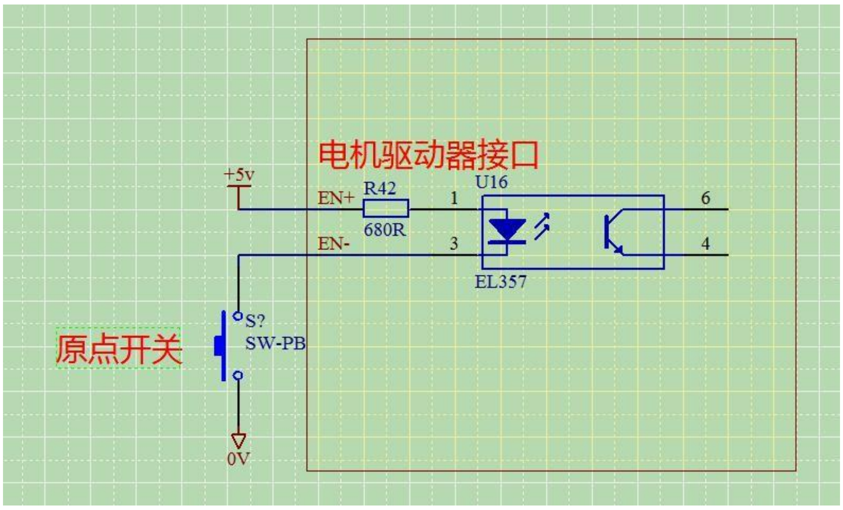
The modbus enables you to send the 1

Special feature (address 0x19) Send 8 (the mechanical origin is automatically found immediately)

Parameter save Send 1(The need to automatically find the mechanical origin can be achieved by saving this parameter)

After re-charging, it will automatically reverse to the EN signal, and then the motor positively turns the encoder origin as the origin.(The DIR polarity is 1 or 0 can set the direction of finding the origin)

The origin switch requires an EN signal.The wiring diagram is as follows:



8. Communication mode clear location

Clear the absolute position: If the absolute position needs to be cleared to 0 during operation, the electronic gear molecule sends 0 first (the electronic gear is invalid in communication mode for this special function. If the communication control can be directly saved to the electronic gear molecule into 0), and then the absolute position (0x 16) sends 0, clear directly to the absolute position 0.

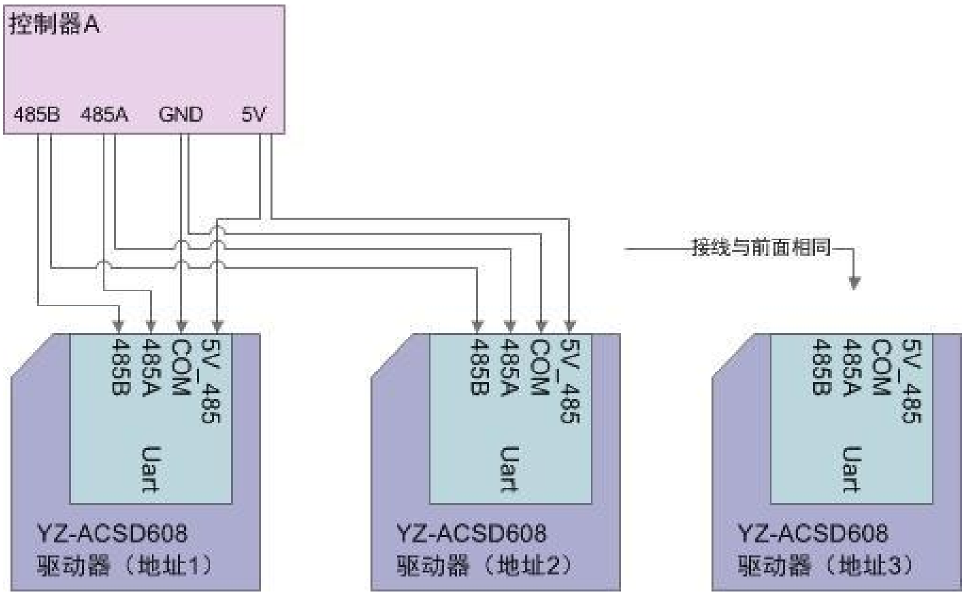
Emergency stop: In communication mode, if many pulses need to go, need to stop. First electronic gear molecular send 0 (electronic gear is invalid in communication mode for this special function. If the communication control can be stored directly into the electronic gear molecule into 0, and the target position (0x0C) sends 0 to 0, it can be stopped urgently. Emergency stop also has a small amount of deceleration distance, and the length of the deceleration distance is controlled by the position loop KP.

9. Power default communication control

As long as the electronic gear molecule is set to 0, modbus enables the default to 1.

V. Modbus control mode

1. Hardware connection



Inside the drive 485 are all isolated by optical coupling, solving the problem of connectin g multiple attendants prone to interference and damage.

2. register declaration

The drive can control the drive via the modbus (RTU mode).The host can set the drive parameters and control the run through the modbus's read and write register function.The drive-supported function codes are 0x3 (read register), 0x6 (write register), 0x78 (write target location), and 0x7a (modify the device address).

The register list is as follows:

address	Parameter name	Read / read and write only	Parameter range	parameter
0x00	The Modbus enables the	read-write	0~1	0: The modbus is prohibited 1: The modbus enables the power
0x01	Drive output enables the	read-write	0~1	0: Drive output is prohibited 1: Drive output enables
0x02	Motor target speed	read-write	0~3000 r/min	Speed mode, target speed Maximum speed when in position mode
0x03	Motor acceleration	read-write	0~60098 (r/min)/s	When the parameter is less than 60000, the acceleration and deceleration curve occurs inside the drive. When the parameter is equal to 60000, no acceleration process, the deceleration size determines the Dayu 60000 to 60098 according to the position KP, individual bits and 100~98 corresponding to the position feedforward. 0%~98%。The larger the position feedforward, the smaller the follow-pulse lag.
0x04	Weak magnetic Angle	read-write	0~306 r/min	Internal parameters do not require additional settings
0x05	Speed ring proportional ratio coefficient	read-write	0~10000	Represents 0.0~10.0 The greater the value, the stronger the rigidity Individual bit is even: pulse input polarity is effective at the open moment Each bit is odd: pulse input polarity is effective at the pass-on moment
0x06	Speed loop integration time	read-write	2~2000 ms	Integration time of 2 ~ 2,000 ms The smaller the value, the stronger the rigidity
0x07	Position ring proportional coefficient	read-write	60~30000	Position KP, the greater the values, the stronger the rigidity One bit is even: alarm output often open (normal often open, alarm often closed) one bit is odd: alarm output often closed (normal often closed, alarm often open)
0x08	Speed feed	read-write	0~12.0V/KRPM	327 Represents 1V / KRPM and is not self-set
0x09	DIR polarity	read-write	0~1	0: The external DIR does not rotate through clockwise way

				1: External DIR guide rotates clockwise
0x0A	Electronic gear molecules	read-write	0~65535	A 16-bit electronic gear molecule If the electronic gear molecule is 0, special functions can be realized in the details described above
0x0B	Electronic gear denominator	read-write	1~65535	The 16-bit electronic gear denominator
0x0C	Target location is 16 bits lower	read only		High 16 higher steps required
0x0D	Target position is 16 bits high	read only		Low 16 steps are required
0x0E	Alarm code	read only		
0x0F	System current	read only	0~32767	Actual current is x / 2000 (A)
0x10	Motor current speed	read only	-30000~30000 r/min	Actual motor speed = the current motor speed / 10
0x11	system voltage	read only	0~32767	Actual voltage is x / 327 (V)
0x12	System temperature	read only	0~100	degree Celsius
0x13	The PWM of the system output	read only	-32768~32767	Representative for-100%~100%
0x14	Parameter saving flag	read-write	0~1	0: Parameter is not saved 1: Save the parameters in 2: Save it
0x15	device address	read only	0~255	device address
0x16	Absolute position is 16 bits lower	read-write		Walk through 16-high steps
0x17	Absolute position is 16 bits higher	read-write		Walk a low 16 steps
0x18	Still maximum allowed output	read-write	0~609	0~609 Corresponding to the allowable maximum output of 0~60.9% bits 1~9 corresponding to the blocking transfer alarm Between.The 1 bit 0 block turn does not alarm, and the automatic output is reduced to 1 / 2 of the original after the blocking turn 3S.

0x19	specific function	read-write	0~100	0: Pulse + direction mode 1: Automatically find the mechanical origin and turn 36° (automatically reverse to mechanical zero and stop walking 36°) 2: The Encoder follows the mode 3: Speed mode, duty-cycle speed adjustment (10%~90% corresponding to 0~1000RPM) 10~32768: The Angle of the automatic transfer algorithm is: X * 360° / 32768 4: Automfinds mechanical origin and is turning to encoder zero (upper power automatically reverses to mechanical zero and stops to encoder zero) 5: Z_OUT simulation origin signal mode: the pulse of the feed motor control motor is slowly reversed until the blocking, Zout output the communication signal, and then the controller feed motor pulse is transferred to the ZOUT signal disappears is the system origin (Zout disappears after zero point of an encoder to ensure the accuracy of the origin) 8: Automatically find the origin (the origin is connected to the EN signal).Uppower automatically reverses to the EN There is a signal, then turn to EN no signal stop.

3. Modbus communication format

A. modbus host read data and slave response format (function code 03)

Host reads the data form							
device address	FC	The first to send, the memory is high bit address	The first to send, the memory is low bit address	High number of registers	The number of registers is low	CRC tall Position	CRC low Position
0x01	0x03	0x00	0x00	0x00	0x01	0x84	0x0a

Answer from machine						
device address	FC	DL	First, one, one, the number According to the high bytes	First, one, one, the number According to low bytes	CRC high-order	CRC low-order
0x01	0x03	0x02	0x00	0x01	0x79	0x84

The data received by the serial port is unsigned. If the register is signed number, the send is binary complement format. The algorithm converted into signed number is as follows (VB code):

```
If modbus.data(11) > 32767 Then
    disp_modbus_data.PU = (modbus.data(11) - 32768) * 65536 + modbus.data(10)
    disp_modbus_data.PU = -((&H7FFFFFFF - disp_modbus_data.PU) + 1)
Else
    disp_modbus_data.PU = dmodbus.data(11) * 65536 + modbus.data(10)
End If
```

Note: modbus.The data (11) is 16 bit modbus.The data (10) is 16 bits below the target position

The b.modbus Host write data and slave response format (function code 06)

Host writes data form							
device address	FC	The first send High-level address of the memory	The first send Low address of the memory	Data high	Data low	CRC tall Position	CRC low Position
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

Answer from machine form							
device address	FC	The first send High memory bit address	The first send Low memory bit address	Data high	Data low	CRC tall Position	CRC low Position
0x01	0x06	0x00	0x00	0x00	0x01	0x48	0x0a

Write pulses of c. modbus host (function code 0x10)

Host write double-byte data (write PU pulses)						
device address	FC	First, one, one, send High-level address of the memory	First, one, one, send Low address of the memory	High number of registers	The number of registers is low	DL
0x01	0x10	0x00	0x0c	0x00	0x02	0x04
PU: 8~15 bits	PU: 0~7 bits	PU:24~31 Position	PU:16~23 Position	CRC high-order	CRC low-order	
0x27	0x10	0x00	0x00	0xf8	0x8b	

The number of pulses is the signed number, and a negative number (assuming X) algorithm converted to 32-bit 16 decimal numbers is as follows (vb code):

```
If X < 0 Then
    X = &H7FFFFFFF + (X + 1)
    PU24_31 = Fix(X / (256 * 65536)) + &H80
Else
    PU24_31 = Fix(X / (256 * 65536))
End If
    PU16_23 = Fix(X / 65536) mod 256
    PU8_15 = Fix(X / 256) mod 256
    PU0_7 = X mod 256
```

Note: The fix () takes the whole function

Answer from machine form							
device address	FC	High level of the first register bit address	Low of the first register bit address	High number of registers	The number of registers is low	CRC tall Position	CRC low Position
0x01	0x10	0x00	0x0c	0x00	0x02	0x81	0xcb

Write incremental pulses for d.modbus host (special function code 0x78)

Host special function code 0x78 Format (Write of PU pulses)							
device address	FC	PU:24~31. Position	PU:16~23. Position	PU:8~15 Position	PU:0~7 Position	CRC tall Position	CRC low Position
0x01	0x78	0x00	0x00	0x27	0x10	0xbb	0xfc

Answer from machine form							
device address	FC	PU:8~15 Position	PU:0~7 Position	PU:24~31. Position	PU:16~23. Position	CRC tall Position	CRC low Position
0x01	0x78	0x27	0x0e	0x00	0x00	0xca	0xb7

The e. modbus host write absolute location (special function code 0x7b)

Host special function code 0x78 Format (Write of PU pulses)							
device address	FC	PU:24~31. Position	PU:16~23. Position	PU:8~15 Position	PU:0~7 Position	CRC tall Position	CRC low Position
0x01	0x7b	0x00	0x00	0x27	0x10	0xff	0xfc

Answer from machine form							
device address	FC	PU:8~15 Position	PU:0~7 Position	PU:24~31. Position	PU:16~23. Position	CRC tall Position	CRC low Position
0x01	0x7b	0x27	0x10	0x00	0x00	0xee	0xb1

4. CRC check sample code

```
unsigned short CRC16(puchMsg, usDataLen)
unsigned char * puchMsg; / * The number of bytes in the
message * / unsigned short usDataLen * to be CRC; / *
message is */
{
unsigned char uchCRCHi = 0xFF; / * High CRC Bytes initialization
* / unsigned char uchCRCLo = 0xFF; / * Low CRC Bytes initialize
the index */ in the * / unsigned uIndex; / * CRC cycles
while (usDataLen- -) / * Transport message buffer */
{
uIndex = uchCRCHi ^ * puchMsgg + +; / * to calculate CRC
* / uchCRCHi = uchCRCLo ^ auchCRCHi [uIndex};
uchCRCLo = auchCRCLo [uIndex];
}
return (uchCRCHi << 8 | uchCRCLo) ;
}
/* CRC High Byte value table */
```

```
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80,
0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00,
0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;
/* CRC Low Byte value
table * / static char
auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D,
0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B,
0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16,
0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6,
0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39,
0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61,
0xA1,  0x63, 0xA3,  0xA2, 0x62,  0x66, 0xA6,  0xA7, 0x67,  0xA5, 0x65,  0x64, 0xA4, 0x6C,  0xAC,  0xAD, 0x6D, 0xAF,
0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A,
0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72,
0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58,
0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C, 0x44, 0x84,
0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40 }
```

5. m odbus mode host control process

a: position mode

The OFF via the dial switch SW1 is in position mode. The power can first set the following parameters by our providing the upper computer software:

1. Modbus enables to send 1 (only modbus enables 1 can change other parameters and the external pulse signal is invalid.)

   HEX source command: 01060000000001480 A
2. Motor acceleration transmission 5000 (set acceleration without setting,

20000 default parameter) HEX source command: 01 06 00 03 13 88 74 9C

3. Target speed Send 1500 (set the running speed according to the actual operation needs, namely, use the default parameter 2800) HEX source command: 01 06 00 02 05 DC 2A C3

4. The electronic gear molecule sends 0 (after the electronic gear molecule is saved to 0, the next power mdobus can default is 1) HEX source command: 01 06 00 0A 00 00 A9 C8

5. Parameter save flag sent 1 (after this parameter the previously set parameters are saved internally)
HEX source command: 01 06 00 14 00 01 08 0E

6. Power up again to see if the parameters have been saved correctly.The above settings only need the provided, and the HEX source does not need to be sent through the serial port itself.

After the parameters are set, you can issue the location command through PLC or single chip, or your own computer software.The location command only needs to pass the 0x10 command to send the location to go.

1. The incremental position (the meaning of the incremental position is the data where the motor needs to go forward or backward) e. g., a walk forward (assuming the motor encoder is 1000 line encoder, 4000 pulses is 4000) HEX source command: 01 10 00 0C 00 02 04 0F A0 00 00 F0 CC

For example, a binary circle forward (assuming the motor encoder is 1000 line encoder and the number of pulses is-4000) -4000 is calculated as follows: the binary method of 4000 is 00 00 0F A0. (Note: 0= FF FF FF FF + 1)

-4000 is 0-00 00 0F A0 =FF FF FF FF-00 00 0F A0 + 1=FF FF F0 5F + 1 = FF FF F0 60
HEX Source command: 01 10 00 0C 00 02 04 F0 60 FF FF C1 54

2. Absolute absolute position (the meaning of absolute position is that the position is defined as 0 after the power or absolute position is clear 0 or automatically find the origin, the absolute position is to go to the new position, such as the first hair 4000 to walk a circle, the second hair has gone to the position of 4000 again, and then send the same command
Motor does not go)

For example, the motor is required to go to the 2 lap position (assuming the motor encoder is 1000 line encoder, 2 loop pulses is 8000) HEX source command: 01 10 00 16 00 02 04 1F 40 00 00 74 89

For example, the motor is required to go back to the origin (when the electronic gear molecule is 0, the send 0 is to clear the current position, so go back to the origin to send 1, when a pulse does not affect the precision)

HEX source command: 01 10 00 16 00 02 04 00 01 00 00 23 49

Note: the control motor only needs to send the required position first (try to use the absolute position instruction as far as possible, because you can repeat many times, still go to the same position), and then you can read the absolute position contrast to the setting position, to determine whether to execute the next instruction (pay attention to the judgment needs to allow + -2 error).Or by connecting the PF signal, the drive will give a photocoupled output of the switch amount signal.

Read the absolute position instruction is as follows: 01 03 00 16 00 02 25 CF

6. Modify the Porter rate

Modifying the Porter rate can be sent through the epistaster software we provide.Send it specifically as follows:

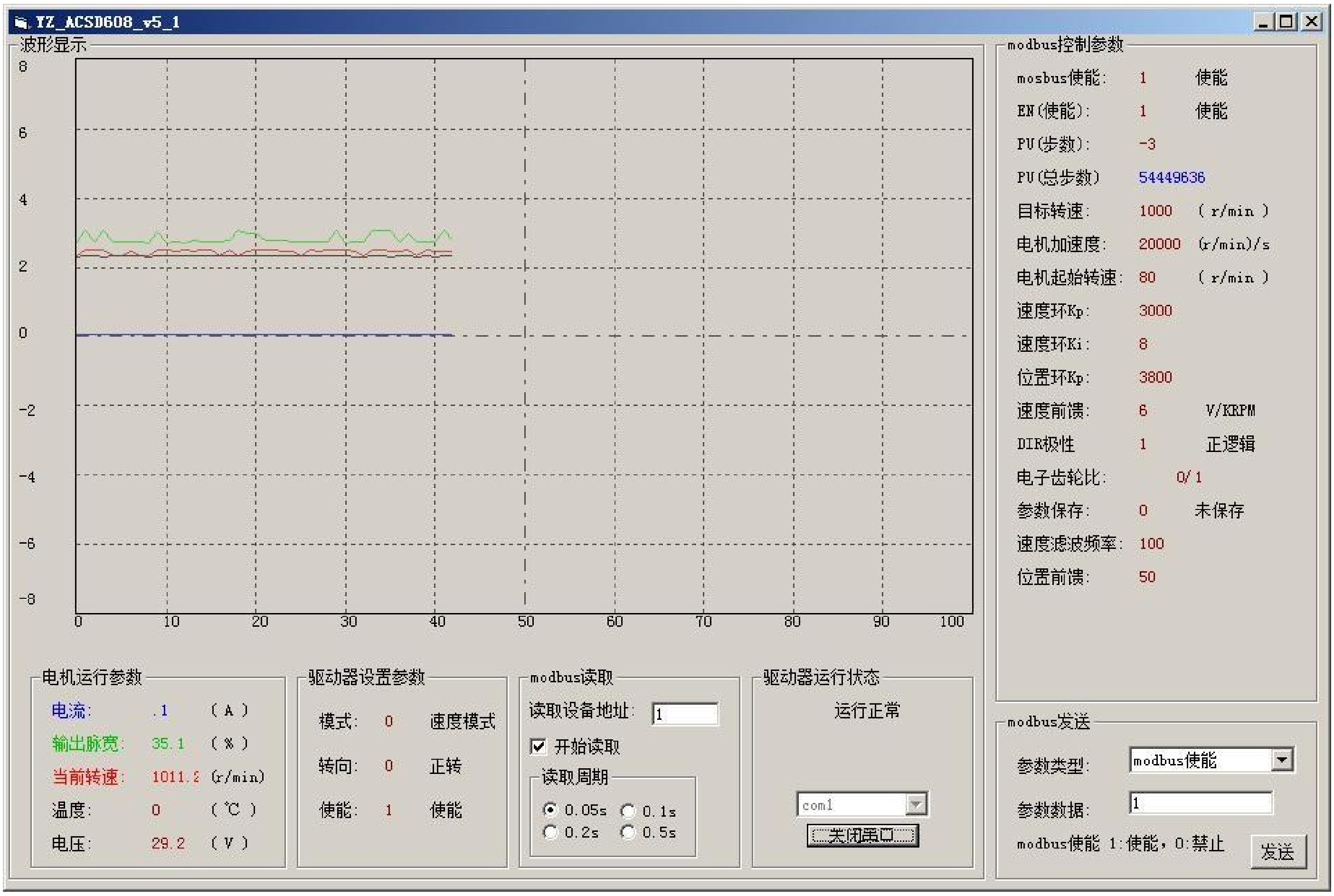
A.Modbus enabling (Address 0)	transmit by radio	1
B.Motor acceleration (Address 2)	transmit by radio	803 (Note 802:38400 801:19200 800:9600）803:115200
C.Weak magnetic Angle (Address 3)	transmit by radio	129
D.Modbus enabling (Address 0)	transmit by radio	506

Effective after recharging

Note: Send parameter saving is not required because this is the internal parameter.Just send it to strictly follow the steps above.

six. Instructions for upper computer software

This drive provides an epistasis software for monitoring and testing the drive.You can view and set the internal drive parameters through the software.



电机运行参数
电流: .1 (A)
输出脉宽: 35.1 (%)
当前转速: 1011.2 (r/min)
温度: 0 (℃)
电压: 29.2 (V)

驱动器设置参数
模式: 0 速度模式
转向: 0 正转
使能: 1 使能

modbus读取
读取设备地址: 1
☑ 开始读取
读取周期
● 0.05s ○ 0.1s
○ 0.2s ○ 0.5s

驱动器运行状态
运行正常
com1
关闭串口

modbus控制参数
mosbus使能: 1 使能
EN(使能): 1 使能
PU(步数): -3
PU(总步数) 54449636
目标转速: 1000 (r/min)
电机加速度: 20000 (r/min)/s
电机起始转速: 80 (r/min)
速度环Kp: 3000
速度环Ki: 8
位置环Kp: 3800
速度前馈: 6 V/KRPM
DIR极性 1 正逻辑
电子齿轮比: 0/1
参数保存: 0 未保存
速度滤波频率: 100
位置前馈: 50

modbus发送
参数类型: modbus使能
参数数据: 1
modbus使能 1:使能，0:禁止
发送

As shown in the figure above, the software is divided into several parts, such as waveform display and motor operation parameters.The following describes the functions and functions of each part.Waveform display: There are 4 channels, respectively represented in 4 colors.Color and font face in the motor operation parameters, The same color.That is: blue indicates the current, green indicates the pulse width of the output, red indicates the current speed, and black indicates the voltage.Motor operation parameters: represent the real-time data of the motor operation.

Drive setting parameters: display the dial switch of the drive, and direction enabling setting.This bar is invalid if it is in modbus mode.

Drive running status: This bar displays the alarm status of the drive and displays normal operation if there is no alarm.

Modbus Control Parameters: The parameters in this bar are the parameters within the drive, and if you want to modify them, you must be able to write 1 to the modbus first. Referring to the register description.

Modbus Read: This bar sets the drive address, the cycle of read drive data, and whether or not.

Modbus Send: This bar is used to modify the drive parameters, first select the parameter type, then set the parameter data, and then point to send it.