DC Integrated Torque Servo

YZ-AIM-EasyCan_v3.5

I）. Product Features

1. Isolated CAN communication (EasyCan protocol, easy, fast to get started, rate 1M). Support contour position mode, and, cycle synchronization mode.
2. 15-bit absolute encoder with up to 32768 pulses per revolution.
3. Multi-turn absolute value (requires battery). Pulse mode: Automatically return to power-off position after power-on again. Communication mode: Power-off record position.
4. Multi-stage dd motor structure, high torque output.
5. Integrated servo, simplified wiring, ultra-small size.
6. Low noise, low vibration, high-speed positioning, and high reliability.
7. foc field oriented vector control, support position / velocity closed loop.
8. It can work in a given pulse state with zero lag, following zero lag.
9. 16-bit electronic gear function.
10. Provide a serial host computer to monitor the motor status and modify parameters.
11. Position mode, support pulse + direction signal, encoder follow
12. Speed mode, support pwm duty cycle signal speed regulation
13. With stall, over-current protection, over-voltage protection.

Parameter table

Model parameters		42AIM15C	42AIM10C
power supply	Voltage	24VDC±10%	24VDC±10%
	Current	2.2A	1.6A
Motor parameters	Torque	0.48NM	0.33NM
	Rated speed	1000RPM	1000RPM
	Speed	1500RPM	1500RPM
	power	50W	35W
Feedback signal		Single-turn 15-bit magnetic encoder (single-turn 32768 pulses)	
cooling method		Free cooling	
weight			
Position control mode	Maximum input pulse frequency	500KHz	
	Pulse command mode	Pulse + direction, phase a + phase b	
	Electronic gear ratio	Setting range 1 ~ 65535 than 1 ~ 65535	
	Position sampling frequency	2KHz	
Protective function		Stall alarm	
Communication Interface		Easycan (CAN communication, speed 1M) Serial ttl (19200, 8, n, 1) (monitor motor status and modify parameters).	
Use environment	Ambient temperature	0~40°	
	Motor maximum temperature	85°	
	humidity	5~95%	

Model parameters		57AIM15C	57AIM15CH	57AIM30C	57AIM30CH
power supply	Voltage	24~36VDC	24~36VDC	24~36VDC	24~36VDC

	Current	2.2A	2.2A	4.4A	4.4A
Motor parameters	Torque	0.48NM	0.24NM	0.96NM	0.48NM
	Rated speed	1000RPM	2000RPM	1000RPM	2000RPM
	Speed	1500RPM	2500RPM	1500RPM	2500RPM
	power	50W	50W	100W	100W
Feedback signal		Multi-turn absolute encoder (32768 pulses per turn, 15 bits per turn)			
cooling method		Free cooling			
weight					
Position control mode	Maximum input pulse frequency	500KHz			
	Pulse command mode	Pulse + direction, phase a + phase b			
	Electronic gear ratio	Setting range 1 ~ 65535 than 1 ~ 65535			
	Position sampling frequency	2KHz			
Protective function		Stall alarm			
Communication Interface		Easycan (CAN communication, speed 1M) Serial ttl (19200, 8, n, 1) (monitor motor status and modify parameters).			
Use environment	Ambient temperature	0~40°			
	Motor maximum temperature	85°			
	humidity	5~95%			

Model parameters		60AIM25C	60AIM25CH
power supply	Voltage	36VDC±10%	36VDC±10%
	Current	7A	7A
Motor parameters	Torque	2NM	1NM
	Rated speed	1000RPM	2000RPM
	Speed	1500RPM	2500RPM
	power	200W	200W
Feedback signal		Single-turn 15-bit magnetic encoder (single-turn 32768 pulses)	
cooling method		Free cooling	
weight			
Position control mode	Maximum input pulse frequency	500KHz	
	Pulse command mode	Pulse + direction, phase a + phase b	
	Electronic gear ratio	Setting range 1 ~ 65535 than 1 ~ 65535	
	Position sampling frequency	2KHz	
Protective function		Stall alarm	
Communication Interface		Easycan (CAN communication, speed 1M)	
		Serial ttl (19200, 8, n, 1) (monitor motor status and modify parameters).	

Use environment	Ambient temperature	0~40°
	Motor maximum temperature	85°
	humidity	5~95%

2) Drive interface

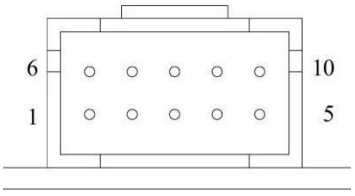
1. Power and control signal interface

Terminal number	name	Features
1	+24V	Positive DC power supply, + 24v. Reverse connection will directly short the power supply and may damage the driver.
2	GND	DC power ground. Positive and negative connection will directly short-circuit the power supply and may damage the driver.
3	PU+ (+5V)	Pulse control signal: the rising edge of the pulse is valid; pu- high level 3.3 ~ 5v, low level 0 ~ 0.5v. In order to respond to the pulse signal reliably, the pulse width should be greater than 1.2μs. If + 12V or + 24V is used, a series resistor is required.
4	PU- (PU)	
5	DIR+ (+5V)	Direction signal: high / low level signal.To ensure reliable commutation of the motor, the direction signal should precede the pulse signal. Established at least 5μs. DIR- 3.3 ~ 5V at high level, 0 ~ 0.5V at low level.
6	DIR- (DIR)	

Terminal serial number: facing the terminal, the left is the first.

The aim series adopts a differential interface circuit that can be used for differential signals, single-ended common negative and common positive interfaces, and a built-in high-speed photocoupler that allows receiving signals from long-line drivers, open collectors, and pnp output circuits.

2. Communication and output interface



Terminal serial number: facing the terminal, the lower row is 12345 from left to right, and the upper row is 6 7 8 9 from left to right.

Terminal number	name	Features
1	CANL	Can communication port, CAN communication needs to supply CAN_5V, COM 5V
2	RX	Driver serial port (ttl level)
3	TX	Drive serial port (ttl level)
4	CANH	Can communication port, CAN communication needs to supply CAN_5V, COM 5V
5	GND	Serial gnd
6	COM	The output signal is common ground with 485 power.
7	WR	Alarm signal output, internally optocoupler npn output. Normally, it is in

		high-impedance state, and it will be conductive with com during alarm.
8	RDY	Servo ready signal. The optocoupler npn outputs the ON signal after the servo works normally, and the battery power supply is in a high impedance state after power off.
9	ZO	Multi-turn encoder zero output. Encoder position is greater than 0: npn output conduction signal, encoder position is less than 0: npn Non-conducting
10	CAN_5V	485 communication 5v power supply, requires external power supply. (This power supply is supplied by the controller)

3. Status indication and alarm

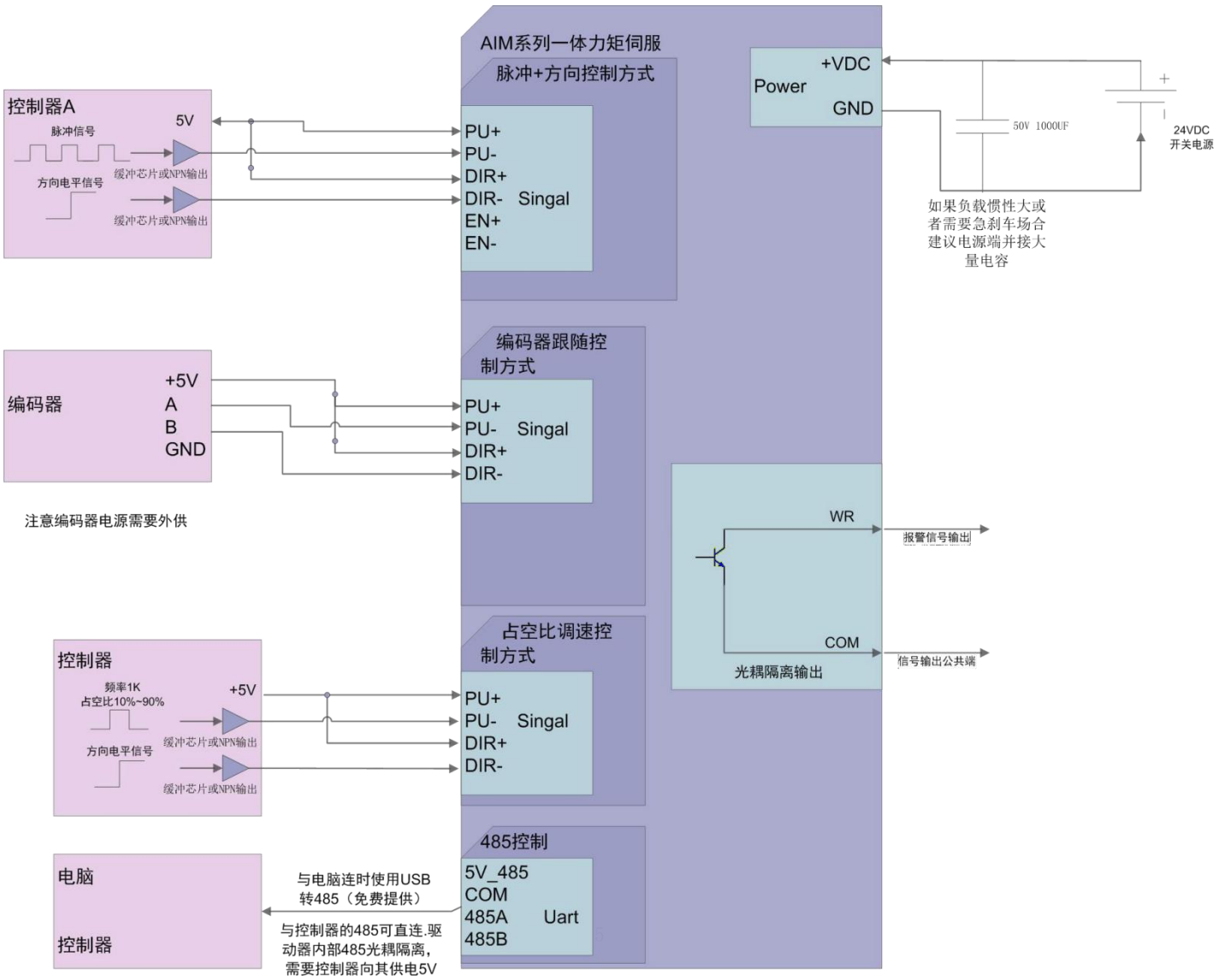
After power on, the red and green lights are on once to check whether the LED is working normally. Then the green light is on and the red light is off. If you encounter an alarm state, you can determine the cause by flashing red, or you can read the alarm code through modbus .

Alarm code	Flashing red	Reason for alarm	Alarm processing
0x14	One long flash and four short flashes	Stall alarm	Downtime

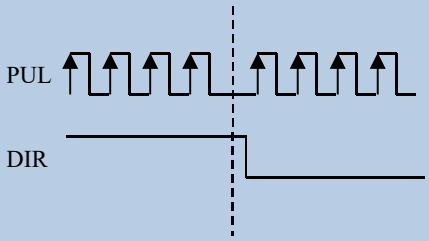
Note: Stall alarm, stall time can be set, see the register description for details.

3）. Driver wiring diagram and control method

1. Driver typical wiring diagram



2.Command pulse + direction position control mode

Pulse	CCW	CW	Parameter
Pulse			Command pulse

If you need 3200 pulses per revolution

Electronic gear is set to 32768 (Number of pulses per encoder cycle) 3200 (number of pulses to be set per revolution) After approximation: 256 to 25

If 8192 pulses are required (default parameter)

Electronic gear is set to 32768 (Number of pulses per encoder cycle) 8192 (number of pulses to be set per revolution) After approximation: 32768 than 8192

After approximation: 4 to 1

Note: Can be reduced as much as possible. The electronic gear molecule is 32768. The value is too large, which will affect the following performance.

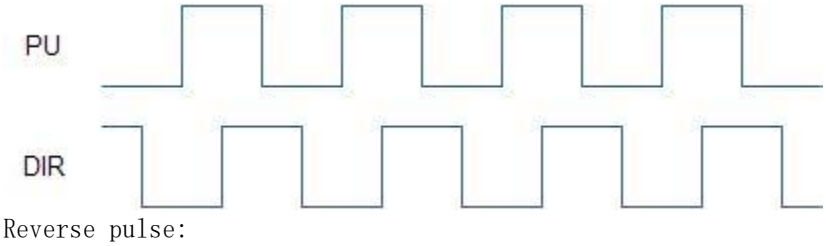
Command pulse frequency = (requires the rotation speed of the motor / 60) * The number of pulses per revolution For example: Need to click 1000rpm The number of pulses per revolution is 8192

Pulse frequency = 1000/60 * 8192= 136533HZ

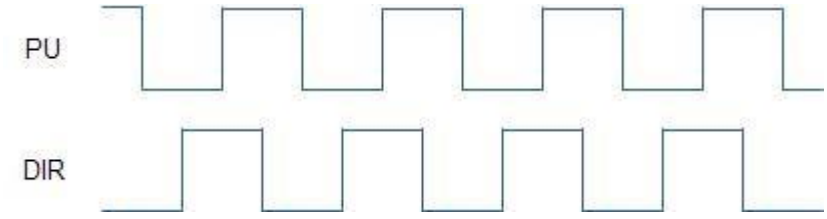
3. Quadrature command pulse position control mode

By setting the special function (0x19 address) to 2, after the power is turned on again, it is the encoder following mode. This mode can be used for encoder following. For example, if an encoder is connected to a shaft, the encoder output is connected to the driver (wiring (Such as the typical wiring diagram of the driver), the driver can control the servo motor, and follow the control encoder according to the input encoder signal. You can set the ratio of the control encoder and motor rotation angle by adjusting the electronic gear.

Forward rotation pulse:



Reverse pulse:



The direction of motor rotation: pu rising edge leads dir rising edge is forward rotation.pu rising edge lags dir rising edge is reverse rotation.

4）. Parameter debugging

Depending on the load connected to the motor, the parameters need to be adjusted to achieve the best results.

1. Internal acceleration / deceleration curve

Select whether to use the internal acceleration / deceleration curve according to the output signal of the controller. Use the internal acceleration curve:

When the motor acceleration is less than 60000, the driver will enable the internal acceleration and deceleration curve, and the specific acceleration will be the same as the set value.

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

Disable internal acceleration curve:

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

Use occasions: For example, the engraving machine, the pulse output by the controller has acceleration and deceleration, there is no need for the acceleration curve inside the driver.

2. Screw load

First introduce the torque, first use a 400W motor, 1.3NM. The load is a screw with a 5mm pitch, that is, the motor shaft rotates one load to move 5mm. In this case,

Load Equivalent Arm = 5mm / 3.14 = 1.592 mm

The thrust that the motor can provide is

Thrust through screw drive = 1.3NM / (1.592mm * 0.001) = 816 N

The weight that can push the load is about 80kg, this is vertical, and the flat push can be slightly larger.

Because the screw rod load motor moves a short distance for one revolution, the parameters of the drive (acceleration can be large, such as 20000, and the position loop kp can be large, such as 3000). The servo motor is most suitable for this kind of load.

3. Pulley load

The servo motor is actually not very suitable for this kind of load. Because the pulley is generally larger in diameter, for example, 30mm in diameter. Then the motor moves once, and the

distance the load moves is 30mm * π = 94.2, which is many times larger than the screw 5mm mentioned above. .

The thrust that the motor can provide is

Thrust through belt drive = 1.3NM / (30mm * 0.001) = 43.3 N

The weight that can push the load is about 4.3kg. So the servo motor is not suitable for connecting the synchronous wheel, because the synchronous wheel rotates the load for a long distance and the arm is long. If you want to use a servo motor in this case, you can choose to directly Try to make the synchronous wheel as small as possible or connect the small synchronous wheel through the motor shaft, and connect the large synchronous wheel to the load. This can reduce the speed several times to achieve better results. In this case, the driver parameters (the acceleration setting is small, such as 5000,) The purpose of setting parameters is to reduce acceleration and deceleration, because the equivalent inertia of the load is large.

4. Disc load

This kind of load servo cannot be driven directly, and generally requires a reducer. For example, a disk with a diameter of 200mm and a weight of 10KG. The radius is 100mm and the equivalent weight is 50mm. The arm is very large. Reducer is then connected to the load.

If the disc is not particularly heavy, you can sacrifice some positioning accuracy and rigidity to control. For specific methods, set the motor acceleration to a relatively small value, such as 1000 Left and right. The speed ki is set to 2000, the integral effect is cancelled. The position kp is changed to 1000. These parameters can also be used for general disk loads.

5. Automatic origin finding function

The function of automatically finding the origin is selected by changing the parameter of register address 0x19 (special function). If you need to find the origin automatically after power on, the setting method is as follows under:

modbus enable send 1

Special function (address 0x19) send 10 ~ 32768 (32768 corresponds to 360 ° of the motor) Parameter save send 1

After the power is turned on, it will automatically find the origin. Because it is an absolute encoder, it can automatically find any position in a circle after power on. (Dir polarity is 1 or (Person 0 can set the direction to find the origin)

6. Automatically find mechanical origin function

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

modbus enable send 1

Special function (address 0x19) send 1 (at this time, the machine origin will be automatically found)

Parameter save and send 1 (Need to re-power on to find the mechanical origin automatically, you can save this parameter)

After the power is turned on again, it will automatically reverse to the motor stall, and then

the motor reverses 36 ° as the origin. (The polarity of dir is 1 or 0, you can set the direction to find the origin)

7. Communication method clear location

Clear absolute position: If the absolute position needs to be cleared to 0 during operation, first the electronic gear molecule sends 0 (the electronic gear is invalid in communication mode, used for this special function. If the communication control can directly save the electronic gear molecule as 0), then Send 0 to the absolute position (0x16), and clear the absolute position directly.

Emergency stop: In the communication mode, if there are many pulses left to go, an emergency stop is required. First, the electronic gear molecule sends 0 (the electronic gear is invalid in the communication mode, which is used for this special function. If the communication control can directly the electronic gear molecule Save it to 0, and then send 0 to the incremental position (0x0C), you can emergency stop. There is also a small amount of deceleration distance in the emergency stop. The length of the deceleration distance is controlled by the position loop KP.

8. Power-on default communication control

As long as the electronic gear numerator is set to 0, after saving, power on again, the modbus enable default is 1.

9. Multi-turn absolute value instructions (with battery solution)

a. Set the multi-turn position.

Before the motor is installed, the motor is connected to the battery, power supply and communication lines. Use the host computer software to send 0 to the absolute position to let the motor go to the origin first. At this time, install the motor to the system, which corresponds to the origin of the system.

If the origin is not set during installation, the single-turn origin of the motor is the origin of the encoder and cannot be set. Multi-turn return to zero can be cleared by turning off the power and battery at the same time.

Note: After using the multi-turn absolute value function, the clear absolute position command cannot be used, because the clear absolute position command only clears the position register inside the microcontroller and cannot clear the multi-turn information inside the encoder. If you want to reset the zero point, you need to press Repeat the steps above.

b. How to use the multi-turn function in the communication control mode.

After the origin position is set by method (a), the current position of the motor can be known each time the power is read through communication to read the absolute position.

c. Control how to use the multi-turn function through pulse control.

Method 1: You can connect the motor to output the zo signal. When the motor position is greater than the zero position, the zo signal is always output. When the motor position is less than the zero position, the zo signal is always output. Continuity, reverse to zo non-conduction is zero. Just power on, the controller reads zo, when zo is not conductive, forward rotation to zo conduction to zero.

Method 2: The special function of the motor parameters can be saved as 6. This way, each time it

is powered on, it will automatically return to the set origin.

10. Multi-turn absolute value brake instructions

A. Multi-turn origin setting method:

First control the motor rotation to the original position of the device through the host computer provided by us. (Mdobus enable, send 1 first, then send pu steps to control the motor rotation. turn)

At this time, the parameter is saved and sent 3 (the role is to save the single-turn origin, and the parameter will be displayed 2 after saving).

Unplug the motor power plug directly (note that the power to the motor terminals can only be directly turned off, and the switching power supply cannot be indirectly powered off by disconnecting the 220v power).

Position, the motor will automatically record as the origin. When the 220V switching power supply is cut off, the voltage drops slowly because the capacitor of the switching power supply can store energy, and the motor detects the power down process and will automatically save the current position. Automatically read and update to the absolute position register. (Note: So if the capacity of the switching power supply capacitor is too small, it may not work properly. In this case, please connect a 1000UF / 63v electrolytic capacitor on the DC output side of the switching power supply).

B. Communication control method:

The first requirement is that the motor is a motor with a brake. First, turn on the multi-turn absolute value function. The settings are as follows:

Modbus enable
transmission 1. Special
function transmission 7.
Parameter save
transmission 1
.Po
wer on again.

When powering on, directly read the value of the absolute position register, which is the position value of the current motor. 32768 is a circle.

C. How to use pulse control:

Method 1:

The first requirement is that the motor is a motor with a brake. First, turn on the multi-turn absolute value function. The settings are as follows:

Modbus enable
transmission 1. Special
function transmission 7.
Parameter save
transmission 1
.Po
wer on again.

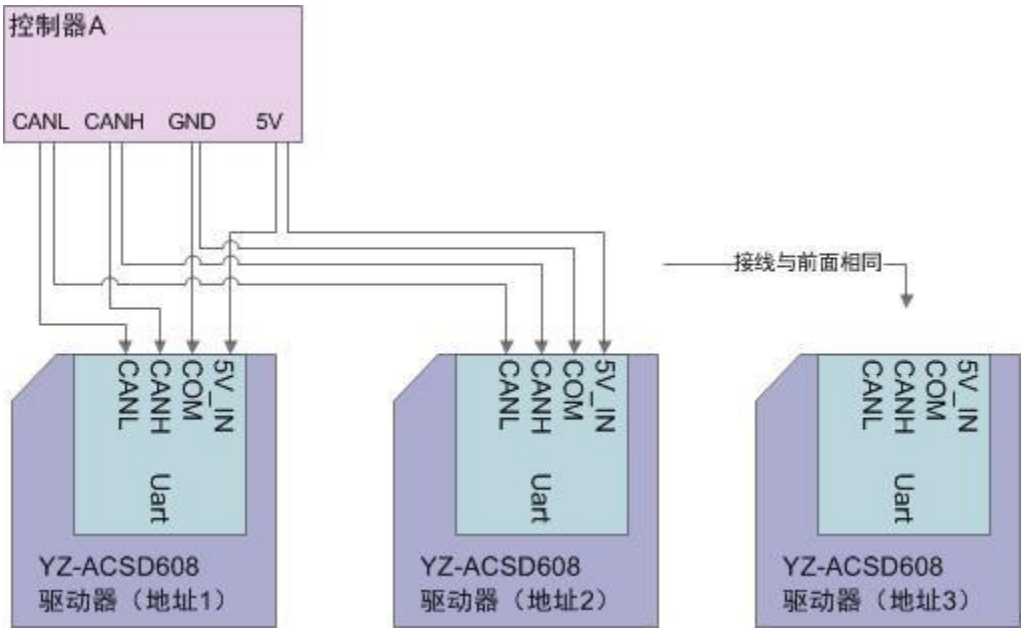
You can output the zo signal by connecting the motor. When the motor position is greater than the zero position, the zo signal is always output. When the motor position is less than the zero position, the zo signal is always output. Reverse to zo does not conduct to zero. Just power on, the controller reads zo, when zo does not conduct, forward to zo conduct to zero.

Method 2: You can save the special function of the motor parameter as 8. In this way, each time the power is turned on, it will automatically go

back to the set origin. Modbus Enable Send 1.
Special function
sending 8.
Parameter saving
sending 1
.P
ower on again.

V. Can communication control mode

1. Hardware connection



The internal 485 of the driver is isolated by optocouplers, which solves the problem that one master is easily interfered and damaged when connecting multiple slaves.

2. Register description

The driver can control the driver through EasyCan. The host can set the driver parameters and control the operation through the read and write register function of can. The register list is as follows:

address	parameter name	Read-only / Read and write	Parameter range	Parameter Description
0x00	Modbus enabled	Read and write	0~1	0: Modbus prohibited 1: modbus enabled
0x01	Driver output enable	Read and write	0~1	0: Drive output is disabled 1: driver output enable
0x02	Motor target speed	Read and write	0~3000 r/min	Target speed in speed mode Speed in position mode

0x03	Motor acceleration	Read and write	0~65535 (r/min)/s	When the parameter is less than 60000, the acceleration / deceleration curve is generated inside the drive, and the parameter is greater than 60000 When no acceleration / deceleration pulse is generated inside the driver
0x04	Field weakening angle	Read and write	0~306 r/min	Internal parameters do not need to be set separately
0x05	Speed loop proportionality factor	Read and write	0~10000	0.0 ~ 10.0
				The larger the value, the stronger the rigidity Units are even: The polarity of the pulse input is valid at the moment of turning off. Units are odd: the polarity of the pulse input is valid at the moment of turning on.
0x06	Speed loop integration time	Read and write	2~2000 ms	Integration time 2 ~ 2000ms The smaller the value, the stronger the rigidity
0x07	Position loop proportionality factor	Read and write	60~30000	Position kp, the larger the value, the stronger the rigidity Single digits are even: alarm output normally open (normally normally open, alarm normally closed) Single digits are odd: alarm output normally closed (normally normally closed, alarm normally open)
0x08	Speed feedforward	Read and write	0~12.0V/KRPM	327 stands for 1v / krpm, no need to set by yourself
0x09	dir polarity	Read and write	0~1	0: External dir does not conduct clockwise rotation 1: external dir turns clockwise
0x0A	Electronic gear molecule	Read and write	0~65535	16-bit electronic gear numerator If the electronic gear molecule is 0, special functions can be achieved.
0x0B	Electronic Gear Denominator	Read and write	1~65535	16-bit electronic gear denominator
0x0C	Incremental lower 16 bits	Read and write		16-bit higher number of steps required
0x0D	16-bit increment position	Read and write		Lower 16 bits of steps 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 = current motor speed / 10
0x11	System voltage	Read-only	0~32767	Actual voltage is x / 327 (V)
0x12	System temperature	Read-only	0~100	Celsius
0x13	Pwm output by the system	Read-only	-32768~32767	-100% ~ 100%

0x14	Parameter save flag	Read and write	0~1	0: Parameters are not saved 1: saving parameters 2: Saved
0x15	Device address	Read-only	0~255	Device address (can communication needs to save the new address to take effect after power-on)
0x16	16-bit lower absolute position	Read and write		16 steps high
0x17	16-bit higher absolute position	Read and write		Lower 16 bits of steps
0x18	Maximum allowable output at standstill	Read and write	0~609	0 ~ 609 corresponds to the allowed maximum output 0 ~ 60.9% unit digits 1 ~ 9 corresponds to the stall alarm time. Single digit 0 does not alarm
0x19	special function	Read and write	0~100	0: Pulse + direction mode 1: Automatically find the origin of the machine and rotate it forward 36 ° (reverse to the mechanical zero automatically after power-on, and walk 36 ° forward to stop) 2: Encoder follow mode 3: Speed mode, duty cycle speed regulation (10% ~ 90% corresponds to 0 ~ 1000rpm) 4: Automatically find the mechanical origin and forward to the encoder zero. 6. With the battery, the power goes back and forth and returns to zero. 7. With brake motor, brake multi-turn absolute value function, power-on multi-turn position update in the absolute position register. 8. With brake motor, multi-turn absolute value of brake function, it will automatically return to the origin when power is turned on. 9. Communication interpolation (no hysteresis) requires 10k of 50% duty cycle on pu input signal

				10 ~ 32768: The angle to automatically turn on after power on, the algorithm is: x * 360 ° / 32768
0x1a	16-bit lower absolute position (can Communication location cache)	Read and write	-32768~32767	Absolute position cache. In synchronous mode, the system reads the cache every 2ms, and Go to this location.
0x1b	16-bit upper absolute position (can Communication location cache)	Read and write	-32768~32767	Absolute position cache, synchronize multiple by writing a synchronization signal, or changing the dir state The motors run simultaneously.
0x1c	Can communication synchronization control word	Read and write	0~65535	0: Disable the synchronization mode. When it is greater than 0, the internal 2ms time base will be synchronized and the synchronization mode is turned on. When the value is 255, the synchronization is synchronized by the DIR signal. number.
0x1d	Maximum allowable current	Read and write	0~10009	Unit mA, corresponding to the maximum allowable working current of 0 ~ 10.009a Units 1 ~ 9, corresponding to the maximum current 1 ~ 9 seconds, it will alarm and stop When the unit digit is 0, the maximum current is reached and the maximum current is maintained without alarm.

3. EasyCan communication format

The EasyCan communication protocol uses a standard data frame format. The data frame format is shown below:

SOF: Frame interval.

Identifier: The range is 0 ~ 255, which indicates the address of the target device. The master sends data to the slave, and the identifier is the slave address. When the slave sends data to the host, the identifier is the master address.

RTR: 0: Data frame. (1: Remote frame). EasyCan only uses data frames.

r1 : 0: Standard identifier. (1: Extended identifier). EasyCan uses only standard identifiers.

r0 : Receive bit.

DLC: Data length code.

Data segment: See the table below for specific protocols.

a. Master read data and slave response format (function code 03)

The master reads a 16-bit register. The master address: 0xff. The slave address: 0x1.

Host reads 16-bit register (function code 03)

Arbitration	Control section	Data segment			
can identifier (slave address)	dlc (data segment length)	Host address	function code	Register address	Data length
0x01	0x04	0xff	0x03	0x00	0x01

Slave response (function code 03)							
Arbitration	Control section	Data segment					
can identifier (host address)	dlc (data segment length degree)	From the machine site	Features code	Register ground site	Data length degree	Data low 8 Bit	Data high 8 Bit
0xFF	0x06	0x01	0x03	0x00	0x01	00	00

The master reads two 16-bit data. Master address: 0xff. Slave address: 0x1.

Host reads two 16-bit data (function code 03)					
Arbitration	Control section	Data segment			
can identifier (slave address)	dlc (data segment length)	Host address	function code	Register address	Data length
0x01	0x04	0xff	0x03	0x00	0x02

Slave response									
Arbitration	Control section	Data segment							
can identifier (main Machine address)	dlc (data Segment length)	From the machine site	Features code	register address	Data length degree	Register 1 Lower 8 bits	Register 1 Upper 8 bits	Register 2 Lower 8 bits	Register 2 Upper 8 bits
0xFF	0x08	0x01	0x03	0x00	0x02	00	00	01	00

b. Master write data and slave response format (function code 06)

The master writes a 16-bit register. The master address: 0xff. The slave address: 0x1.

The host writes a 16-bit register (function code 06)		
Arbitration	Control section	Data segment

can identifier (slave address)	dlc (data segment length degree)	Host ground site	Features code	Register ground site	Data length degree	Data low 8 Bit	Data high 8 Bit
0x01	0x06	0xff	0x06	0x00	0x01	01	00

Slave response							
Arbitration	Control section	Data segment					
can identifier (host address)	dlc (data segment length degree)	From the machine site	Features code	Register ground site	Data length degree	Data low 8 Bit	Data high 8 Bit
0xFF	0x06	0x01	0x06	0x00	0x01	01	00

The master writes two 16-bit registers. The master address is 0xff. The slave address is 0x1.

The host writes two 16-bit registers (function code 06)									
Arbitration	Control section	Data segment							
can identifier (from Machine address)	dlc (data segment length)	Host address	Features code	register address	Data length degree	Register 1 Low post	Register 1 High	Register 2 Low post	Register 2 High
0x01	0x08	0xff	0x06	0x00	0x02	01	00	01	00

Slave response									
Arbitration	Control section	Data segment							
can identifier (main Machine address)	dlc (data segment length)	From the machine site	Features code	register address	Data length degree	Data 1 low 8 bit	Data 1 high 8 bit	Data 2 low 8 bit	Data 2 high 8 bit
0xFF	0x08	0x01	0x06	0x00	0x02	01	00	01	00

Note: When the written register address is 0xC (incremental position), the returned data is the current absolute position. When the written register address is 0x16 (absolute position), the returned data is the current absolute position.

c. Host write absolute position buffer, 2ms communication interpolation mode

Master address: 0xff. Slave address: 0x1

The master writes the absolute position cache (the slave does not reply)		
Arbitration	Control section	Data segment

can identifier (slave site)	dlc (data segment length degree)	Absolute position 0 ~ 7 digits	Absolute position 8 ~ 15 people	Absolute position 16 ~ 23	Absolute position 24-31
0x101	0x04	0x00	0x00	0x00	0x00

Note: The lower 8 bits of the identifier are the actual slave address. When the lowest 3 bits of the upper 3 bits of the identifier are 1, it means that this frame is sent as an absolute position buffer, and the motor receives non-reply data, which is used to write the position at high speed. .

Master writes absolute position cache (slave reply)					
Arbitration	Control section	Data segment			
can identifier (slave site)	dlc (data segment length degree)	Absolute position 0 ~ 7 digits	Absolute position 8 ~ 15 people	Absolute position 16 ~ 23	Absolute position 24-31
0x301	0x04	0x00	0x00	0x00	0x00

Slave response					
Arbitration	Control section	Data segment			
can identifier (host ground site)	dlc (data segment length degree)	Absolute position 0 ~ 7 digits	Absolute position 8 ~ 15 people	Absolute position 16 ~ 23	Absolute position 24-31
0xFF	0x04	0x00	0x00	0x00	0x00

Note: The lower 8 bits of the identifier are the actual slave address. When the least significant three bits of the identifier are 1 and the second bit is also 1, it means that the frame is sent as an absolute position buffer, and the motor needs to respond to the data received current position.

d. Emergency stop instruction

Master address: 0xff. Slave address: 0x1

Host write emergency stop			
Arbitration	Control section	Data segment	
can identifier (slave address)	dlc (data segment length)	Host address	function code
0x01	0x02	0xff	0x10

Slave response			
Arbitration	Control section	Data segment	
can identifier (host address)	dlc (data segment length)	Slave address	function

			code
0xFF	0x02	0x01	0x10

Note: After the host writes an emergency stop, the motor stops immediately.

e. Absolute position clear instruction

Master address: 0xff. Slave address: 0x1

The host writes an emergency stop (the slave does not reply)			
Arbitration	Control section	Data segment	
can identifier (slave address)	dlc (data segment length)	Host address	function code
0x01	0x04	0xff	0x11

f. Read multiple register instructions at once

Master address: 0xff. Slave address: 0x1

Host reads three 16-bit registers						
Arbitration	Control section	Data segment				
can identifier (slave site)	dlc (data segment length degree)	Host ground site	Features code	First register address	Second register address	Third register address
0x01	0x05	0xff	0x12	0x00	0x01	0x02

Slave response									
Arbitration	Control section	Data segment							
can identifier (host address)	dlc (data segment length)	Slave address	function code	First register Lower 8 bits	First register Upper 8 bits	Second register low 8 Bit	Second register high 8 Bit	Third register low 8 Bit	Third register high 8 Bit
0xFF	0x08	0x01	0x12	0x01	0x00	0x01	0x00	0x01	0x00

4. EasyCan mode host control process

a: Position mode (contour position mode)

In contour position mode, the driver receives the target position command sent by the master station, and performs trajectory planning according to the speed and acceleration planning parameter settings within the driver. For example: the application requires the servo axis to move to a target position at the set speed and acceleration After receiving the instruction, the driver calculates the position that the servo axis should reach in each NC cycle (for example: 1ms) in the internal trajectory generator, and then sends it to the position loop for execution.

First determine whether the special function is the position mode. (When the special function is 3, it is the speed mode. If it is the speed mode, it must be changed to 0.) If it is the speed mode, it needs to be changed to the position mode. The instructions are as follows:

1.Special Function Send 0

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x06	x| FF 06 19 01 00 00
接收	0x00FF	数据帧	标准帧	0x06	x| 01 06 19 01 00 00

ff: host address
01: Slave address
06: Write function code
19: Register address is 0x19 (special function)
01: data length is 1
00 00: Write data is 0

2.Parameter save and send 1

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x06	x| FF 06 14 01 01 00
接收	0x00FF	数据帧	标准帧	0x06	x| 01 06 14 01 01 00

ff: host address
01: Slave address
06: Write function code
19: Register address is 0x14 (parameter save flag)
01: data length is 1
01 00: Write data is 1

Contour position mode: It is through the acceleration and deceleration curve of the product inside the driver. After the parameters are set, you can use the PLC or the microcontroller, or yourself The designed upper computer software sends the position command. The process of sending the position command is as follows:

First set the required target speed and acceleration. The target speed address is 0x2 (see the register table) and the acceleration address is 0x3.

For example, a speed of 1000 rpm (0x03E8 in hexadecimal) and an acceleration of 10000RPM / S (0x2710 in

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 02 02 E8 03 10 27
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 02 02 E8 03 10 27

hexadecimal) are required.
ff: host address
01: Slave address
06: Write function code
02: Register address is 0x2 (target speed)
02: Data length is 2
E8 03 (lowest bit first): data written to 0x2 address register is 0x03E8 (1000 decimal)
10 27 (lowest bit first): data written to 0x3 address register is 0x2710 (decimal 10000)

Send incremental position (the meaning of incremental position is that the data sent is the position where the motor needs to go forward or backward). For example, it needs to go forward one round (the motor encoder is a 15-bit absolute encoder, one turn The number of pulses is 32768). The incremental position register address is 0x0C. The instruction is as follows:

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 0C 02 00 80 00 00
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 0C 02 0E 37 00 00

ff: host address
01: Slave address
06: Write function code
0C: Register address is 0xC (incremental position)
02: Data length is 2
00 80 00 00 (lower bit first): Data written to 0xC and 0xD address registers is 0x0000 8000 (32768 decimal)
0E 37 00 00 (lowest bit first): The current position data returned is 0x0000 370E

For example, you need to make a forward and backward turn (the motor encoder is a 15-bit absolute value encoder, and the number of pulses per turn is 32768). The binary calculation method of inversion is -32768 is as follows: The binary of 32768 is 00 00 80 00 (the high order is first)). (Note: 0 = ff ff ff ff +1)

-32768 is 0-00 00 80 00 = ff ff ff ff-00 00 80 00 + 1 = ff ff 7f ff +1 = ff ff 80 00

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 0C 02 00 80 FF FF
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 0C 02 0E 37 00 00

ff: host address
01: Slave address
06: Write function code
0C: Register address is 0xC (incremental position)
02: Data length is 2
00 80 FF FF (lower bit first): Data written to 0xC and 0xD address registers is 0xFFFF 8000 (-32768 decimal)
0E 37 00 00 (lowest bit first): The current position data returned is 0x0000 370E

Send the absolute position (the meaning of the absolute position is that the absolute position is cleared to 0 or the position is defined as 0 when the origin is automatically found. The absolute position is to go to the new position, such as the first time to send 32768 to the 360 ° position of the motor. The second time it has been sent to the position of 32768.
If you go, send another 16384, it will reverse half a circle. If you send an absolute position, you need to go to that position. Unlike the incremental position, the incremental position is based on the current position to go forward or backward.) Absolute position address 0x16.

For example, the motor needs to go to the position of 2 turns (the number of pulses of a motor is 32768, and the number of turns is 65536)

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 16 02 00 00 01 00
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 16 02 0E 37 00 00

ff: host address
01: Slave address
06: Write function code
0C: Register address is 0x16 (absolute position)
02: Data length is 2
00 00 01 00 (lower bit first): The data written to the 0xC and 0xD address registers is 0x0010 0000 (32768 decimal)
0E 37 00 00 (lowest bit first): The current position data returned is 0x0000 370E

For example, the motor needs to go back to the origin (when the electronic gear numerator is 0, sending 0 is to clear the current position, so going back to the origin and sending 1, at this time, a pulse does not affect the accuracy)

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 16 02 01 00 00 00
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 16 02 11 37 00 00

Note: The control motor only needs to send the required position first (as far as possible, use the absolute position command, because it can be sent multiple times and still go to the same position), and then return to the absolute position to compare whether it has reached the set position (note that it is necessary to judge when (Allow + -10 error) to determine whether to execute the next instruction. Or you can connect the pf signal and after it reaches the position, the driver will give a switching signal output by the optocoupler.

b: Position mode (contour position mode) control process

The contour position mode only needs to set the required acceleration and speed, and then send the absolute position or incremental position to be reached.

（1）The target speed is set to 1000 and the acceleration is set to 10000. (For detailed

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 02 02 E8 03 10 27
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 02 02 E8 03 10 27

instructions, see the detailed explanation)

（2）Absolute position send 32768

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 16 02 00 00 01 00
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 16 02 0E 37 00 00

（3）Absolute position send 1

传输方向	ID号	帧类型	帧格式	长度	数据
发送	0x0001	数据帧	标准帧	0x08	x| FF 06 16 02 00 00 01 00
接收	0x00FF	数据帧	标准帧	0x08	x| 01 06 16 02 0E 37 00 00

（4）Repeatedly send the absolute position that needs to be reached. To determine whether it is in place, you can judge whether it has reached by reading the absolute position and the absolute position sent. (Note that the motor will allow a certain number of pulses depending on the position kp. error).

c: Position mode (periodic synchronous position mode) multiple motors

synchronous control process (motors accumulate deceleration pulses)

Cyclic synchronous position mode is different from Profile Position Mode in that its trajectory generator is located on the controller side, not in the driver. In this mode, the controller only needs to issue periodically. The target position is sufficient (similar to the principle of position interpolation mode).

Period synchronous position mode, the motor acceleration is set to 60000 in advance, which can be modified and saved by the provided upper computer software. If it is 3 The motors are set to different addresses in advance (for example, address 1, address 2, and address 3 respectively). The target speed of the cycle synchronous position mode is limited Maximum speed, so set the maximum speed of the motor in advance (e.g. 1500rpm).

The synchronization period is 2ms. Now suppose we need the motor to run at 600rpm. Then we calculate the number of pulses that the motor needs to pass in 2ms. The calculation method is as follows:

(600R/MIN) / 60 = 10R/S。

（10R/S） / 500 = 0.02R/(2ms) (Note: 1 second divided by 500 = 2ms)

0.02 / (2ms) * 32768 = 655.36 P / (2ms) (Note: the number of pulses per motor revolution is 32768)

Therefore, an incremental motor that sends 655 increments to the absolute position of the motor every 2ms runs at 600RPM. The acceleration of the start and the deceleration curve of the stop are also generated by the controller. Calculate the position that needs to be increased every 2ms, and then write to the motor separately. Write sync signal.

Preliminary preparation process:

（1） Via the provided upper computer software, modbus enable sending 1 (enable communication)

（2） The electronic gear molecule sends 0 (after the electronic gear molecule is saved as 0, the next time the power is turned on again, the modbus enable defaults to 1).

（3） The motor acceleration sends 20000 (the motor acceleration cannot be 60000, 60000 is no hysteresis mode).

（4） Parameter saving standard Send 1 (the parameters will take effect next time after saving after saving)

（5） Power on again. Actual control process:

（1） Write 1 to the CAN communication synchronization control word (register address 0x1c) of 3 motors (enable CAN synchronization control).

（2） Read the absolute positions of the three motors (register address 0x16 0x17).

（3） Write the absolute position buffers of 3 motors respectively (refer to the host write absolute position buffer, 2ms communication interpolation mode format) write 2ms The number of pulses to be walked (note that the current absolute position must be read for the first time, and then the incremental pulses that need to be walked are added to the current absolute position before sending.

（4） Repeat step (3)

d: Position mode (periodic synchronous position mode) multiple motors synchronous control process (motors without hysteresis)

This mode is different from the previous cycle synchronous position mode, the difference is that the

motor does not accumulate deceleration pulses, and follows in real time according to the pulse amount given by the master. This mode requires the master to give a 10Khz 50% duty cycle synchronization signal input The motor PU port is used to synchronize the clock between the main control and the motor, so this 10KHZ is best generated by the customizer of the main control. CAN communication is still given by each 2MS where the 2ms motor needs to go (the corresponding pulse number). The specific calculation method is the same as the previous one.

Motor parameter preparation:

（1） Via the provided upper computer software, modbus enable sending 1 (enable communication)

（2） The motor acceleration sends 60000 (motor acceleration 60000 is no hysteresis mode).

（3） Special function sends 9 (enable pu pin to input 10k synchronization signal)

（5） Parameter saving standard Send 1 (the parameters will take effect next time after saving after saving)

（6） Power on again. Actual control process:

（1） Write 1 to the CAN communication synchronization control word (register address 0x1c) of 3 motors (enable CAN synchronization control).

（2） Read the absolute positions of the three motors (register address 0x16 0x17).

（3） Write the absolute position buffers of 3 motors respectively (refer to the host write absolute position buffer, 2ms communication interpolation mode format) write 2ms The number of pulses to be walked (note that the current absolute position must be read for the first time, and then the incremental pulses that need to be walked are added to the current absolute position before sending.

（4） Repeat step (3)

5. Communication error code

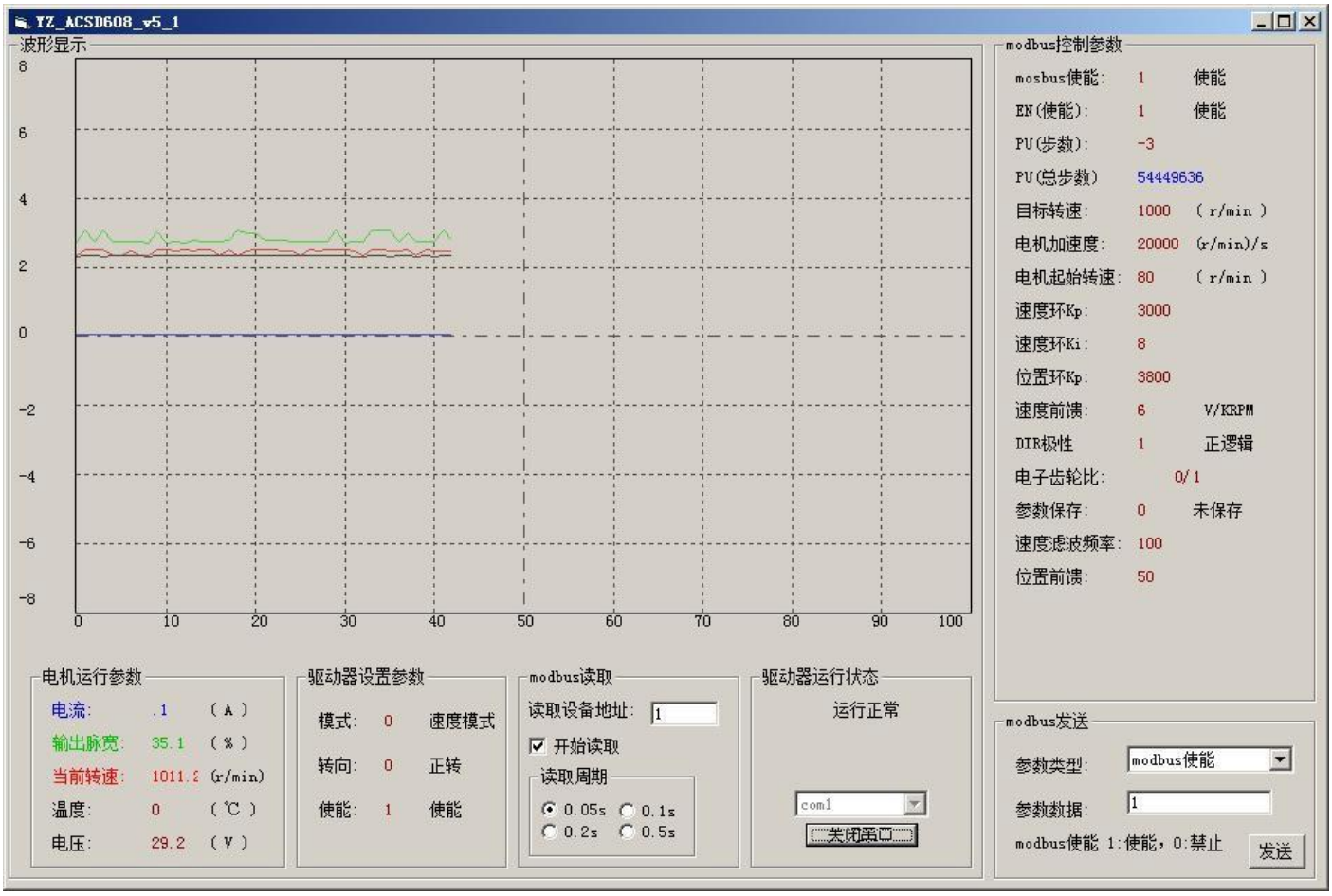
Slave response (error code)				
Arbitration	Control section	Data segment		
can identifier (host address)	dlc (data segment length)	Slave address	function code	error code
0xFF	0x03	0x01	0x90	0x00

error code	Reason for alarm	Alarm processing
0x02	Illegal address	Read or write address exceeds maximum address
0x03	Illegal Function Code	Function codes are only 0x3 and 0x6
0x04	Illegal data length	Only two 16-bit numbers can be written at a time
0x05	2ms communication interpolation frequency is too high	2ms send an interpolation absolute position

六. PC software instructions

The host computer software provided by the driver is realized through modbus (serial port TTL level communication), which can detect the state of the motor and modify the motor parameters

through the software. When debugging CAN, the communication software can also be used at the same time.



As shown in the figure above, the software is divided into several parts such as waveform display and motor running parameters. The functions and functions of each part are introduced below. Waveform display: There are 4 channels in total, which are represented by 4 colors. Color and motor operating parameters Font color inside,
The colors are the same. That is: blue indicates the current, green indicates the output pulse width, red indicates the current speed, and black indicates the voltage. Motor operating parameters: Real-time data of motor operation.

Driver setting parameters: display the DIP switch of the driver, and the direction enable setting. If it is in modbus mode, this
The column is invalid.

Drive running status: This column shows the alarm status of the drive. If there is no alarm, it shows normal operation.

Modbus control parameters: The parameters in this column are the internal parameters of the drive. If you want to modify these parameters, you must first enable the modbus. 1. For the specific parameter meaning, please refer to the register description.

Modbus read: This column can set the address of the drive, the period of reading the drive data, and whether to read.

Modbus Send: This field is used to modify the drive parameters. First select the parameter type, then set the parameter data, and then click Send.