1.Parameters of servo motor

Stroke: 500mm

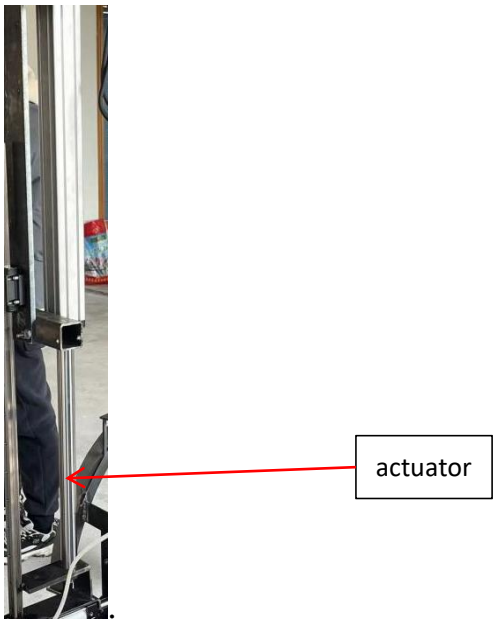
Speed: 3000 laps/min

Load: 700N

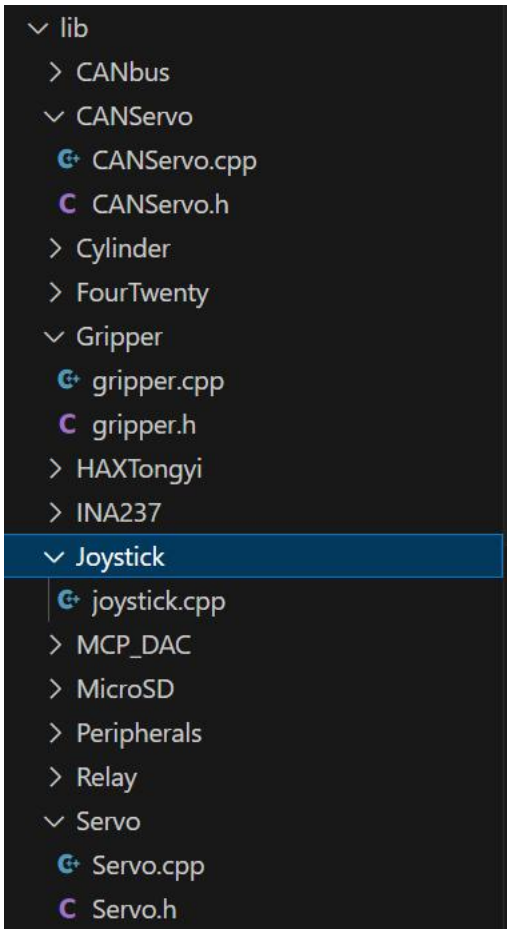
Lap: 50; When the motor runs for 1 laps, the electric actuator will extend by 10mm

Communication frequency: 250k ; Because the communication frequency of the joystick is 250k, and the joystick and the motor use the same bus, the frequencies of the two must be kept consistent.

In order to change the communication frequency, prepare a RS485 to USB module in advance.



2.Description of lib



CANServo.h: It was written by Han Long, but I didn't use it.

Gripper.h: I wrote the code from line 89 to the end. class Gripper is used for the old machine.

HAXTongyi: It is an example provided by the manufacturer.

Servo: function of controlling servo motor

Joystick.cpp: controlCylinders() is used for testing

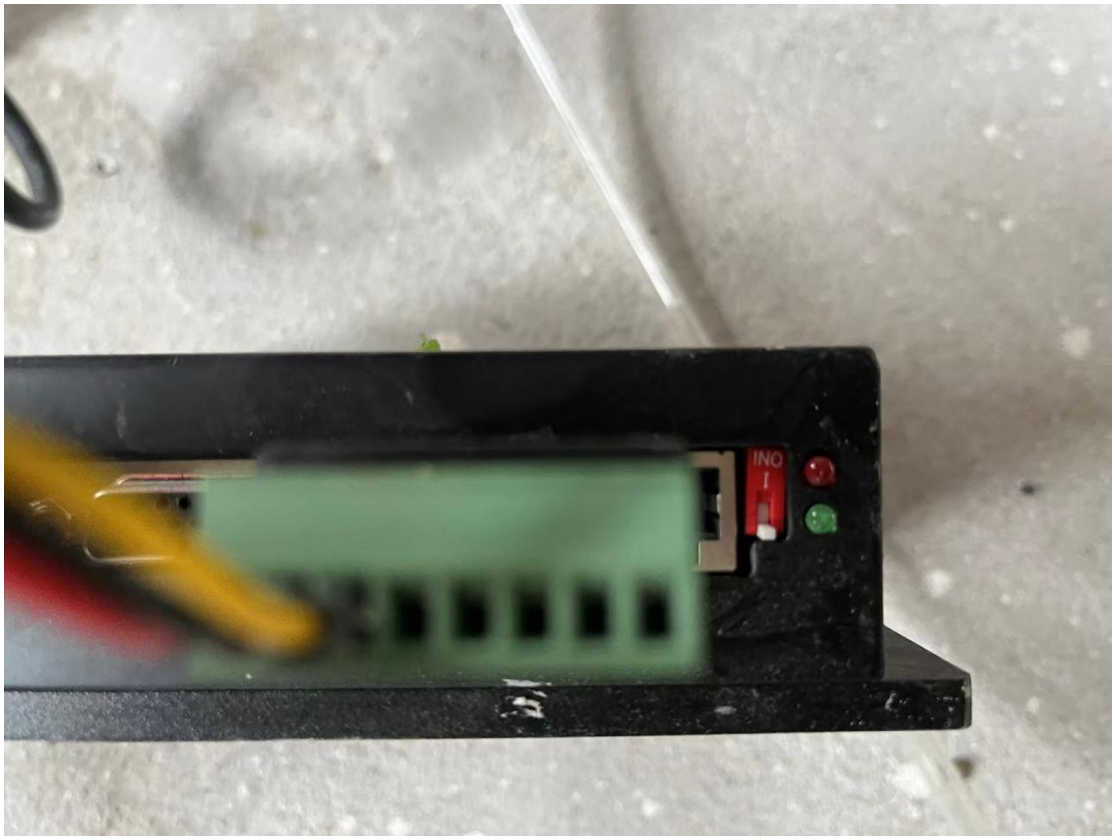
3.Process of controlling the motor

Step1: Turn on the motor power and then turn on the EE box power.

Note: The controller of the motor can only receive the NMT command from the PCB after it is started, and can only start CAN communication after successfully receiving the command.

Step2: Wait

Note: use the current code, after about twenty seconds, the motor will be enabled, and after another ten seconds, you can control the motor using the joystick.

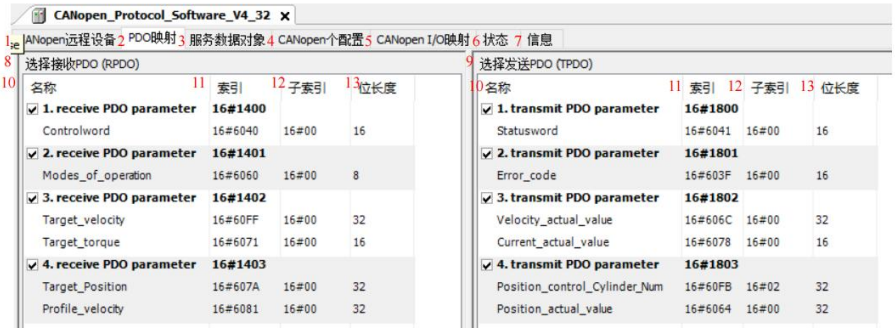


The red LED is an alarm light. When it flashes, it indicates that the motor or controller is running in error. The current solution is to power off and restart.
The green LED is the controller operation indicator. When it is always on, it indicates that the motor is already in the enabled state.

Step3: control

6.8.6 Default PDO Mapping Parameter in Location Mode

The default configuration PDO mapping parameter of the driver is shown in the following table:

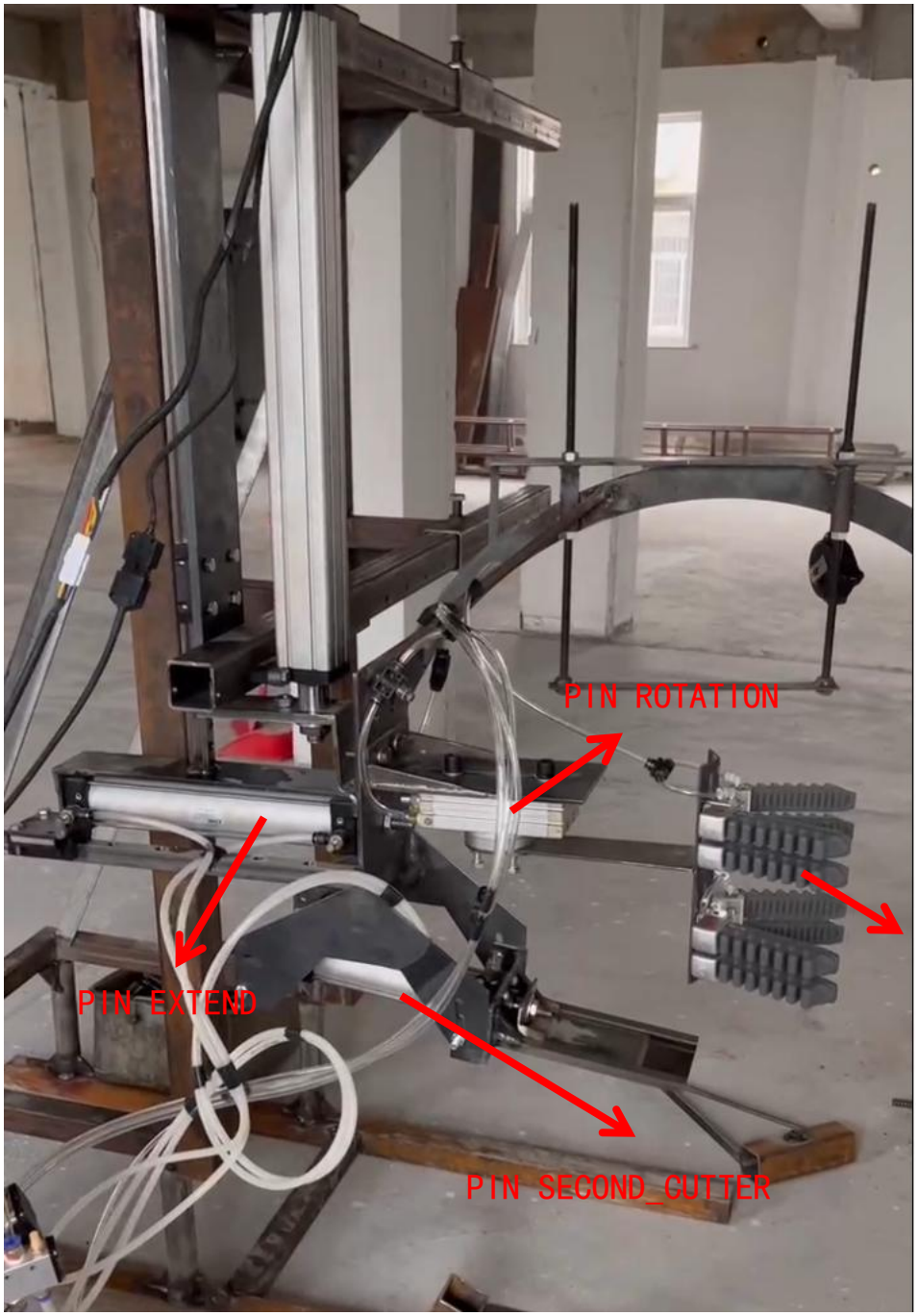


Use PDO; position Mode

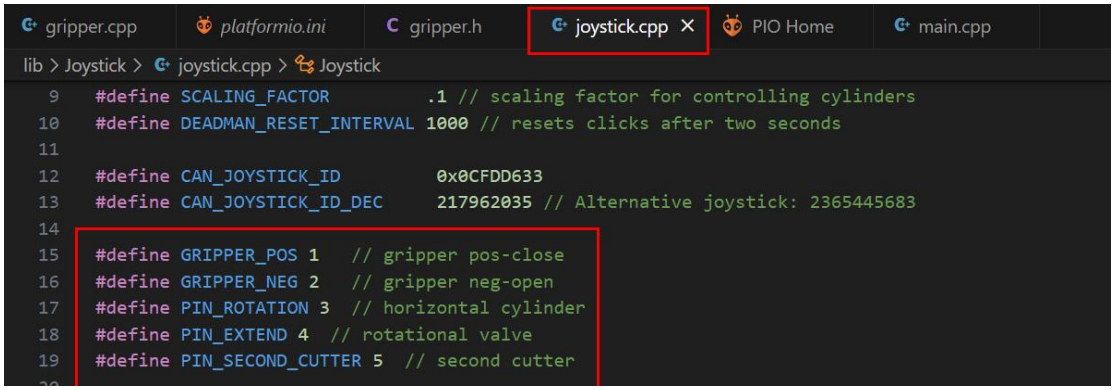
Step4: After operating the actuator to 0 position(Stroke: 500mm), turn off the power

Note: The motor currently used is a relative value motor.

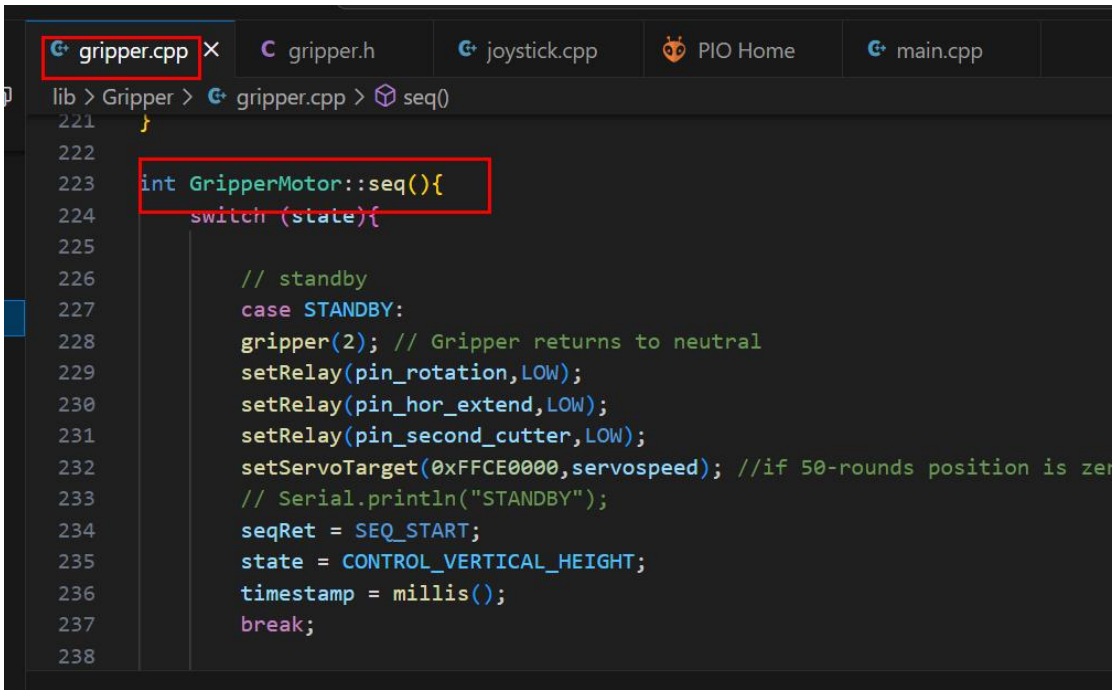
4. code


PIN ROTATION
PIN EXTEND
PIN SECOND_CUTTER
PIN GRIPPER_POS &
PIN GRIPPER_NEG

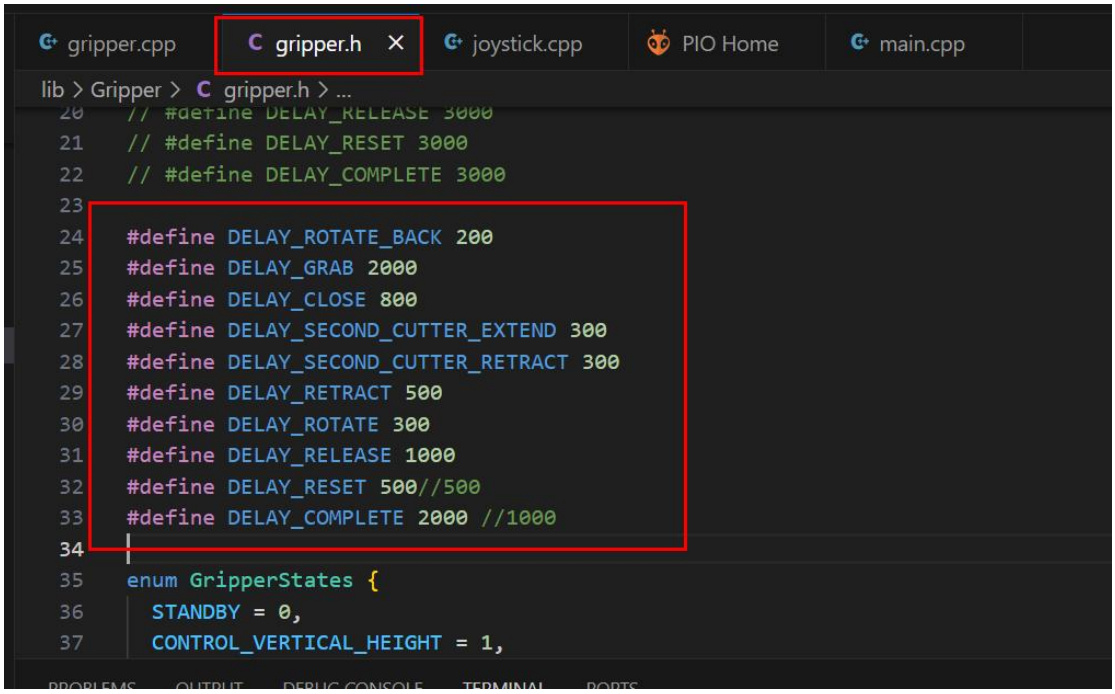
4.1 definition of pin



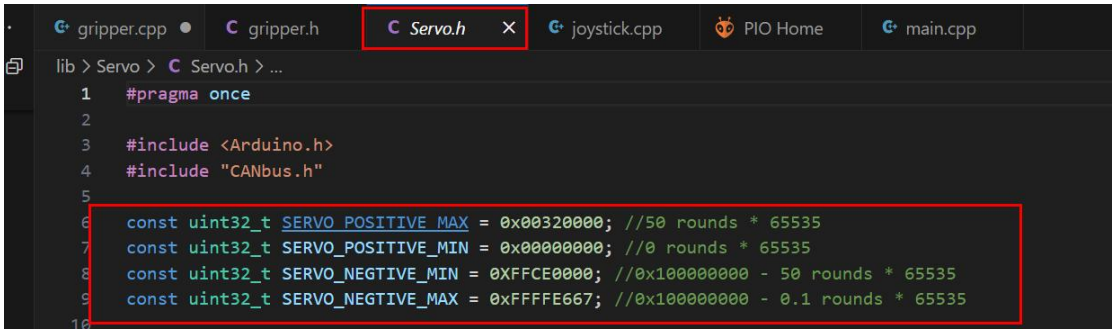
4.2 Sequence



4.3 Time control of sequence

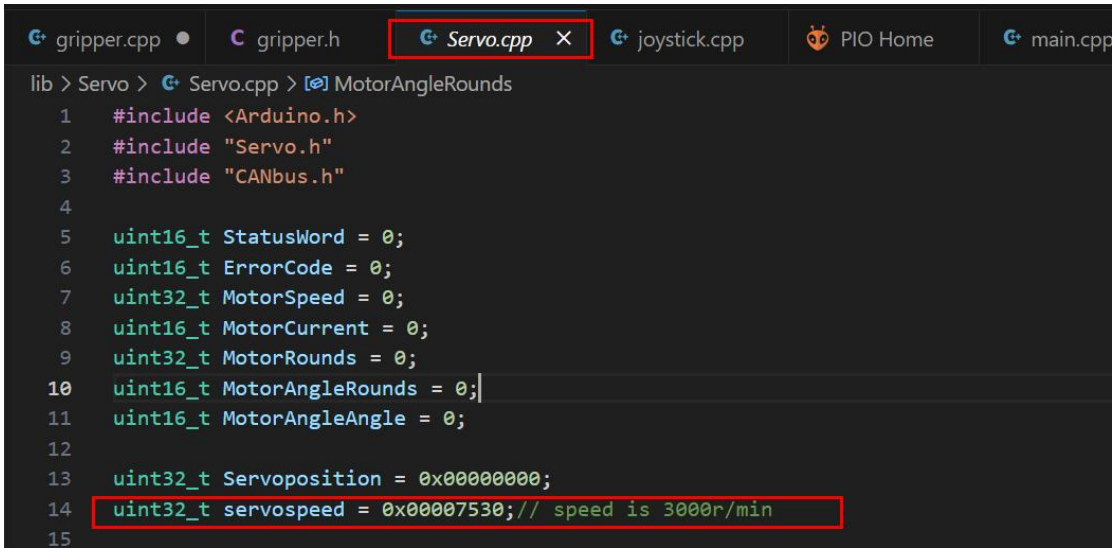


4.4 Set the limit position of the motor

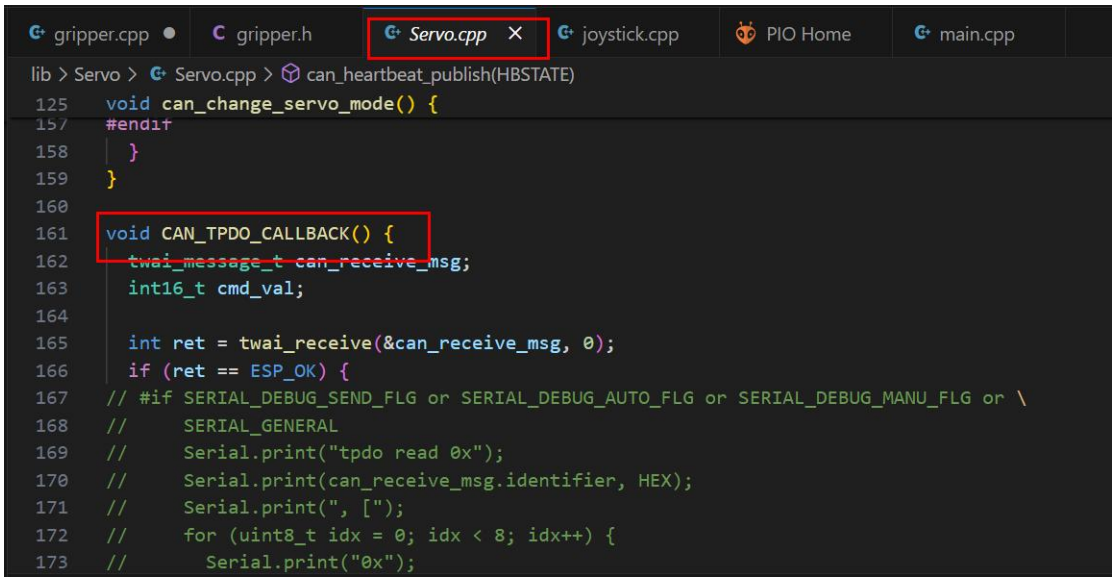


4.5 Set the speed of the motor

Generally set the motor to run at the fastest speed.



4.6 Feedback value of motor



	CAN 标识符	字节数	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7	字节 8
TPDO1	0x180+Node-ID	2	00	00						
			CANOPEN 控制状态反馈，SDO 地址：0x6041							
			禁止时间：20（2ms）;事件时间：50ms							
TPDO2	0x280+Node-ID	2	00	00						
			错误代码反馈，SDO 地址：0x603F							
			禁止时间：20（2ms）;事件时间：50ms							
TPDO3	0x380+Node-ID	6	00	00	00	00	00	00		
			电机速度反馈, 0x606C				电机电流反馈, 0x6078			
			禁止时间：200（20ms）;事件时间：50ms							
TPDO4	0x480+Node-ID	8	00	00	00	00	00	00	00	00
			电机圈数反馈，0x60FB 02				电机实时角度反馈，0x6064 （低 16 位角度+高 16 位圈数）			
			禁止时间：200（20ms）;事件时间：50ms							

5. Set the zero position

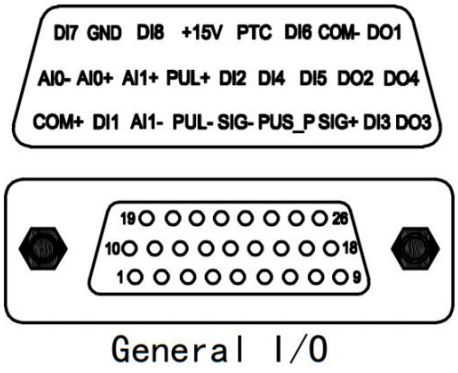
5.1 Currently, the position is set as the zero position when the stroke is 50cm. As shown in the picture.

There are two reasons for this setting. 1)The load of the motor is very heavy (about 20 kilograms), so when the power of the motor is turned off, the actuator will automatically slide to the bottom. 2)When the stroke is the maximum value, it is easier to calculate the command value for controlling the motor. For example, setting the stroke to 20cm means that the motor is retracted by 30 laps.

Note:The newly purchased backup motor is different from the one currently in use. The manufacturer said that the new motor has a "brake", which prevents the actuator from sliding to the bottom when the motor is powered off.

6. Set the zero return mode

Connection:
P1 and P22 are short-circuited
P15 is connected to the brown wire
P20 is connected to the blue wire
Select zero return mode 1



General I/O

01	IO_COM+	IO input power supply	Input Common Terminal
15	USRC_DI4	Digital input	NOT input
20	GND_DIG	Digital ground	
22	+15V_DIG	Power supply output	< 100mA

The zero return mode can only be enabled through the SDO communication mode