STM32 car autopilot

1.Experiment preparation

1. knowledge reserve

- Have good programming ability (mainly C language)
- Familiar with the architecture of stm32

2. Material preparation

- Smart car mini chassis *1
- stm32F103RCT6 *1
- Yahboom's dual motor driver board *2 (Other motor driver boards may not be suitable for the source code provided in this tutorial, you need to transplant it yourself)
- Four-way tracking module *1
- 310 motor*4
- k210 viewing angle module *1
- K210 angle of view module heightened bracket *1
- 7.4V battery *1
- Several DuPont lines
- M3 copper pillars, several M3 screws

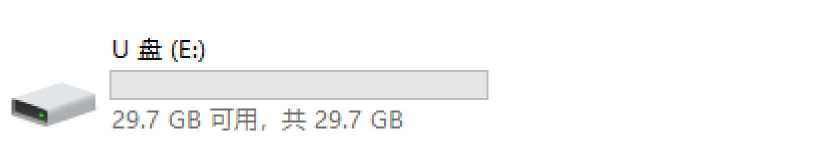
3. k210 viewing angle module preparation

   3.1

   Take the TF card out of the k210 vision module and insert it into the card reader.

3.2

Insert the card reader into the computer and wait for the computer to recognize the U disk

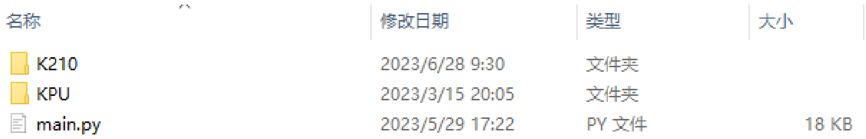


U 盘 (E:)

29.7 GB 可用，共 29.7 GB

3.3

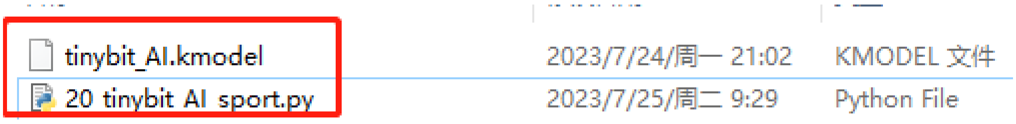
After the computer recognizes it, enter it into the TF card.



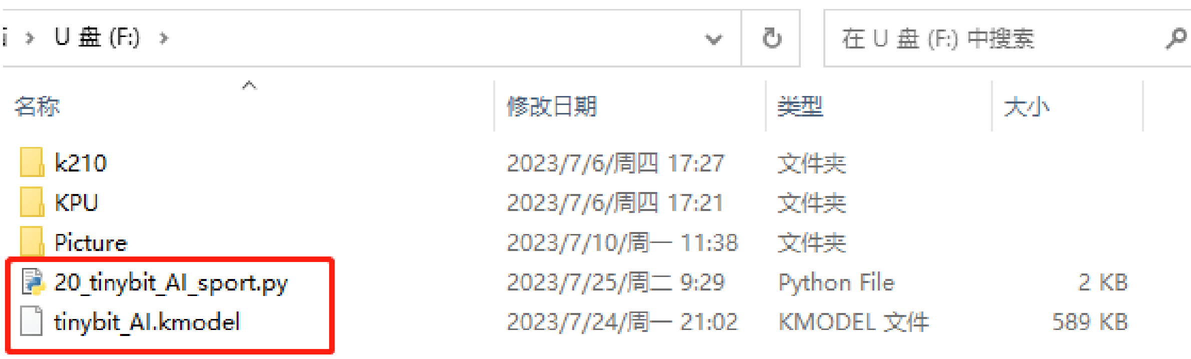
名称	修改日期	类型	大小
K210	2023/6/28 9:30	文件夹	
KPU	2023/3/15 20:05	文件夹	
main.py	2023/5/29 17:22	PY 文件	18 KB

3.4

Find the k210 folder in the source code of the data program, find the **20_tinybit_AI_sport.py** and **tinybit_AI.kmodel** files from the folder, and copy them to the root directory.



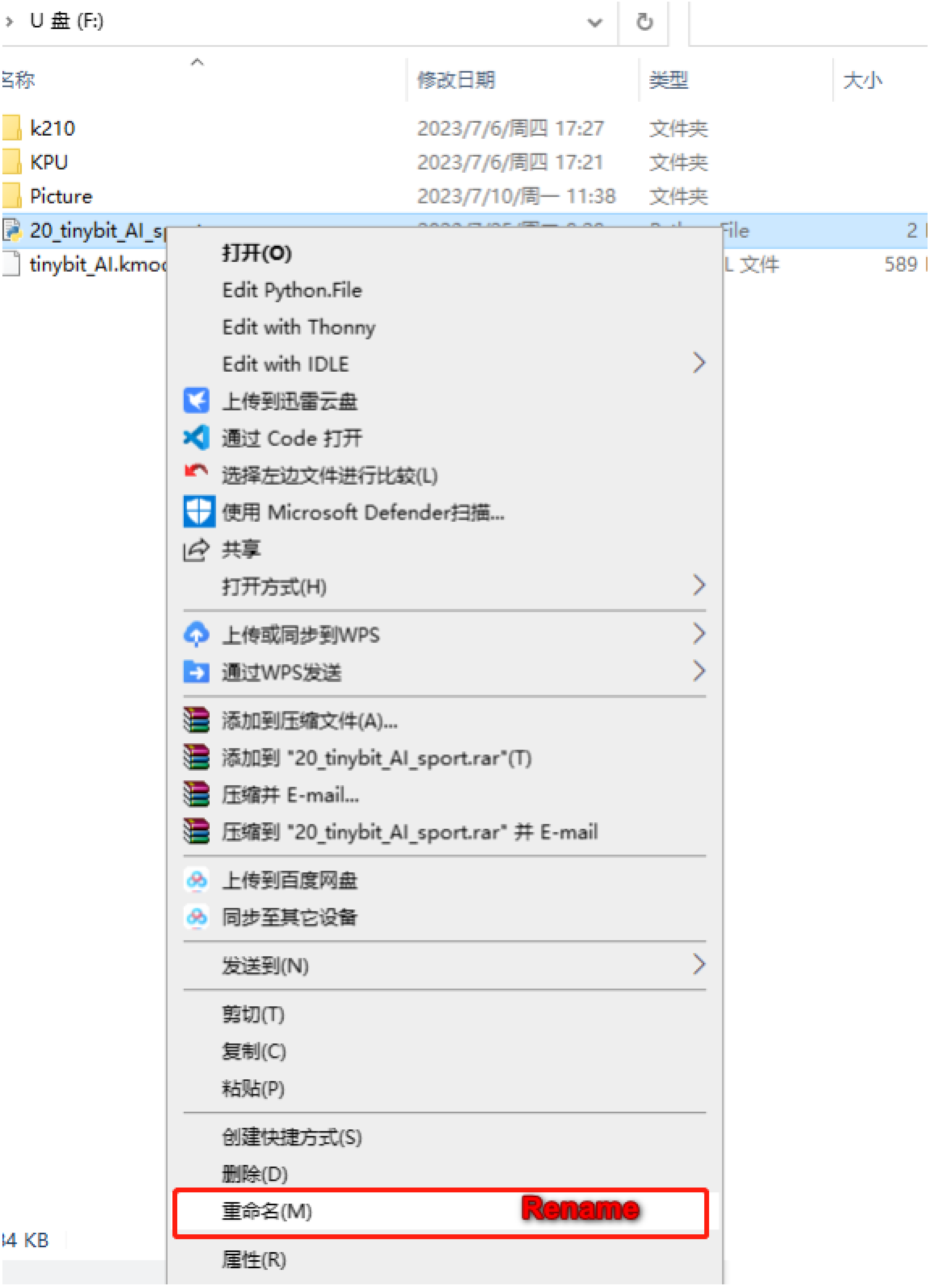
tinybit_AI.kmodel	2023/7/24/周一 21:02	KMODEL 文件
20_tinybit_AI_sport.py	2023/7/25/周二 9:29	Python File

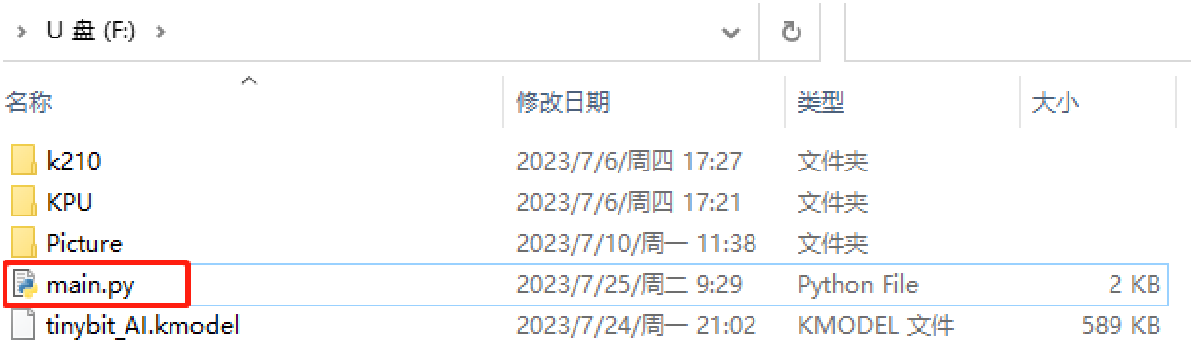


名称	修改日期	类型	大小
k210	2023/7/6/周四 17:27	文件夹	
KPU	2023/7/6/周四 17:21	文件夹	
Picture	2023/7/10/周一 11:38	文件夹	
20_tinybit_AI_sport.py	2023/7/25/周二 9:29	Python File	2 KB
tinybit_AI.kmodel	2023/7/24/周一 21:02	KMODEL 文件	589 KB

3.5

Delete the main file and rename the 20_tinybit_AI_sport.py file to the main file.





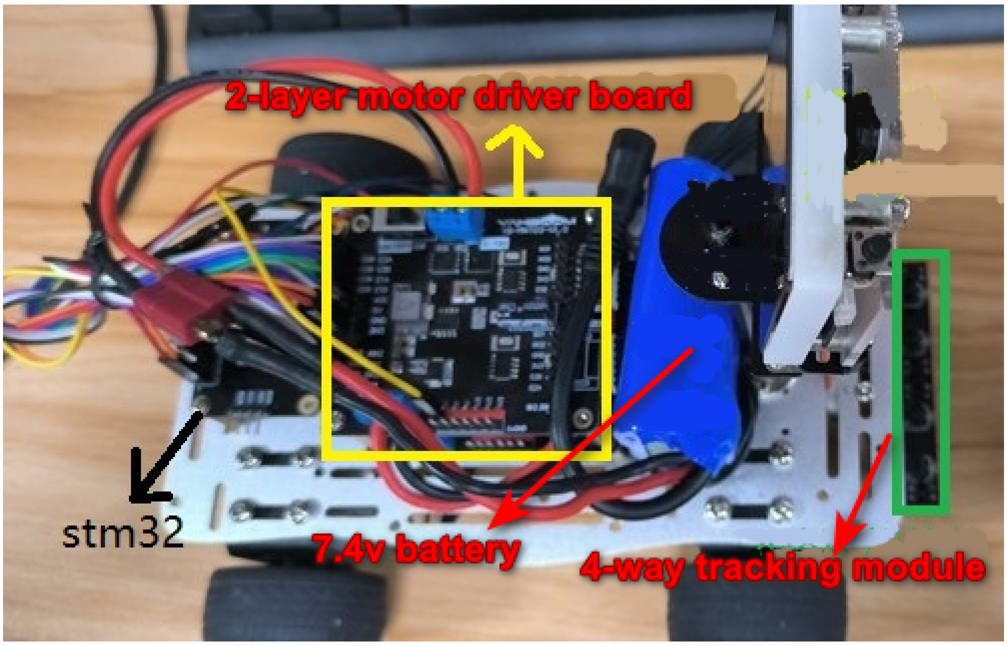
› U 盘 (F:) ›

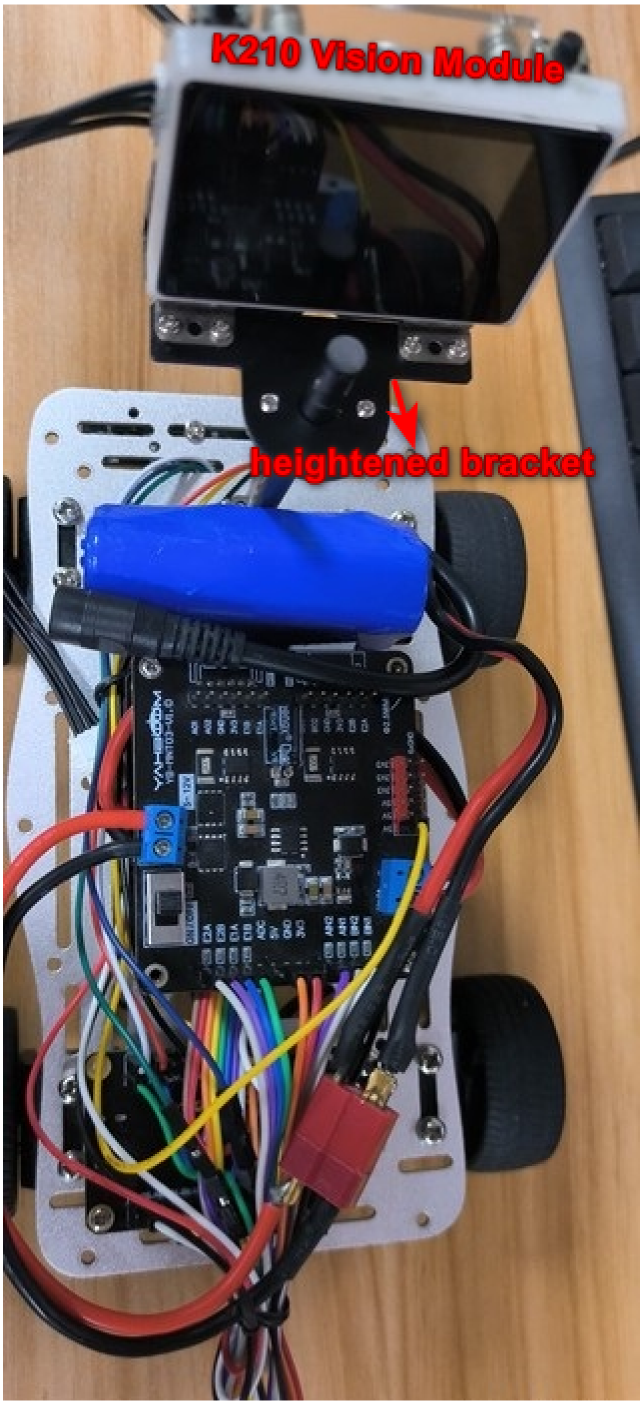
名称	修改日期	类型	大小
k210	2023/7/6/周四 17:27	文件夹	
KPU	2023/7/6/周四 17:21	文件夹	
Picture	2023/7/10/周一 11:38	文件夹	
main.py	2023/7/25/周二 9:29	Python File	2 KB
tinybit_AI.kmodel	2023/7/24/周一 21:02	KMODEL 文件	589 KB

3.6
After renaming, pull out the card reader, take out the TF card, and insert it back into the k210 vision module.

2.Car wiring

After the car is assembled, as shown in the figure below





2.1 Stm32 and dual driver board wiring part

1. Wiring of stm32F103RCT6 and dual motor board (top board)

STM32RCT6	The top 2-channel motor board
PA11	BN1
PA8	BN2
PC6	AN1
PC7	AN2
3.3	3V3
GND	GND
PA0	E2A
PA1	E2B
PA15	E1A
PB3	E1B

2. Wiring of stm32F103RCT6 and dual motor board (lowest board)

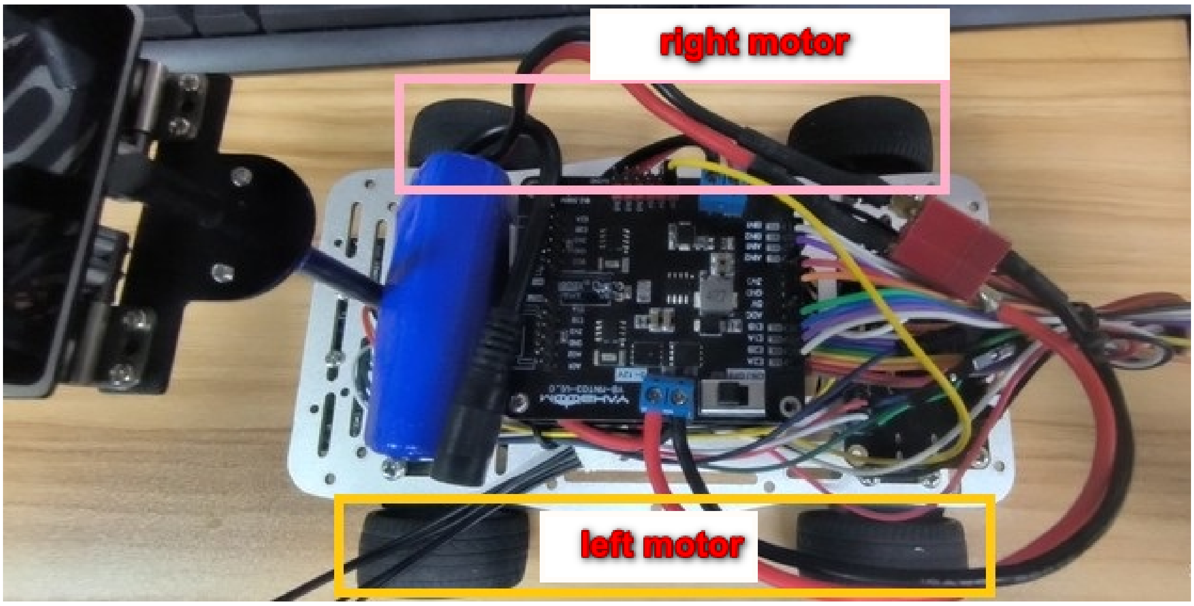
STM32RCT6	The bottom 2-channel motor board
PB0	BN1
PB1	BN2
PC8	AN1
PC9	AN2
3.3	3V3
GND	GND
PA7	E2A
PA6	E2B
PB7	E1A
PB6	E1B

3. The motor drive board on the top layer is the motor of the two wheels (that is, the front motor) that is connected to the electromagnetic sensor，

   motorA--->right motor、motorB--->left motor

4. The motor drive board at the bottom is connected to the two-wheeled motors away from the electromagnetic sensor (that is, the motor at the back)，

motorA--->right motor、motorB--->left motor



2.2 Wiring of STM32F103RCT6 and infrared sensor

STM32RCT6	4-way line inspection sensor
PA5	S1
PA4	S2
PA3	S3
PA2	S4
3.3V	VCC
GND	GND

2.3 Wiring of STM32RCT6 and k210

STM32RCT6	k210
PA9	RXD
PA10	TXD
VCC	VCC
GND	GND

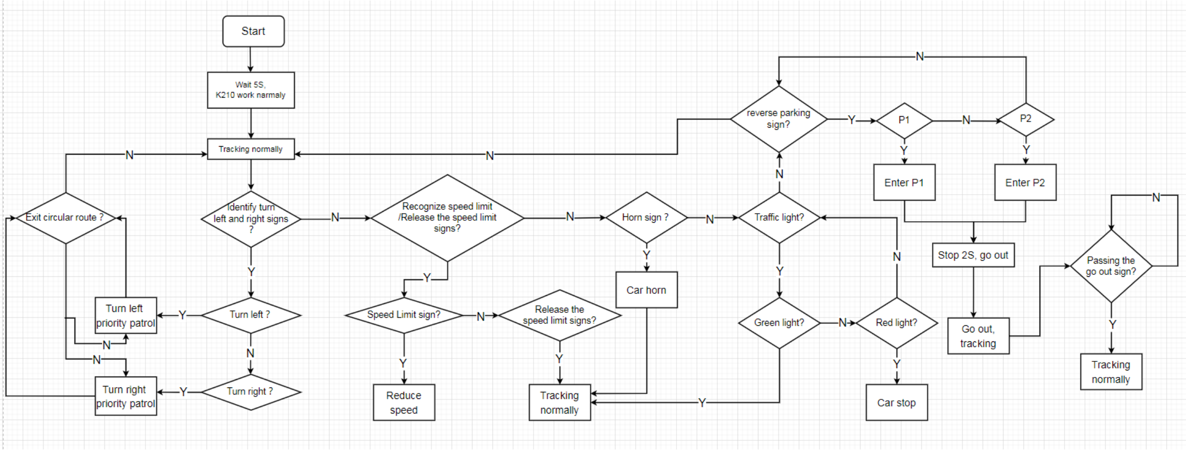
3.Main source code analysis

```
int main(void)
{
    //A series of initialization work...
    while(1)
    {
        switch(id_num_back)
        {
            case red_light:
                Motion_Set_Pwm(0,0,0,0);//car stop
                set_dataid(MAX_id);//back to normal patrol
            break;

            case green_light:
                set_dataid(MAX_id);//back to normal patrol
            break;

            case school:
            case walk:
                set_dataid(MAX_id);//back to normal patrol
                break;

            case right:
                Road_sign_right();//Right turn priority line patrol
            break;

            case left:
                Road_sign_left();//Left turn priority line patrol
                break;

            case freeSpeed:
                set_dataid(MAX_id);//back to normal patrol
                break;

            case limitSpeed:
                Road_sign_speedlimit();//low speed line patrol
            break;

            case chuku_track_line://Line patrol after leaving the warehouse
                    car_outbound_track();
                    break;

            case one:
            case two:
            case horn:
                break;

            default: Track_line();//Call the normal line following function
        }
    }
    //...
}
```

- After booting, stm32 waits for 7 seconds to wait for the k210 module to enter the working state normally
- Track_line: This function is a normal line patrol movement. If no road signs are recognized, this function is usually called for movement.
- Reverse_parking_no1: This is the process of driving the car back to No. 1 parking garage when k210 recognizes the number 1
- Car_outbound : This is to successfully drive into the No. 1 parking garage, stop for 2 seconds, then drive out of the garage, and then perform line inspection after exiting the garage

- Road_sign_left: After k210 recognizes the road sign for turning left, it will perform the processing of line inspection for turning left
- Reverse_parking_no2: This is when the k210 recognizes the number 2, the process of driving the car back into the No. 2 garage
- Car_outbound_track: This is the processing of line inspection after leaving the garage, and it will return to normal line inspection after a period of time.
- Road_sign_right: After the k210 recognizes the right-turn road sign, it will perform the processing of right-turn priority line inspection
- Road_sign_speedlimit: Recognize the speed limit sign, drive the car into the speed limit movement
- set_dataid: This function is to directly set the state of the car.

The flow chart of the program is roughly as follows



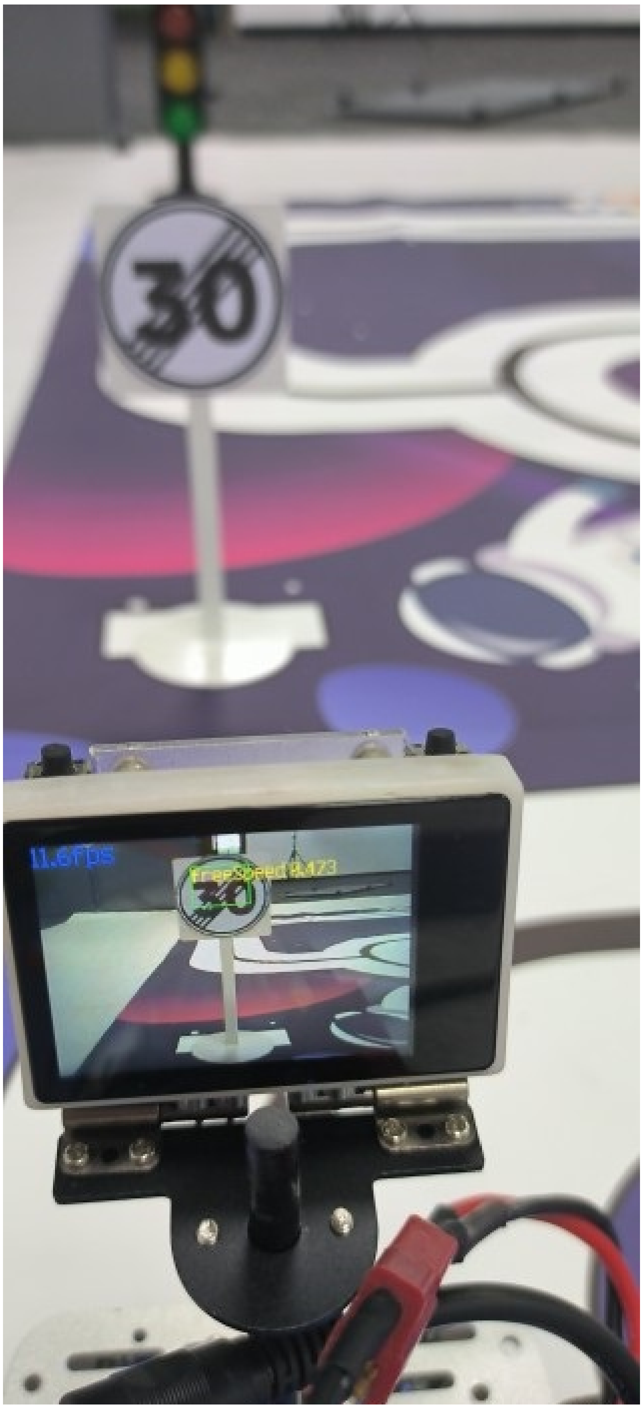
The source code provided in this tutorial retains the RGB\LCD\IMU\external Flash driver on the STM32F103RCT6 board, which can be called directly to facilitate secondary development

4.Experimental results

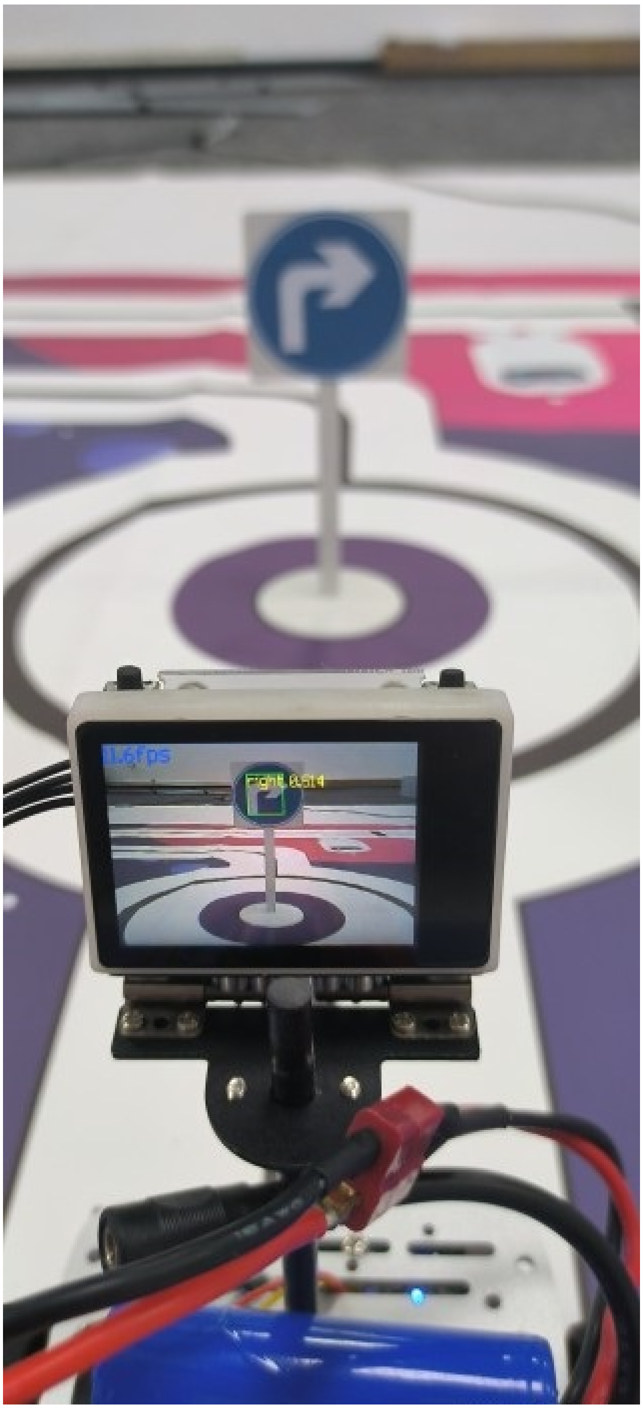
After everything is normal, the car will start along the map and plan the path of the tracking movement. When the road sign is recognized, the corresponding path planning movement will appear.

1. When the speed limit sign is recognized, the car will do a speed patrol movement that does not exceed the speed limit sign

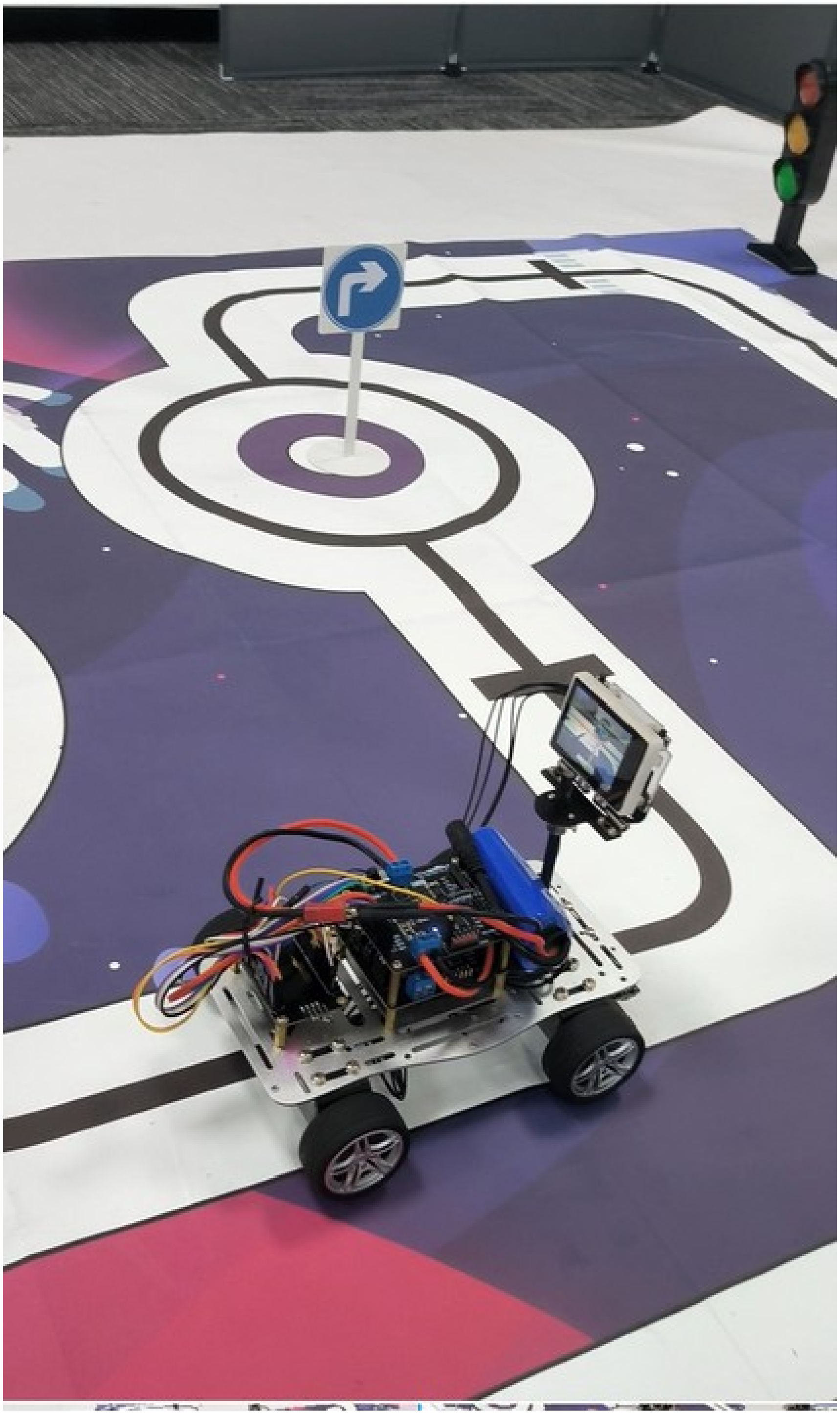
2. When the speed limit sign is recognized, the car will return to the normal line patrol movement.



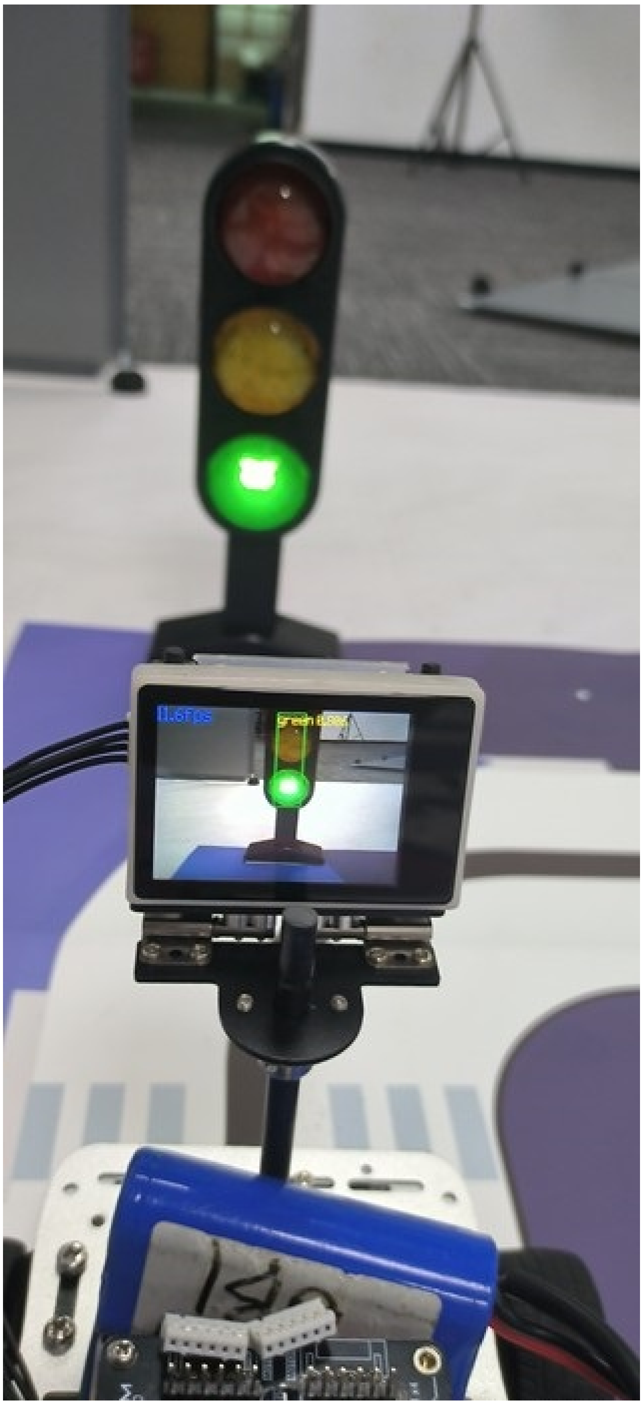
3. When a left/right turn is recognized, the car will make a priority left/right turn patrol movement


11.6fps
right 0.514

4. The horn is recognized, because this car does not have a buzzer, so it is normal to patrol the line.



5. Recognize traffic lights, red lights will stop, green lights will continue to move forward.

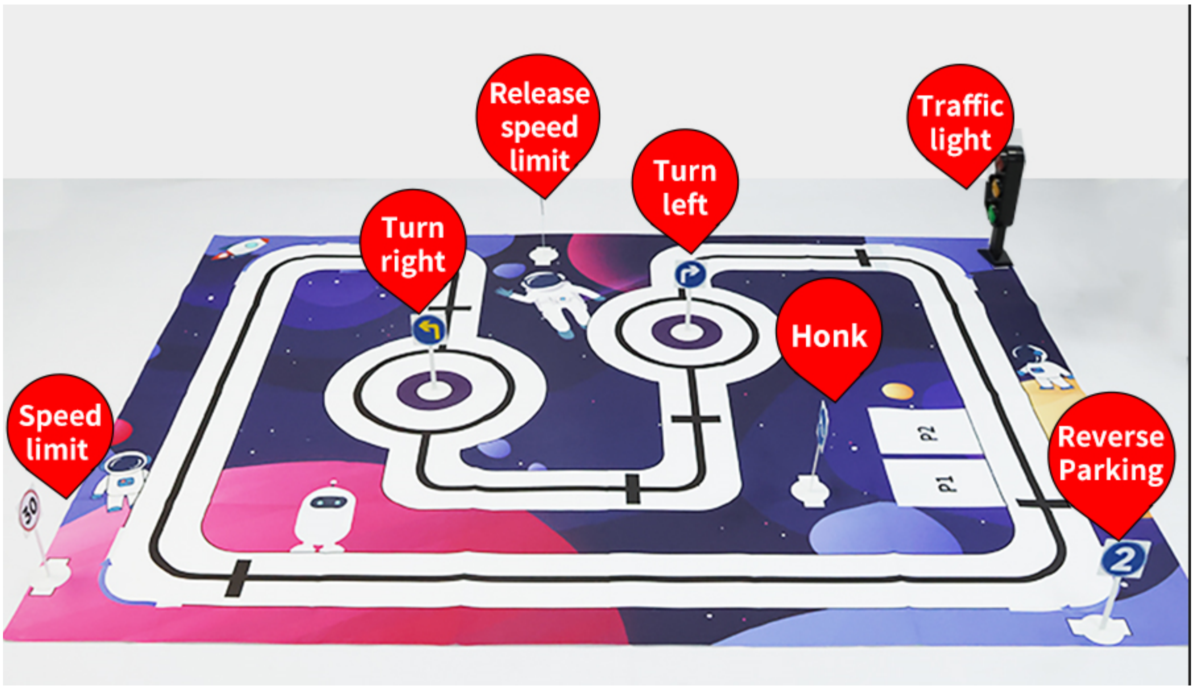


6. Identify No. 1 and No. 2 garages, the car will enter No. 1 garage if it recognizes No. 1 garage; recognize No. 2 garage, and the car will reverse into No. 2 garage; 2 seconds after entering the garage, the car will come out and continue to patrol the line.

1
one 0.658

5.Notes about this routine

1. The signposts placed on this tutorial map are shown in the figure below：



2. The battery must be fully charged, otherwise it cannot work normally

3. If the road signs cannot be recognized, you can adjust the bracket of the k210 or the distance of the road signs

4. You must use the road sign model provided by us, refer to the pictures in the road sign folder, other road signs cannot be recognized

5. Under normal circumstances, the motor is very smooth. If there is a motor jam or a large resistance, these conditions indicate that there is a problem with the motor.

6. 1. The patrol map must be kept clean without too many stains, otherwise it will affect the normal use

7. When patrolling the line, the environment should not be too bright or too dark, normal lighting is fine