4. Infrared patrol + ultrasonic obstacle avoidance

1. Learning objectives

Take the Mailun car drive as an example. We have learned to patrol the line through ultrasonic obstacle avoidance and four-way patrol module. Now we combine these two functions to realize patrolling while ultrasonically detecting obstacles and buzzing.

2. Experimental preparation

1. The car wiring has been installed and installed correctly
2. The indicator light of the debugging four-way patrol module is on when it encounters a black line. The indicator light is off when it is not a black line.

3. Implementation principle

We obtain the distance between the car and the obstacle through ultrasonic waves, and we let the car patrol the black line through the four-way patrol module. When the car patrols the line, the ultrasonic wave detects that the obstacle is less than the distance value we set, the car stops, and the buzzer sounds a reminder.

4. Code analysis

Source code path:

/home/pi/project_demo/05.Comprehensive_gameplay/3.ultrasonic_followup.ipynb

```
import sys
sys.path.append('/home/pi/project_demo/lib')
#导入麦克纳姆小车驱动库 Import Mecanum Car Driver Library
from McLumk_Wheel_Sports import *

speed =30
try:
    # 打开超声波测距功能 Turn on the ultrasonic ranging function
    bot.Ctrl_Ulatist_Switch(1)
    while True:
        # 从I2C读取巡线传感器数据 Read line sensor data from I2C
        track_data = bot.read_data_array(0x0a, 1)
        track = int(track_data[0])

        # 解析巡线传感器的状态 Analyze the status of the line patrol sensor
        x1 = (track >> 3) & 0x01
        x2 = (track >> 2) & 0x01
        x3 = (track >> 1) & 0x01
        x4 = track & 0x01
        """
        X2 X1 X3 X4
        |  |  |  |
        L1 L2 R1 R2
        """
        lineL1=x2
        lineL2=x1
        lineR1=x3
        lineR2=x4

        # 读取超声波传感器的距离 Reading distance from ultrasonic sensor
        diss_H =bot.read_data_array(0x1b,1)[0]
        diss_L =bot.read_data_array(0x1a,1)[0]
        dis = diss_H << 8 | diss_L
        if(dis <200):
            stop_robot()
            bot.Ctrl_BEEP_Switch(1)  #蜂鸣器开 Buzzer on
            time.sleep(0.1)
            bot.Ctrl_BEEP_Switch(0)
        else :
            if lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0:  # 都是黑色，加速前进 All black, speed up
                print("1")
                print(lineL1,lineL2,lineR1,lineR2)
                move_forward(int(speed*3/4))
            elif( (lineL2 == 0 or lineL1 == 0) and lineR2 == 0):#右锐角：右大弯,0表示检测到黑线 Right acute angle: right big bend, 0 means black line is detected
                print("2")
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_right(speed)
                time.sleep(0.02)
            elif lineL1 == 0 and (lineR2 == 0 or lineR1 == 0):  # 左锐角或左大弯 Left sharp angle or left sharp bend
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_left(int(speed*1.5))  # 左急转弯 Sharp left turn
                time.sleep(0.15)
            elif lineL1 == 0:  # 左最外侧检测 Left outermost detection
                print("4")
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_left(speed)  # 左急转弯 Sharp left turn
                time.sleep(0.01)
            elif lineR2 == 0:  # 右最外侧检测 Right outermost detection
                print("5")
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_right(speed)
                time.sleep(0.01)
            elif lineL2 == 0 and lineR1 == 1:  # 中间黑线上的传感器微调车左转 The sensor on the middle black line fine-tunes the car to turn left
                print("6")
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_left(int(speed/3))  # 左转 Turn left
            elif lineL2 == 1 and lineR1 == 0:  # 中间黑线上的传感器微调车右转 The sensor on the middle black line fine-tunes the car to turn right
                print("7")
                print(lineL1,lineL2,lineR1,lineR2)
                rotate_right(int(speed/3)) #右转 Turn right
            elif lineL2 == 0 and lineR1 == 0:  # 都是黑色，加速前进 All black, speed up
```

```
                print("8")
                print(lineL1,lineL2,lineR1,lineR2)
                move_forward(speed)



        # 等待一段时间再进行下一次检测 Wait for a while before the next test
        time.sleep(0.01)

except KeyboardInterrupt:
    # 当用户按下 Ctrl+C 时，关闭超声波测距功能 When the user presses Ctrl+C, the ultrasonic ranging function is turned off
    bot.Ctrl_Ulatist_Switch(0)
    # 当用户中断程序时，确保所有电机停止 Ensure that all motors stop when the user interrupts the program
    stop()
    print("Ending")
```

5. Experimental results

We put the car on a map with black lines, make sure the car's four-way patrol module has been adjusted to light up when encountering a black line, and turn off when not encountering a black line, and then run the program. After running the program, we will see the car patrolling the black line slowly. During the patrol process, when encountering an obstacle in front, the car stops and the buzzer sounds. When we remove the obstacle, the car continues to patrol the line and the buzzer stops.