1.Infrared patrol car

1. Learning objectives

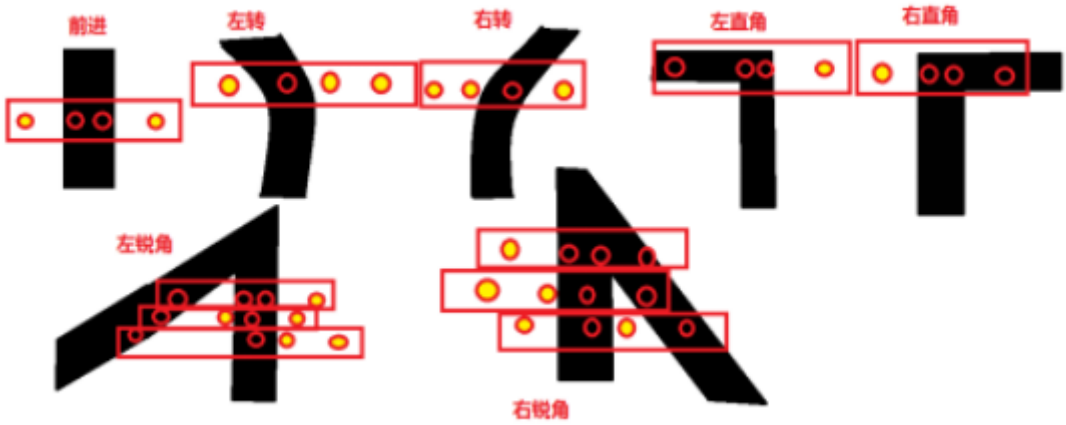
By combining the four-way line patrol module with the car, the car can patrol the black line.

2. Experimental preparation

1. The car wiring has been installed
2. Debug the four-way patrol module. Adjust the knob of the four-way tracking module so that the indicator light is on when encountering a black line; when it is not a black line, the indicator light is off.

3. Implementation principle

When detecting tracks of different shapes, the status of the four-way patrol sensor is as shown in the figure below. Control the car to perform different movements according to the sensor status.



4. Code Analysis

Source code path:

/home/pi/project_demo/05.Comprehensive_gameplay/1.infrared_patrol_line.ipynb

```
import sys
sys.path.append('/home/pi/project_demo/lib')
#Import Mecanum Car Driver Library
from McLumk_Wheel_Sports import *

speed =30#30
try:
```

```
while True:
# Read line sensor data from I2C
track_data = bot.read_data_array(0x0a, 1)
track = int(track_data[0])

# 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

if lineL1 == 0 and lineL2 == 0 and lineR1 == 0 and lineR2 == 0: # All black,
speed up
print("1")
print(lineL1,lineL2,lineR1,lineR2)
move_forward(int(speed))
elif( (lineL2 == 0 or lineL1 == 0) and lineR2 == 0):#Right acute angle: right
big bend, 0 means black line is detected
print("2")
print(lineL1,lineL2,lineR1,lineR2)
rotate_right(speed)
time.sleep(0.05)
elif lineL1 == 0 and (lineR2 == 0 or lineR1 == 0): # Left sharp angle or left
sharp bend
print("3")
print(lineL1,lineL2,lineR1,lineR2)
rotate_left(int(speed*1.5)) # Sharp left turn
time.sleep(0.15)
elif lineL1 == 0: # Left outermost detection
print("4")
print(lineL1,lineL2,lineR1,lineR2)
rotate_left(speed) # Sharp left turn
time.sleep(0.02)
elif lineR2 == 0: # Right outermost detection
print("5")
print(lineL1,lineL2,lineR1,lineR2)
rotate_right(speed)
time.sleep(0.01)
elif lineL2 == 0 and lineR1 == 1: # 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)) # Turn left
elif lineL2 == 1 and lineR1 == 0: # 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)) # Turn right
elif lineL2 == 0 and lineR1 == 0: # All black, speed up
print("8")
print(lineL1,lineL2,lineR1,lineR2)
move_forward(speed)

# Wait for a while before the next test
time.sleep(0.01)

except KeyboardInterrupt:
# Ensure that all motors stop when the user interrupts the program
stop()
print("Ending")
```

5. Experimental results

We put the car on the patrol track and confirmed that the module knob has been adjusted. When it encounters a black line, the indicator light is on, and when it is not a black line, the indicator light is off. After running the code block, the car will patrol the black line.