Raspberry Pi - IO method

Experimental preparation

1. Raspberry Pi motherboard
2. 8-channel line patrol module
3. Several Dupont cables

The Raspberry Pi board needs to download the IO communication source code provided in the document

Experimental purpose

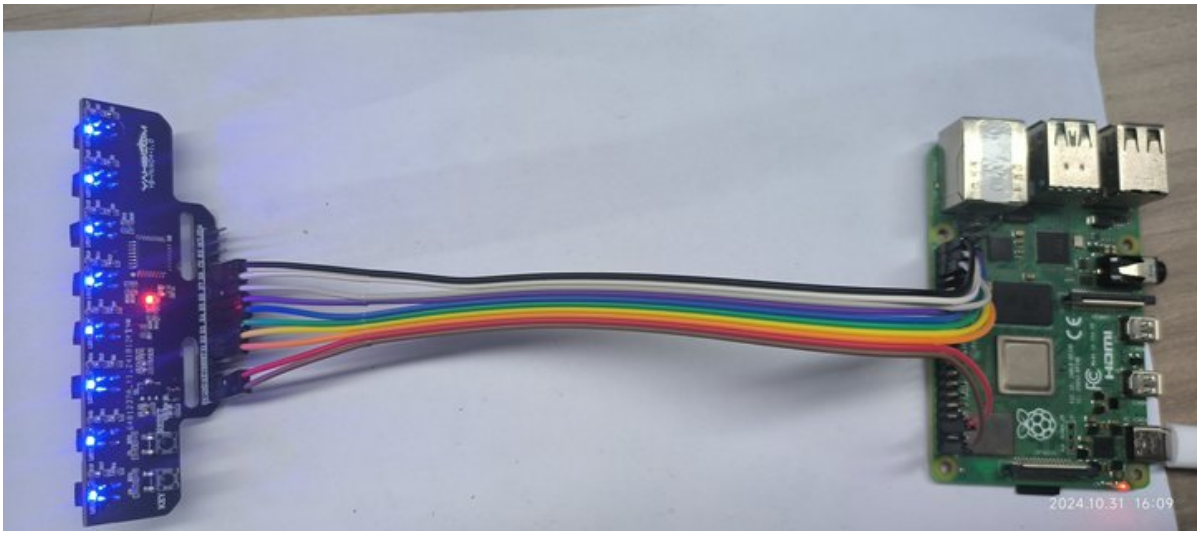
The content of this experiment is mainly to use the Raspberry Pi main control to receive the data of the 8-channel line patrol module through IO.

Experimental wiring

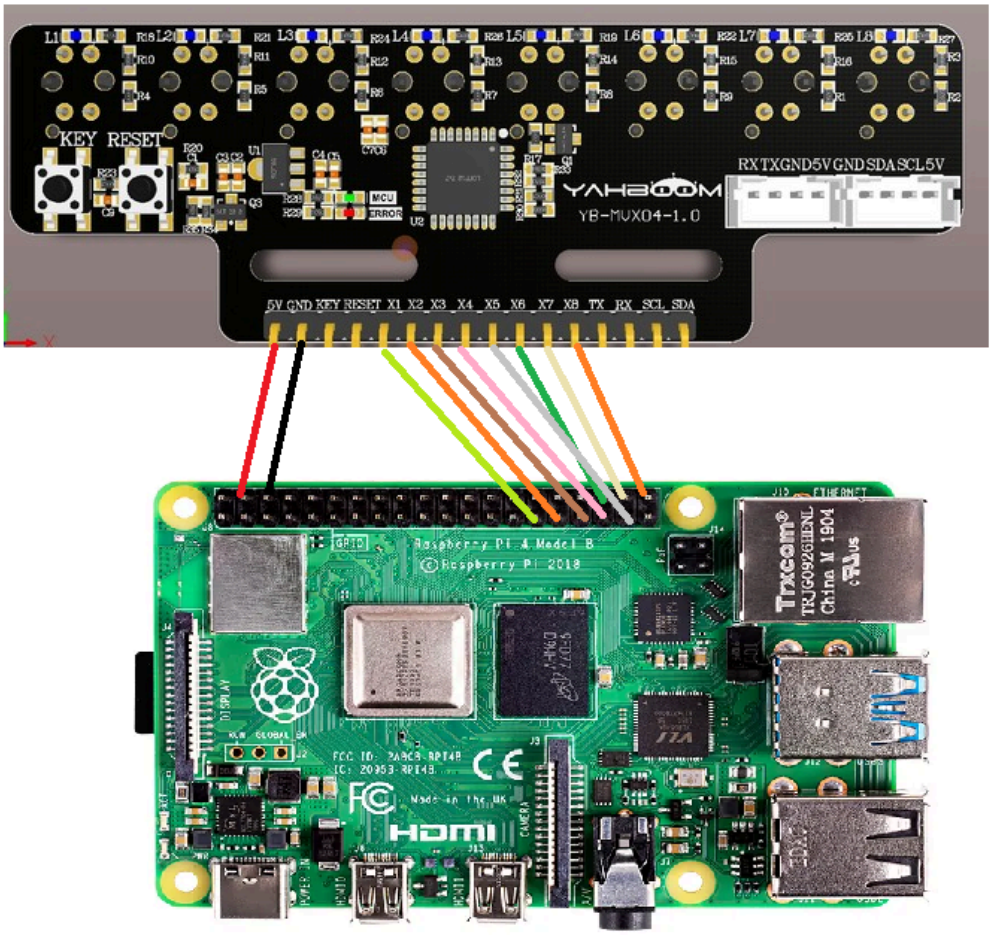
Raspberry Pi	8-channel line patrol module
GPIO.21	x1
GPIO.22	x2
GPIO.23	x3
GPIO.24	x4
GPIO.25	x5
GPIO.27	x6
GPIO.28	x7
GPIO.29	x8
5V	5V
GND	GND

Raspberry Pi 40pin pin comparison table

wiringPi Coding	BCM Coding	Function Name	Physical pin BOARD coding		Function Name	BCM Coding	wiringPi Coding
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29



As shown in the figure:



Experimental steps and phenomena

1. After connecting the wires, **do not power on for wiring, power off for wiring**
   Run the script

```
python3 IR_IO.py
```

```
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:1  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
x1:1  x2:1  x3:0  x4:0  x5:0  x6:0  x7:0  x8:0
```

Experimental source code

```
try:
    while True:
        # Read pin level
        pin_state_x1 = GPIO.input(x1)
        pin_state_x2 = GPIO.input(x2)
        pin_state_x3 = GPIO.input(x3)
        pin_state_x4 = GPIO.input(x4)
        pin_state_x5 = GPIO.input(x5)
        pin_state_x6 = GPIO.input(x6)
        pin_state_x7 = GPIO.input(x7)
        pin_state_x8 = GPIO.input(x8)

        print("x1:"+str(pin_state_x1)+"  x2:"+str(pin_state_x2)+"
x3:"+str(pin_state_x3)+"  x4:"+str(pin_state_x4)+"  x5:"+str(pin_state_x5)+"
x6:"+str(pin_state_x6)+"  x7:"+str(pin_state_x7)+"  x8:"+str(pin_state_x8))
        # Wait for a while
        time.sleep(1)
finally:
    # Clean up GPIO settings
```

```
    GPIO.cleanup()
```

The main function of the source code is very simple. It reads the probe pins of 8 patrol lines through IO and prints them out.