Raspberry Pi - Serial Port 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 serial communication source code provided in the document, and the Raspberry Pi system needs to open the serial communication interface through raspi-config

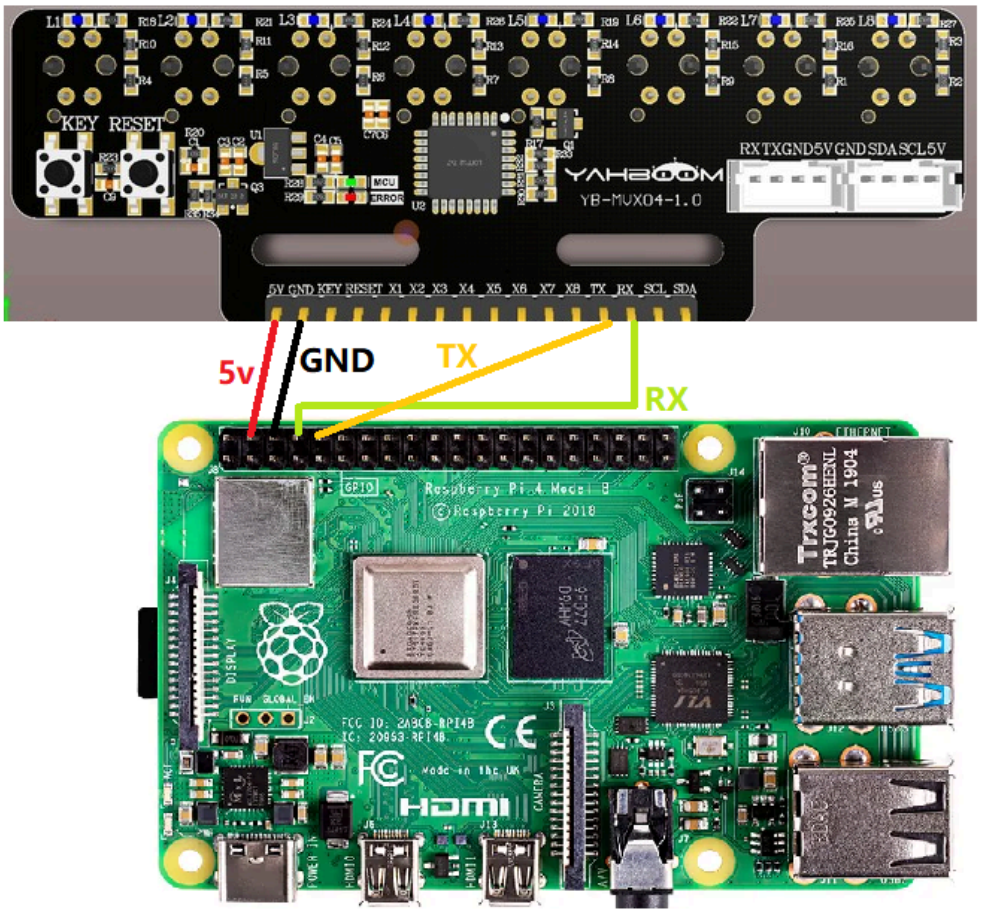
Experimental purpose

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 the serial port.

Experimental wiring

Raspberry Pi	8-channel line patrol module
TX	RX
RX	TX
5v	5v
GND	GND

As shown in the figure:



Open the Raspberry Pi hard serial port (Raspberry Pi 5 does not require this step)

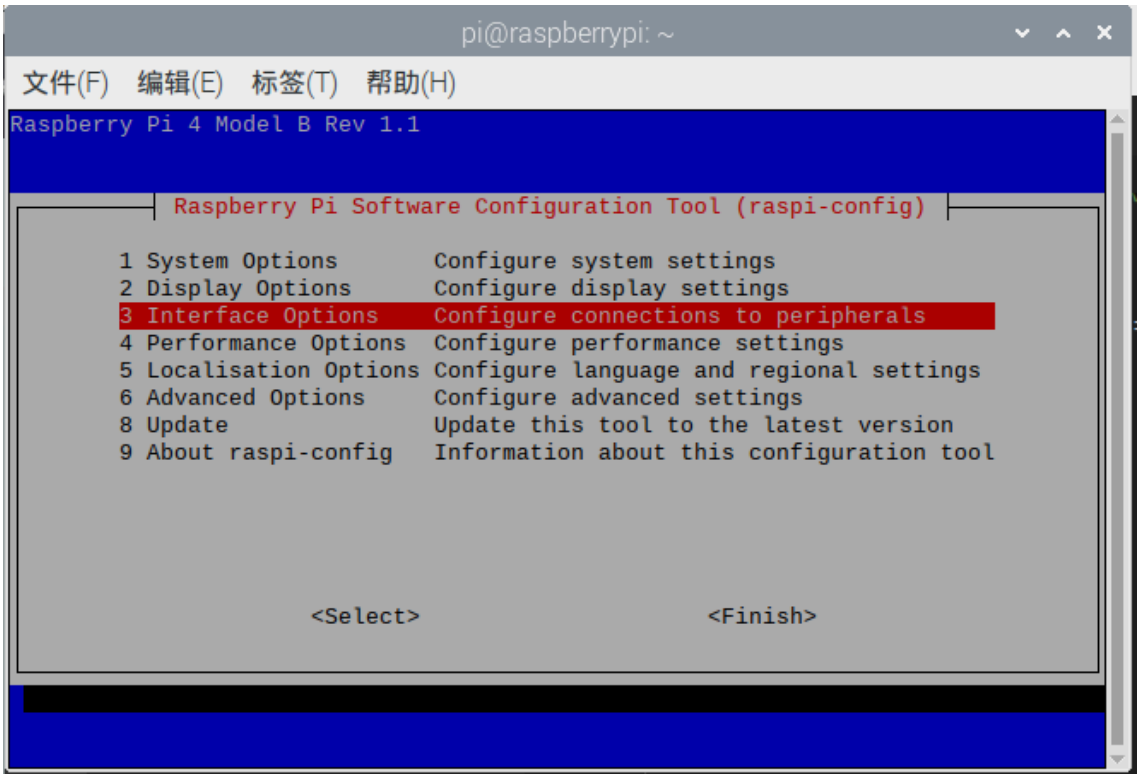
- Configure the Raspberry Pi serial port first, because the Raspberry Pi's hard serial port is used for Bluetooth, and the mini serial port is unstable. This experiment uses the hard serial port. Raspberry Pi pin diagram

- Raspberry Pi 40pin pin comparison table

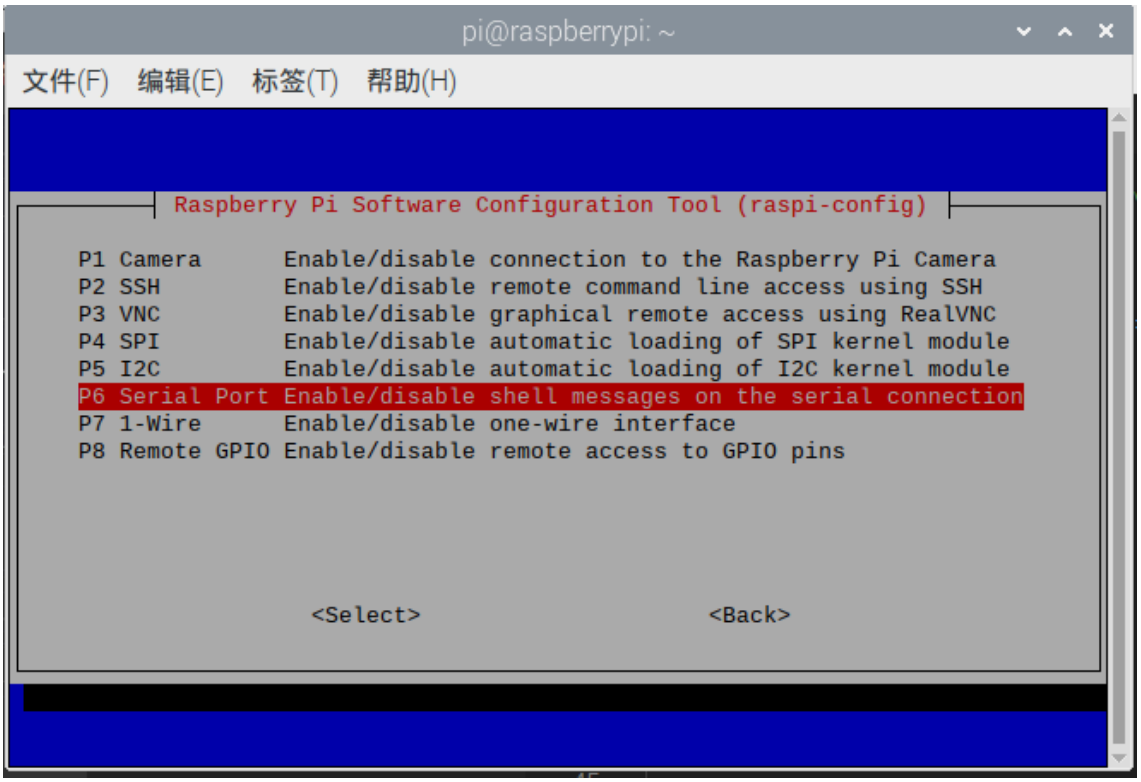
wiringPi Coding	BCM Coding	Function Name	Physical pin BOARD coding		Function Name	BCM Coding	wiringPi Coding
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	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

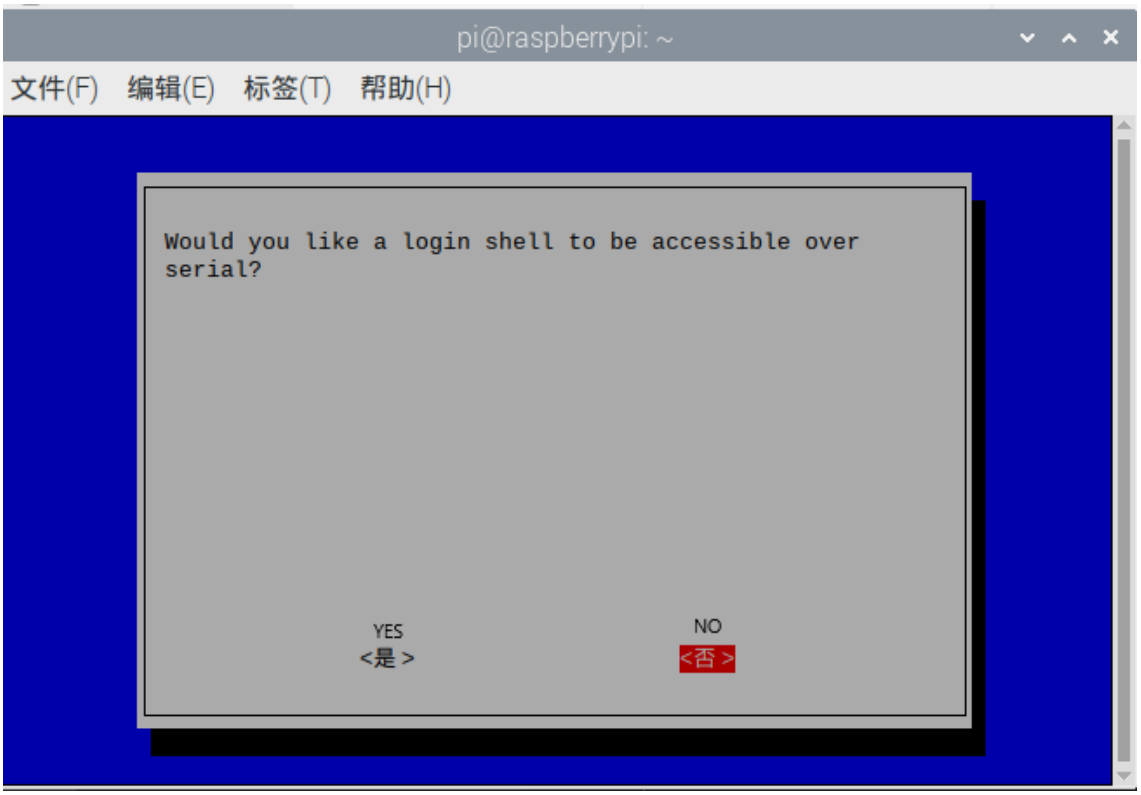
1. First perform the following operations to map the serial port
   Enter `sudo raspi-config` in the terminal

2.


pi@raspberrypi: ~
文件(F) 编辑(E) 标签(T) 帮助(H)
Raspberry Pi 4 Model B Rev 1.1
Raspberry Pi Software Configuration Tool (raspi-config)
1 System Options Configure system settings
2 Display Options Configure display settings
3 Interface Options Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool
<Select>
<Finish>

3.


pi@raspberrypi: ~
文件(F) 编辑(E) 标签(T) 帮助(H)
Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera Enable/disable connection to the Raspberry Pi Camera
P2 SSH Enable/disable remote command line access using SSH
P3 VNC Enable/disable graphical remote access using RealVNC
P4 SPI Enable/disable automatic loading of SPI kernel module
P5 I2C Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins
<Select>
<Back>

4. 

5. 

6. Set the hardware serial port to GPIO serial port, edit /boot/config.txt with root privileges.

   The command is:

```
sudo nano /boot/config.txt
```

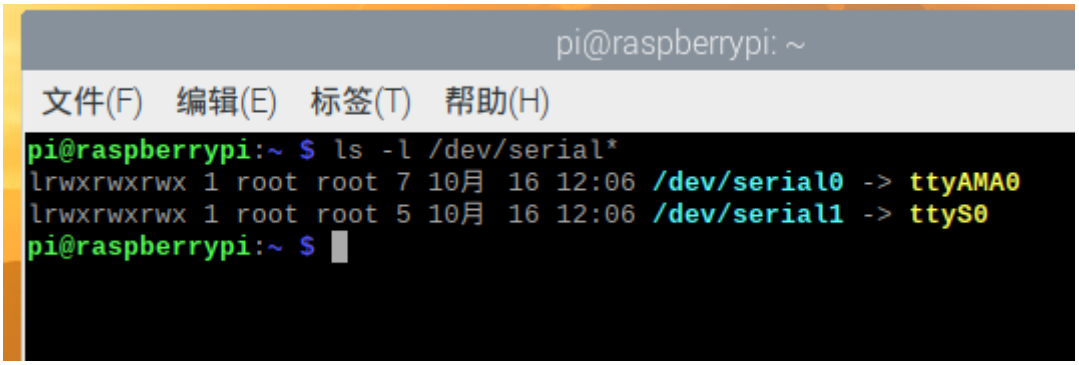
After opening the file, add two lines at the end

```
dtoverlay=miniuart-bt
force_turbo=1
```

```
  GNU nano 3.2                          /boot/config.txt

#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on
start_x=1
gpu_mem=128

dtoverlay=pi3-miniuart-bt

force_turbo=1
```

Save: Ctrl+O, exit: Ctrl+X.

7. Save and exit, then restart the Raspberry Pi and you can see that the serial ports have been swapped.



Reference Links: https://blog.iyatt.com/?p=1817

Experimental steps and phenomena

1. After connecting the wires, run the script

```
python3 UASRT.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

```python
#Main function
if __name__ == "__main__":
    print("start it")

    try:
        while True:
            usart_deal()
    except KeyboardInterrupt:
        pass
    finally:
        ser.write(bytes("$0,0,0#",'utf-8'))
```

This source code only writes numerical data. If you need to parse analog data, you can refer to STM32 serial port parsing