RDKX5 IIC communication

Note: The voice interaction module needs to be burned with factory firmware. If the voice chip has not been flashed with firmware after receiving it, it does not need to be burned

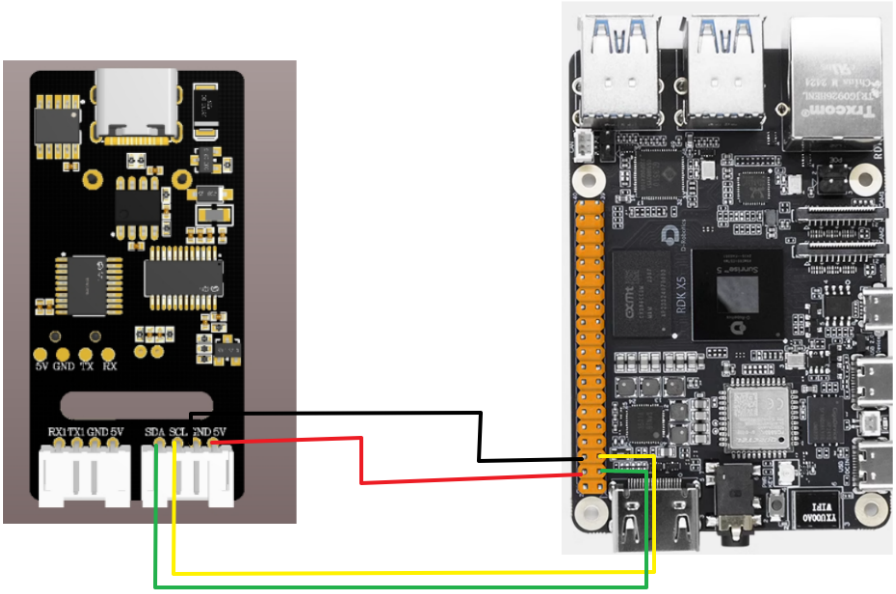
1. Experimental preparation

- RDK X5
- Voice interaction module
- Dupont line

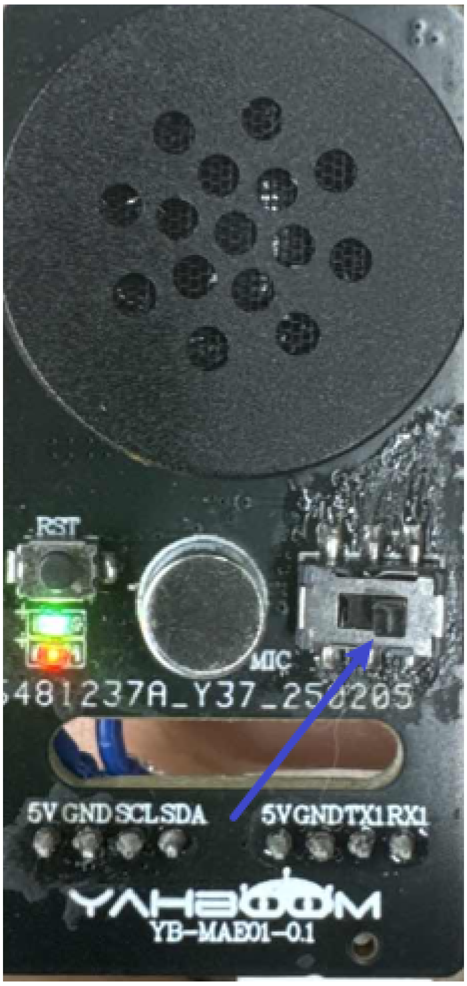
2. Wiring diagram

- Wiring diagram

RDKX5	Voice interaction module
3	SDA
5	SCL
GND	GND
5V	5V

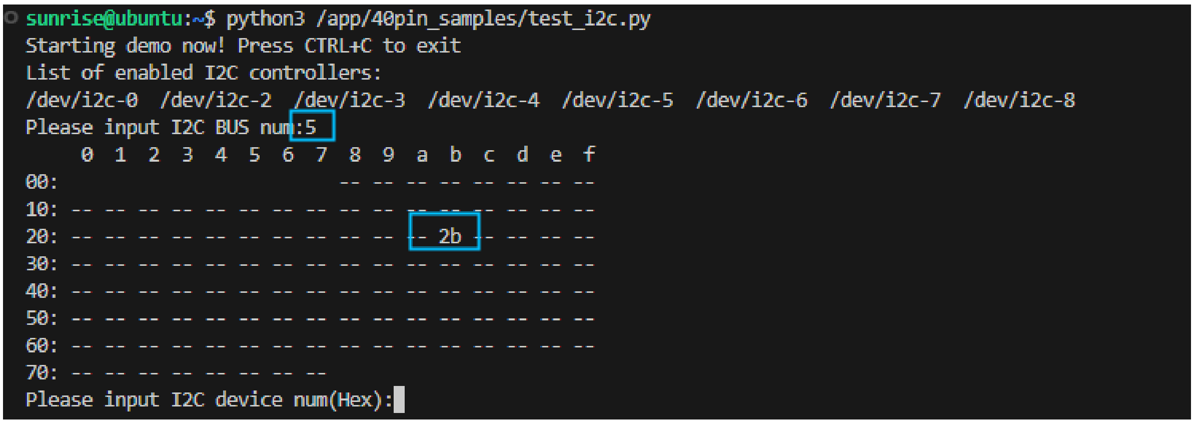


- Slide the module to the right, use the serial port of the stc firmware,



- Input in the terminal, then input 5 and press Enter, the iic device address appears, indicating that it has been recognized normally

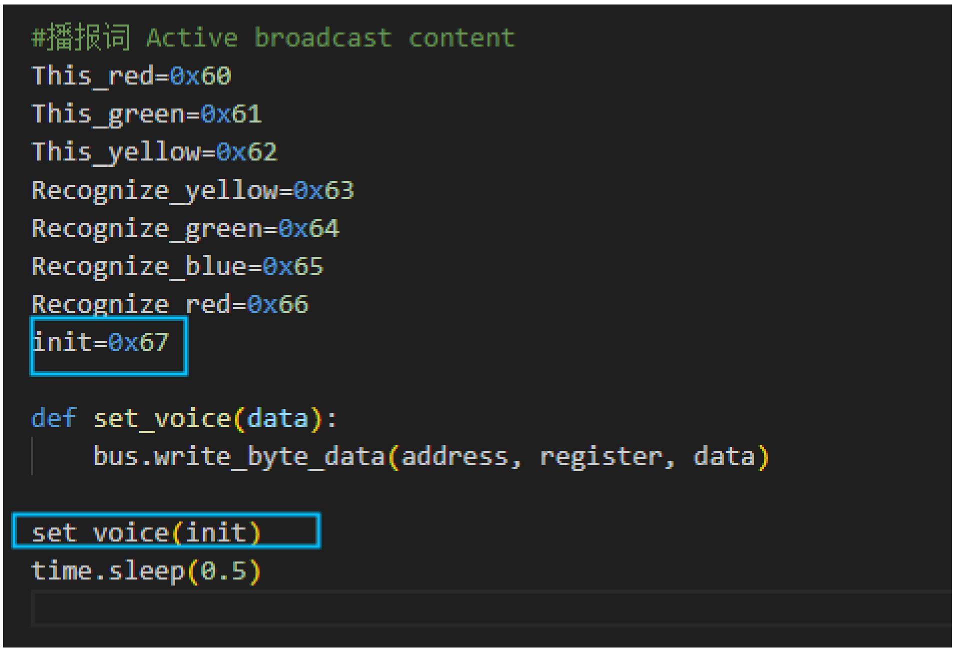
```
python3 /app/40pin_samples/test_i2c.py
```



```
sunrise@ubuntu:~$ python3 /app/40pin_samples/test_i2c.py
Starting demo now! Press CTRL+C to exit
List of enabled I2C controllers:
/dev/i2c-0  /dev/i2c-2  /dev/i2c-3  /dev/i2c-4  /dev/i2c-5  /dev/i2c-6  /dev/i2c-7  /dev/i2c-8
Please input I2C BUS num:5
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- 2b -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
Please input I2C device num(Hex):
```

3.Achieve the effect

- Select the broadcast content by modifying the code in the program as shown below



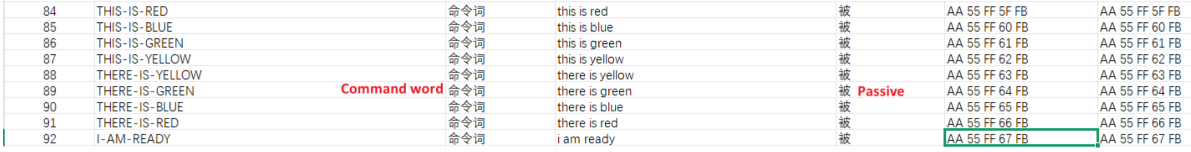
```
#播报词 Active broadcast content
This_red=0x60
This_green=0x61
This_yellow=0x62
Recognize_yellow=0x63
Recognize_green=0x64
Recognize_blue=0x65
Recognize_red=0x66
init=0x67

def set_voice(data):
    bus.write_byte_data(address, register, data)

set_voice(init)
time.sleep(0.5)
```

- The content of the broadcast can be viewed according to the **Command Word Broadcast Word Protocol List V3_EN** file provided in the attachment.

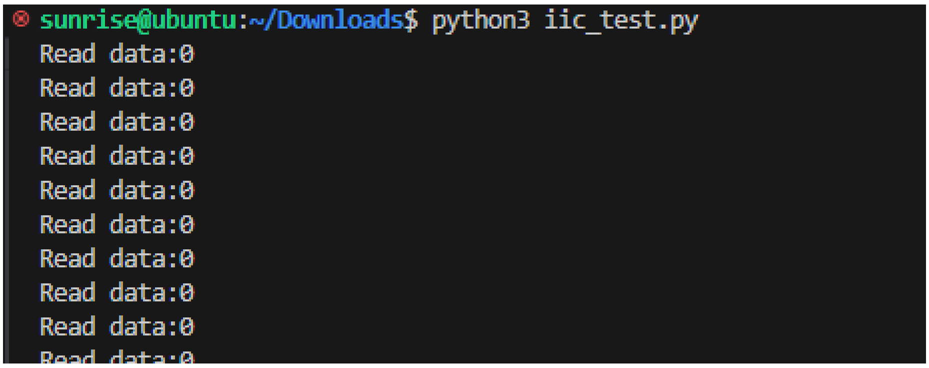
The first and second bytes AA FF represent the frame header of the protocol, the third byte FF represents the broadcast function, and the fourth is the ID of the broadcast content. Here you can see **"I am ready"** is 67 in hexadecimal, so sending 0x67 to register 0x03 in the program can broadcast the corresponding content. The fifth byte is the end frame



84	THIS-IS-RED	命令词	this is red	被	AA 55 FF 5F FB	AA 55 FF 5F FB
85	THIS-IS-BLUE	命令词	this is blue	被	AA 55 FF 60 FB	AA 55 FF 60 FB
86	THIS-IS-GREEN	命令词	this is green	被	AA 55 FF 61 FB	AA 55 FF 61 FB
87	THIS-IS-YELLOW	命令词	this is yellow	被	AA 55 FF 62 FB	AA 55 FF 62 FB
88	THERE-IS-YELLOW	命令词	there is yellow	被	AA 55 FF 63 FB	AA 55 FF 63 FB
89	THERE-IS-GREEN	命令词	there is green	被	AA 55 FF 64 FB	AA 55 FF 64 FB
90	THERE-IS-BLUE	命令词	there is blue	被	AA 55 FF 65 FB	AA 55 FF 65 FB
91	THERE-IS-RED	命令词	there is red	被	AA 55 FF 66 FB	AA 55 FF 66 FB
92	I-AM-READY	命令词	i am ready	被	AA 55 FF 67 FB	AA 55 FF 67 FB

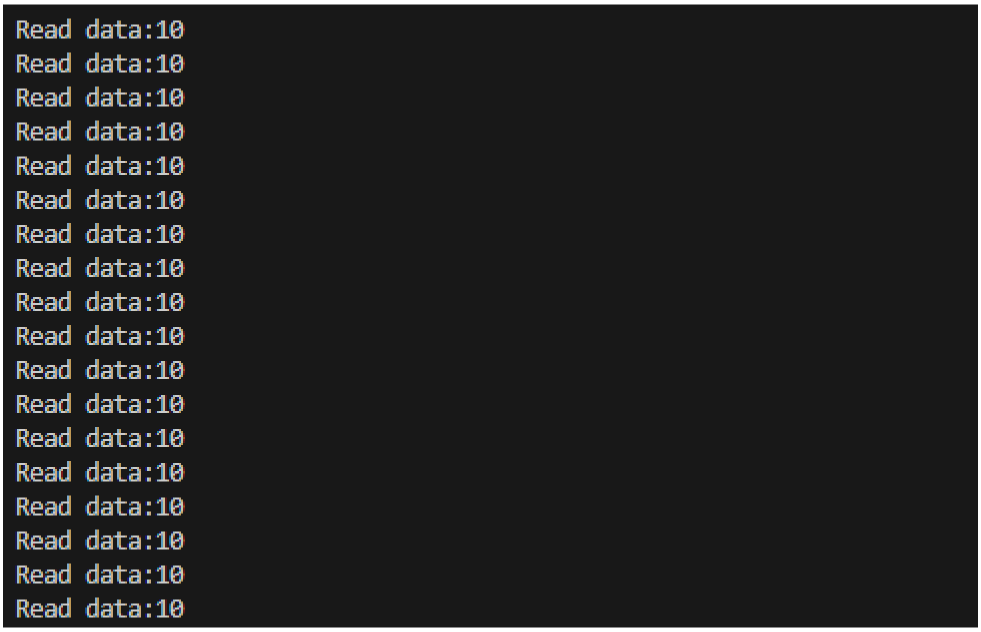
- Enter the following command in the terminal to run the program. When you hear **I am ready**, it means the program is running

```
python3 iic_test.py
```



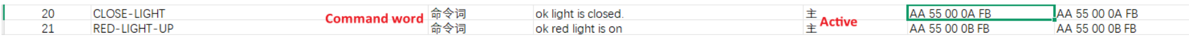
```
sunrise@ubuntu:~/Downloads$ python3 iic_test.py
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
Read data:0
```

- After I say the wake-up word to wake up, I say **"close light"** and the debugging assistant will reply to receive 10



```
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
Read data:10
```

- At this time, you can open the attached **Command Word Broadcast Word Protocol List V3_EN** file to view the "Turn off the light" protocol



20	CLOSE-LIGHT	Command word	命令词	ok light is closed.	主	Active	AA 55 00 0A FB	AA 55 00 0A FB
21	RED-LIGHT-UP		命令词	ok red light is on	主		AA 55 00 0B FB	AA 55 00 0B FB

The first and second bytes AA FF represent the frame header of the protocol, the third byte represents the ID of the ten function words of the chip, and the fourth is the command word ID. Here you can see **"close light"** is hexadecimal 0A, decimal 10. The fifth byte is the end frame.

- Say other command words, the serial port debugging assistant will also print the corresponding command word ID, you can try it yourself