STM32-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

- STM32F103C8T6 minimum core board
- Voice interaction module
- 4pin DuPont adapter cable

2. Wiring diagram

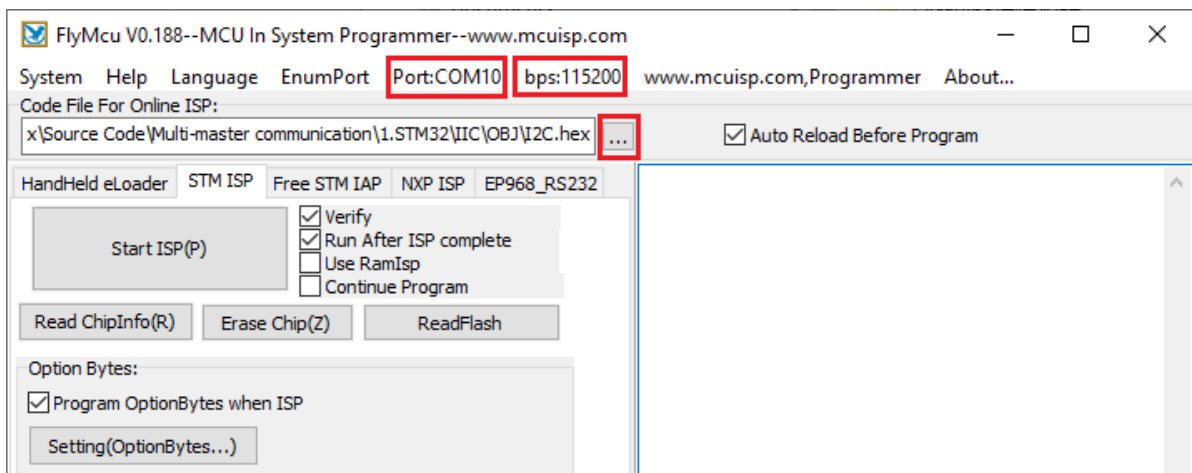| STM32	Voice interaction module
PB10	SCL
PB11	SDA
GND	GND
5V	5V
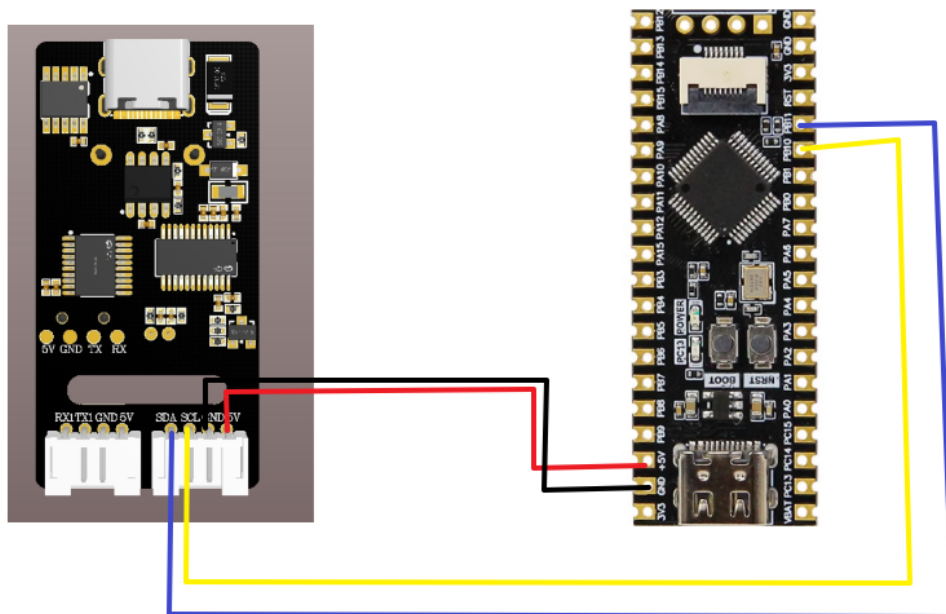
3. Program download

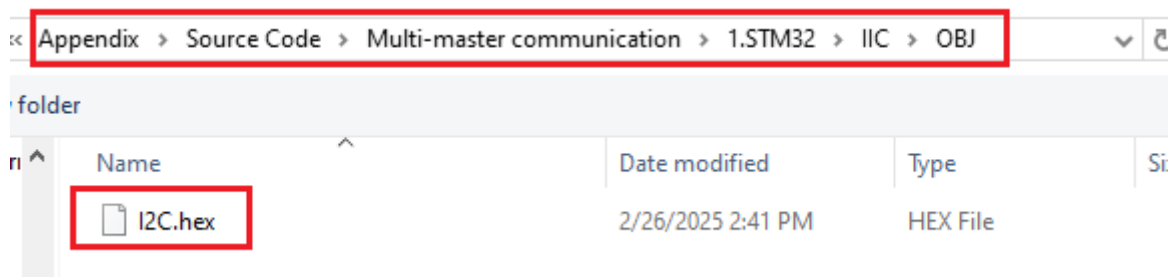- Connect the stm32 motherboard and the computer with a Type-C data cable

Type-C

- Open the stm32 firmware burning tool **FlyMcu.exe**, select the corresponding device port, bit rate 115200, and burn the program to the development board, as shown below
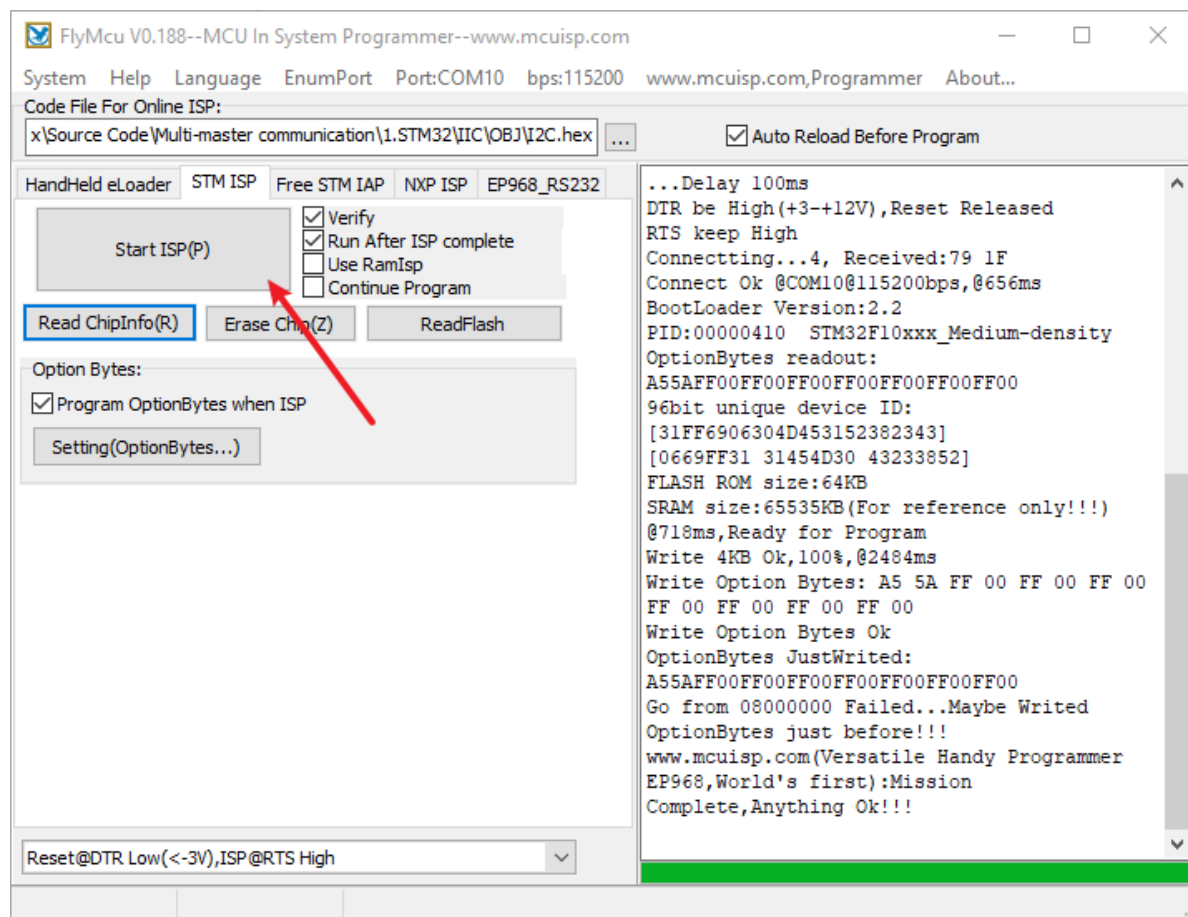

- Click the three dots and select the hex file under the OBJ file in the iic folder under the stm32 communication source code.

- Select the option below in the software
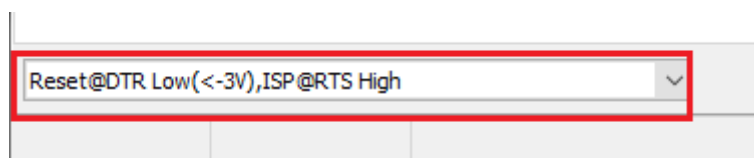
- Click Start Programming to write the program to the stm32 development board

4.Achievement effect

- After the burning is completed, we power on the stm32 development board again, press the stm32 reset button, and you can hear the voice module broadcast **"I am ready"**, which means that the program is running normally
- You can select the broadcast content by modifying the code in the program as shown below

```
//主动播报内容 Active broadcast content
#define This_red 0x5F
#define This_blue 0x60
#define This_green 0x61
#define This_yellow 0x62
#define Recognize_yellow 0x63
#define Recognize_green 0x64
#define Recognize_blue 0x65
#define Recognize_red 0x66
#define init 0x67
```

```
Broadcast(init); //设置播报内容 Set the content of the announcement
```

- The broadcast content 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 the program sends 0x67 to register 0x03 to 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 command word	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 Passive		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
93	WHICH-IS-THE-FIRST-COLOR-TO-SORT	命令词	Which is the first color to sort	被		AA 55 FF 68 FB	AA 55 FF 68 FB

- Open the serial port debugging assistant provided in the attachment, select the corresponding port, and set the baud rate to 115200
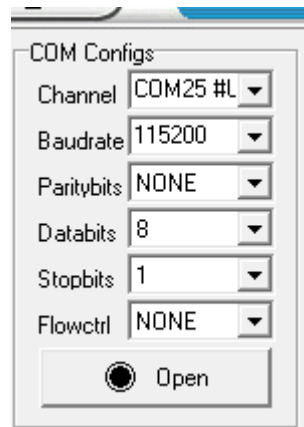
- Click open to see that the terminal will print the received command word ID
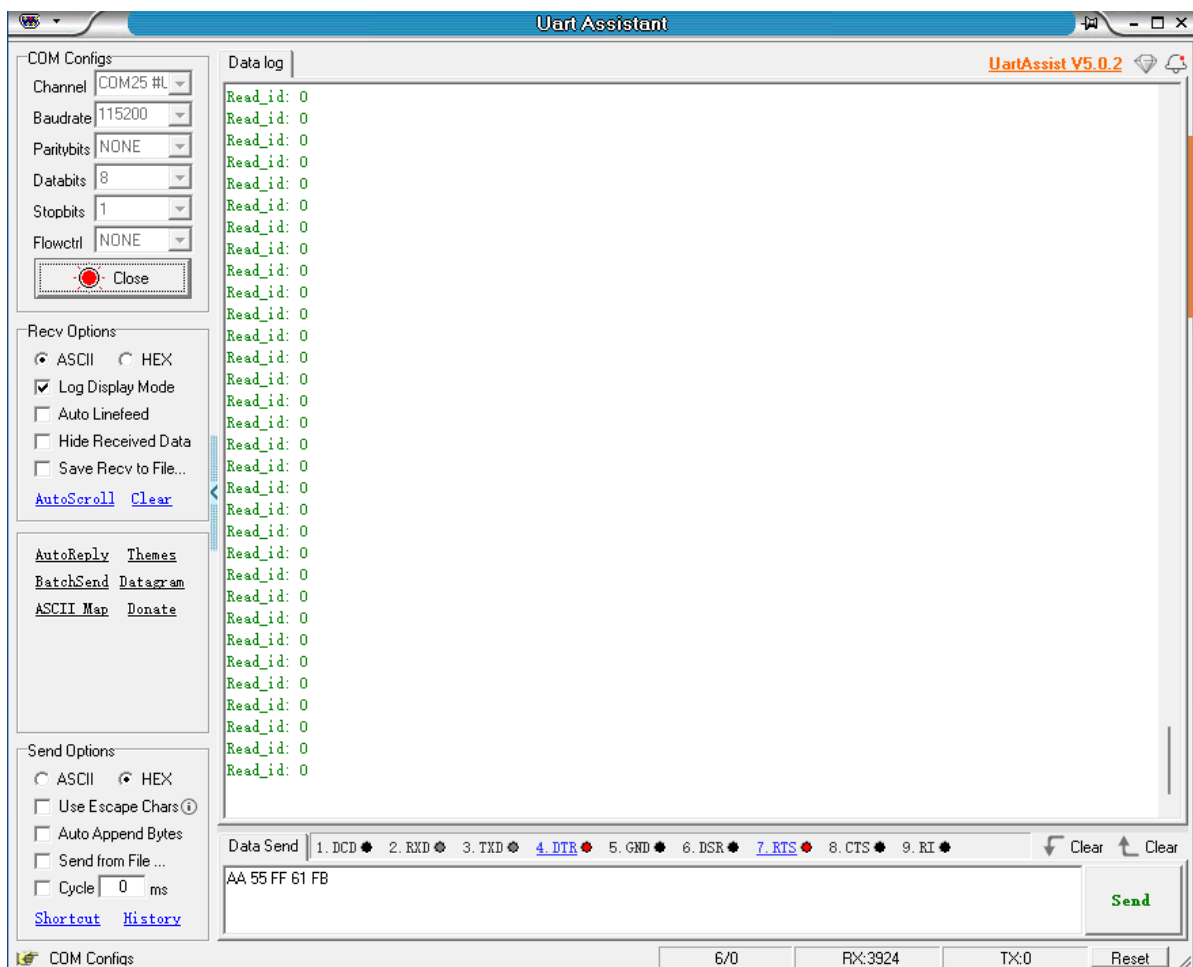
- After I say the wake-up word to wake up, I say **"Close light"** and the debugging assistant will reply with the receiving ID: 10
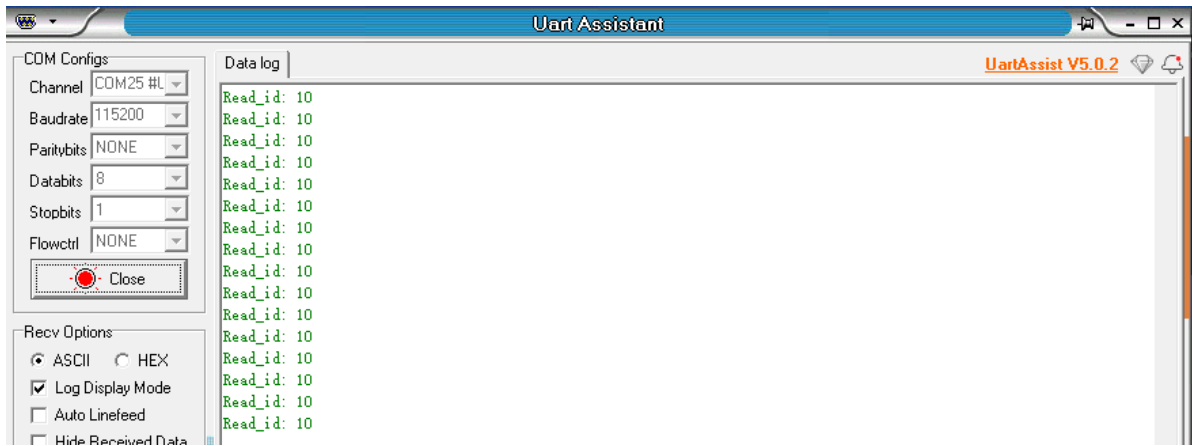
- At this time, you can open the attached **Command Word Broadcast Word Protocol List V3_EN** file to view the "Turn off the light" protocol

| 20 | CLOSE-LIGHT | 命令词 Command word | 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 0A in hexadecimal, so the decimal will return 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