# ESP32 IIC communication
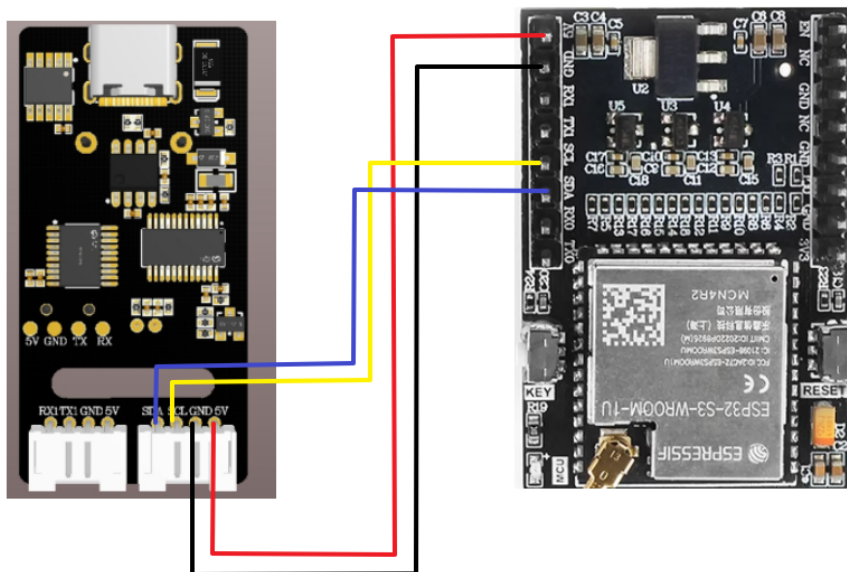
**Note: The voice interaction module needs to be burned with factory firmware. If the voice chip has not been flashed with firmware after receiving it, it does not need to be burned**

## 1. Experimental preparation
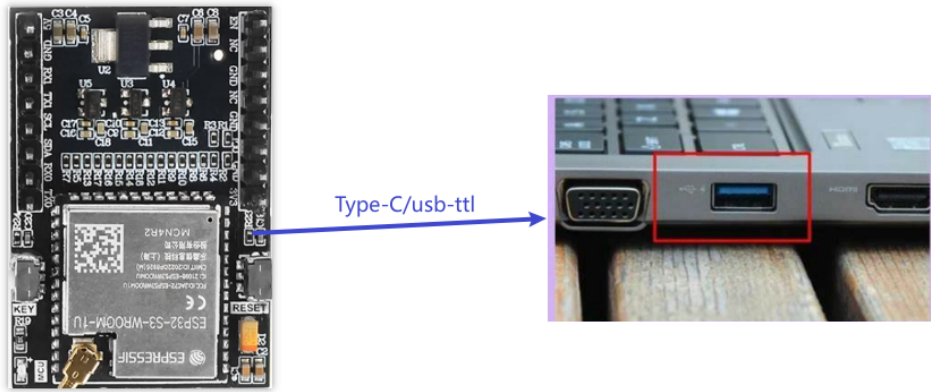
- ESP32
- Voice interaction module
- Dupont line

## 2. Wiring diagram

| ESP32 | Voice interaction module |
| --- | --- |
| 38 | SDA |
| 37 | SCL |
| GND | GND |
| 5V | 5V |



## 3. Program download

- Connect ESP32 to the computer using a serial port module or Type-C
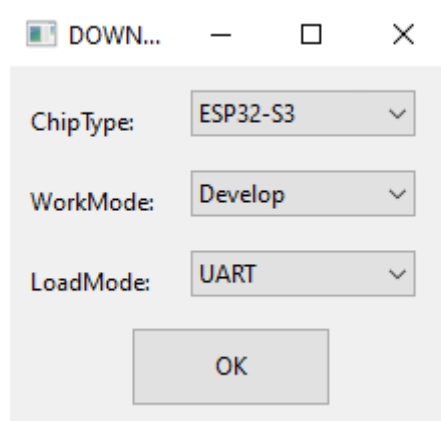
- Download Flash Tool

Download URL:

```
https://www.espressif.com.cn/en/support/download/other-tools
```

**Flash Download Tools**

| | Title | Platform | Version | Release Date ▾ | Download |
|---|---|---|---|---|---|
| ☐ | + Flash Download Tools | HTML | latest | 2024.12.18 |  |

Unzip to get flash_download_tool, double-click to open.

As shown in the figure below, select serial port to burn ESP32-S3. Click OK to open the burning tool.



- Factory firmware burning

Select the firmware to be burned to ESP32S3 in 'SPIDownload'. The correspondence between the file and the address is shown in the following table. Then select the connected COM port and keep the other configurations as default.

| File name | Firmware address | Notes |
|---|---|---|
| bootloader.bin | 0x0000 | Boot file |
| partition-table.bin | 0x8000 | Partition table file |
| microROS_Robot.bin | 0x10000 | Function file |

Select the bin file in the corresponding file directory of the source code,
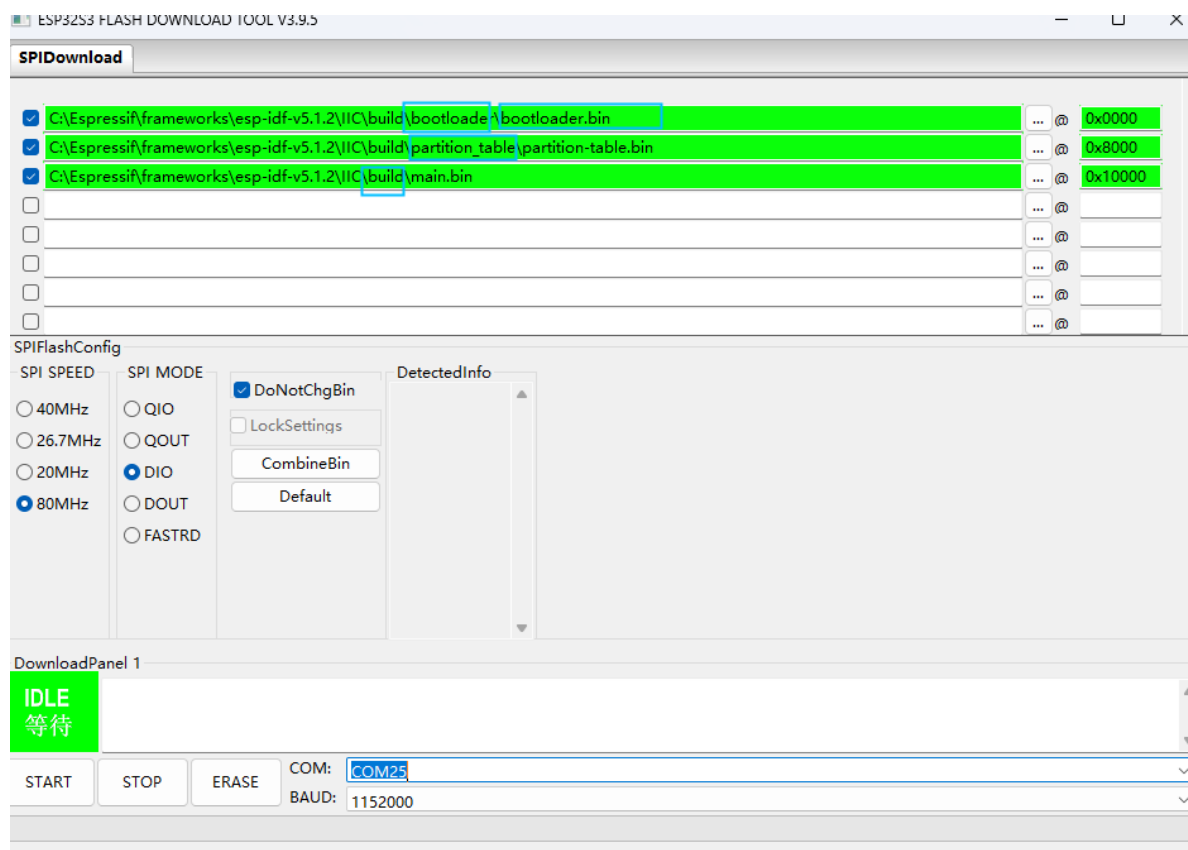
Click the Start button, and the tool will automatically start burning the firmware.

Note: If the firmware does not start burning automatically, please press and hold the boot0 key, then press the reset key, release the boot0 key, and manually enter the burning mode.
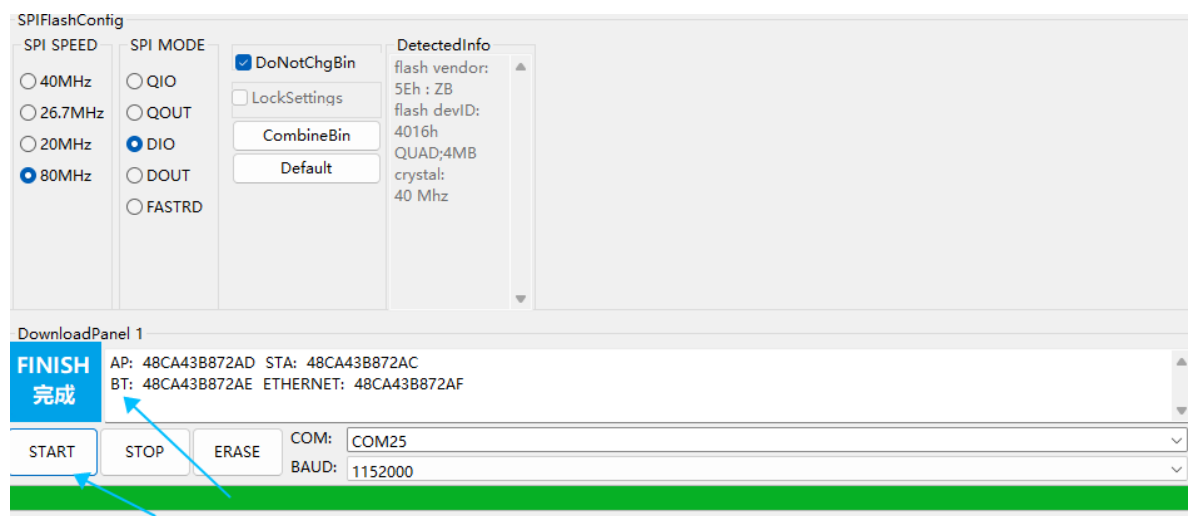


- When you hear the voice module announce **"I am ready"**, it means the program has been written successfully.

# 4.Achievement effect

- You can select the broadcast content by modifying the code in the program as shown below

```
//播报词 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
```

```
void app_main(void)
{
    printf("hello yahboom\n");

    I2C_Master_Init();

    I2C_Master_Write_Byte(VoicceADDR,Write_register,init);

    while (1)
    {
        read_camera_data();
    }

}
```

- The broadcast content can be viewed according to the **Command Word Broadcast Word Protocol List V3_EN** file provided in the attachment.
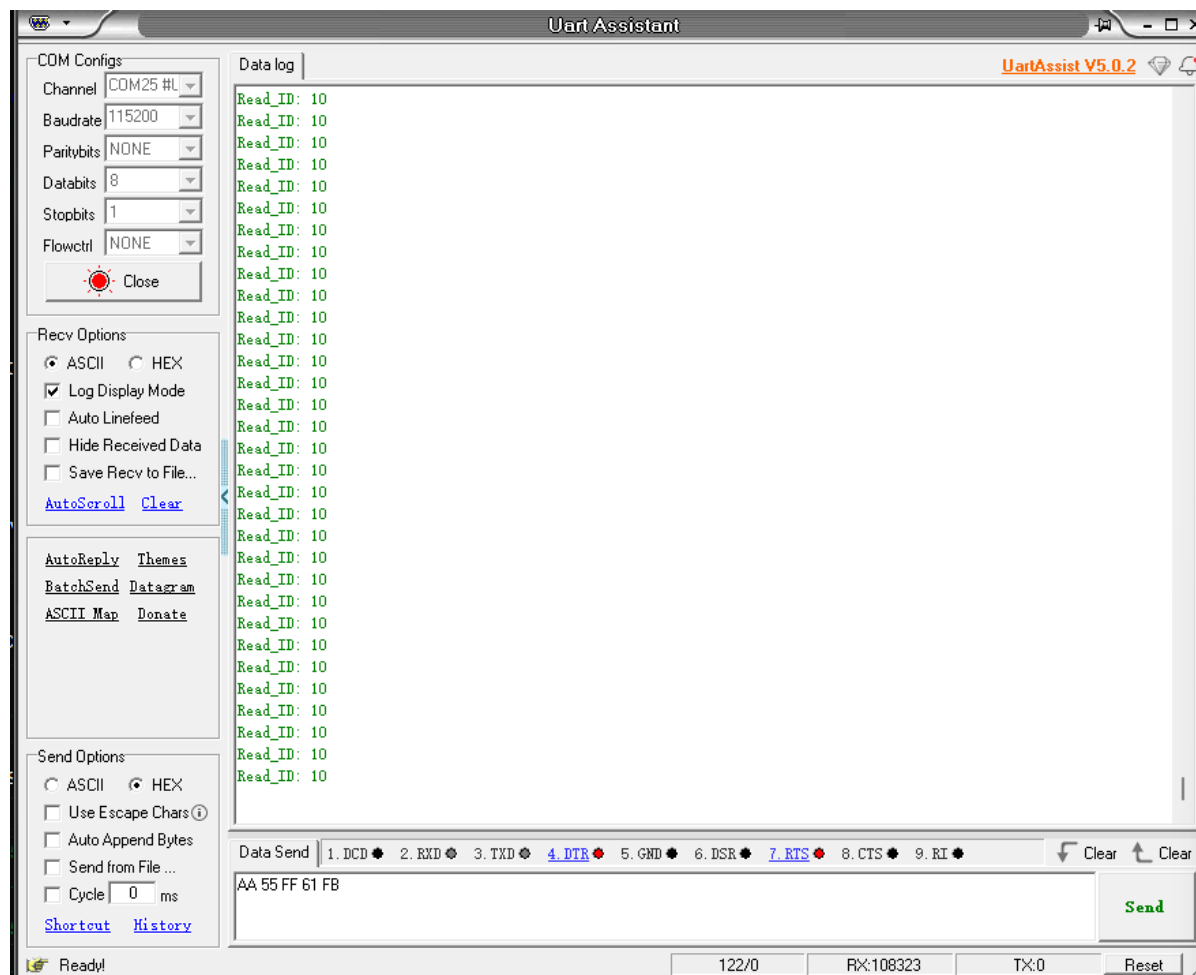
The first and second bytes AA FF represent the frame header of the protocol, the third byte FF represents the broadcast function, and the fourth is the ID of the broadcast content. Here you can see **"I am ready"** is hexadecimal 67, so the program sends 0x67 to register 0x03 to broadcast the corresponding 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 | there is yellow | 被 | AA 55 FF 63 FB | AA 55 FF 63 FB |
| 89 | THERE-IS-GREEN | **Command word** 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 |

- Open the serial port debugging assistant provided in the attachment, select the corresponding port and the baud rate is 115200, and you can see that the terminal will print the received command word ID
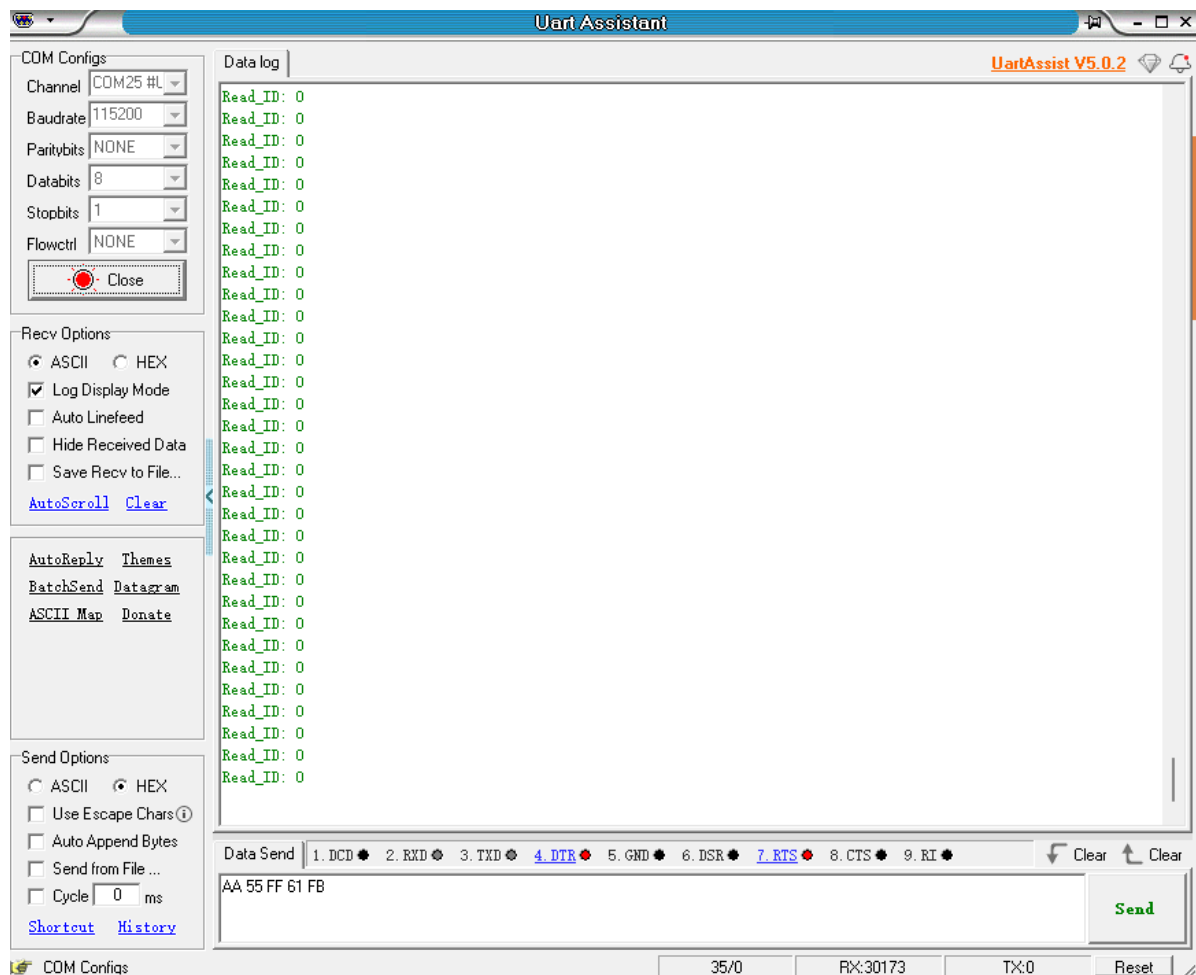
- After I say the wake-up word to wake up, say **"Close light"**, the debugging assistant will reply with the receiving ID: 10

- 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 | 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, so 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