

ESP32-S3-SIM7670G-4G

From Waveshare Wiki

Jump to: navigation, search

Overview

The ESP32-S3-SIM7670G-4G (hereinafter referred to as the development board) is a multifunctional, high-performance microcontroller development board designed by Waveshare. It integrates an SIM7670G 4G communication module, a universal OV camera interface, a TF card slot, RGB colorful LED, an 18650 battery holder (18650 battery is NOT included), a battery voltage measurement IC, a solar panel charging interface, and other peripherals. It employs the ESP32-S3R2, a System On Chip (SoC) that integrates low-power Wi-Fi and BLE5.0. Additionally, it comes with an external 16MB Flash and 2MB PSRAM. The SoC incorporates a hardware encryption accelerator, random number generator (RNG), HMAC, and digital signature modules, meeting the security requirements of the Internet of Things (IoT). The A7670E 4G communication module provides mobile network capabilities, enabling functionalities like portable Wi-Fi and IoT data transmission when combined with the ESP32-S3R2. Its various low-power operating modes cater to power consumption demands in IoT, mobile devices, outdoor monitoring, smart home applications, and other scenarios.

ESP32-S3-SIM7670G-4G



(<https://www.waveshare.com/esp32-s3-sim7670g-4g.htm>)

ESP32-S3R2

Type C UART GPIO

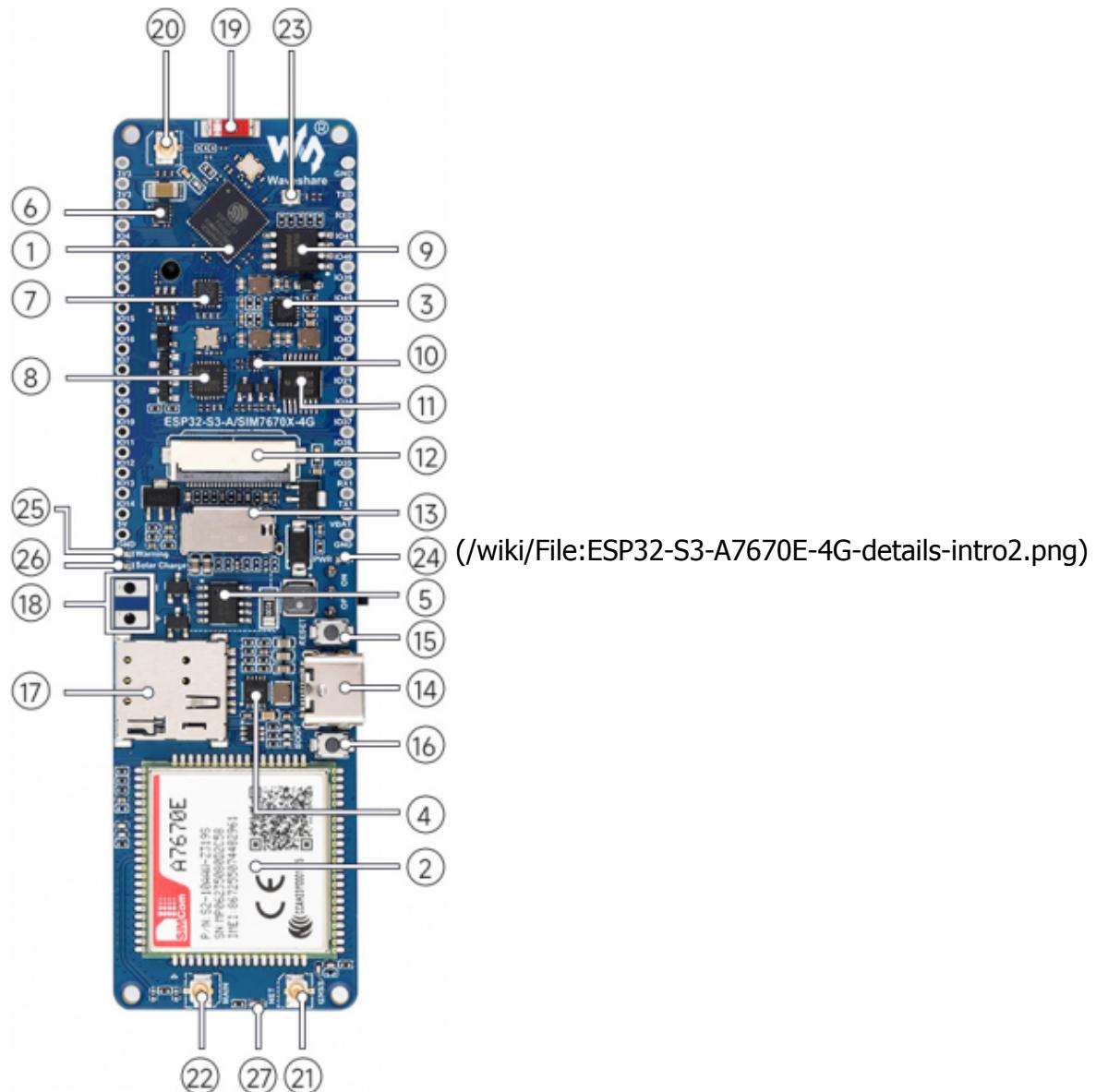
Features

- Comes with a high-performance Xtensa® 32-bit LX7 dual-core processor with up to 240MHz.
- Support 2.4GHz WiFi (802.11 b/g/n) and Bluetooth® 5 (LE) with onboard antenna.
- Built-in 512KB of SRAM and 384KB ROM, with onboard 2MB PSRAM and an external 16MB Flash memory.

Hardware Description

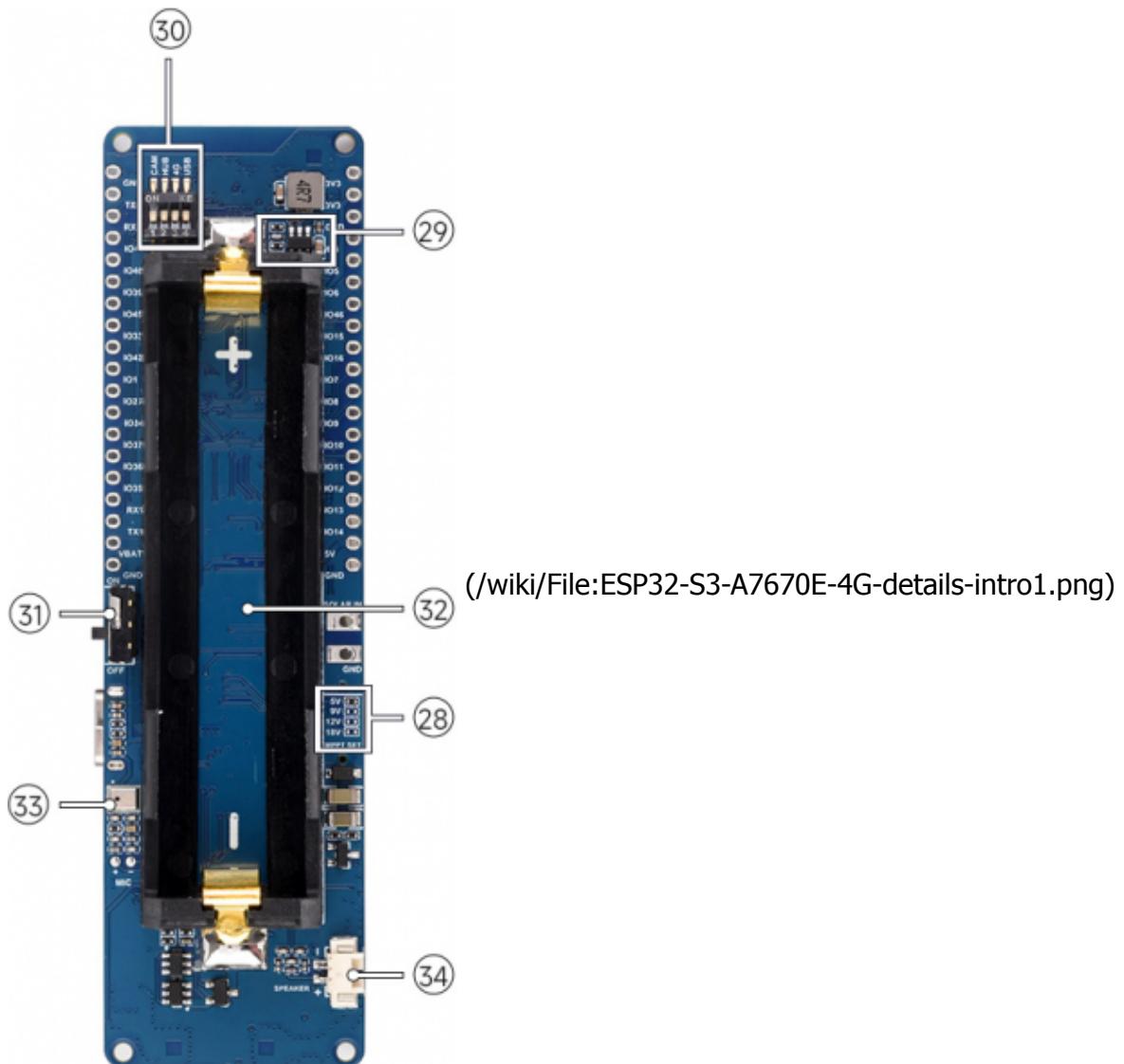
- Onboard patch antenna, you can use it or short it to connect the external antenna as shown in ⑯ and ⑰.
- Onboard RGB colorful beads, WS2812B driver as shown in ⑲.
- Onboard Camera interface: using 24pin camera interface as shown in ⑳.

The supported camera list is shown below:



model	max resolution	color type	Len Size
OV2640	1600 x 1200	color	1/4"
OV3660	2048 x 1536	color	1/5"
OV5640	2592 x 1944	color	1/4"
OV7670	640 x 480	color	1/6"
OV7725	640 x 480	color	1/4"
NT99141	1280 x 720	color	1/4"
GC032A	640 x 480	color	1/10"
GC0308	640 x 480	color	1/6.5"
GC2145	1600 x 1200	color	1/5"
BF3005	640 x 480	color	1/4"
BF20A6	640 x 480	color	1/10"
SC101IOT	1280 x 720	color	1/4.2"
SC030IOT	640 x 480	color	1/6.5"
SC031GS	640 x 480	color	1/6"

- Onboard TF-Card slot, support storing files and pictures as shown in ⑯.
- Onboard solar panel charging interface as shown in ⑰.
 - You can select different resistor values on the back to switch the solar input voltage.
 - When the solar panel is charging, the onboard green LED will light up, as shown in figures ㉖ and ㉘.
- The onboard circuit switch allows you to control the circuit on/off when using the 18650 power supply, as shown in figure ⑳.
- Onboard USB to UART chip and automatic download circuit, after connecting the Type-C cable to program the demo and firmware, as shown in ⑦.
- Using ESP32-S3 USB to connect to the A/SIM7670X USB connector, and using TinyUSB protocol to realize ppp dial-up network access as the portable WIFI.
- The onboard 18650 battery interface is designed for a single 3.7V 18650 lithium battery. Pay attention to the polarity markings on the lithium battery interface.
 - When the battery is reversed, the onboard yellow LED will light up as a warning, as shown in ㉕.
- The development board reserves GPIO pins for external device connections, which can be flexibly configured as I2C, SPI, and other peripheral functions. For detailed functions, please refer to the GPIO allocation description.
- Onboard a GNSS IPEX1 interface. After powering on, you can enable the GNSS positioning function using the relevant commands, as shown in ㉑.

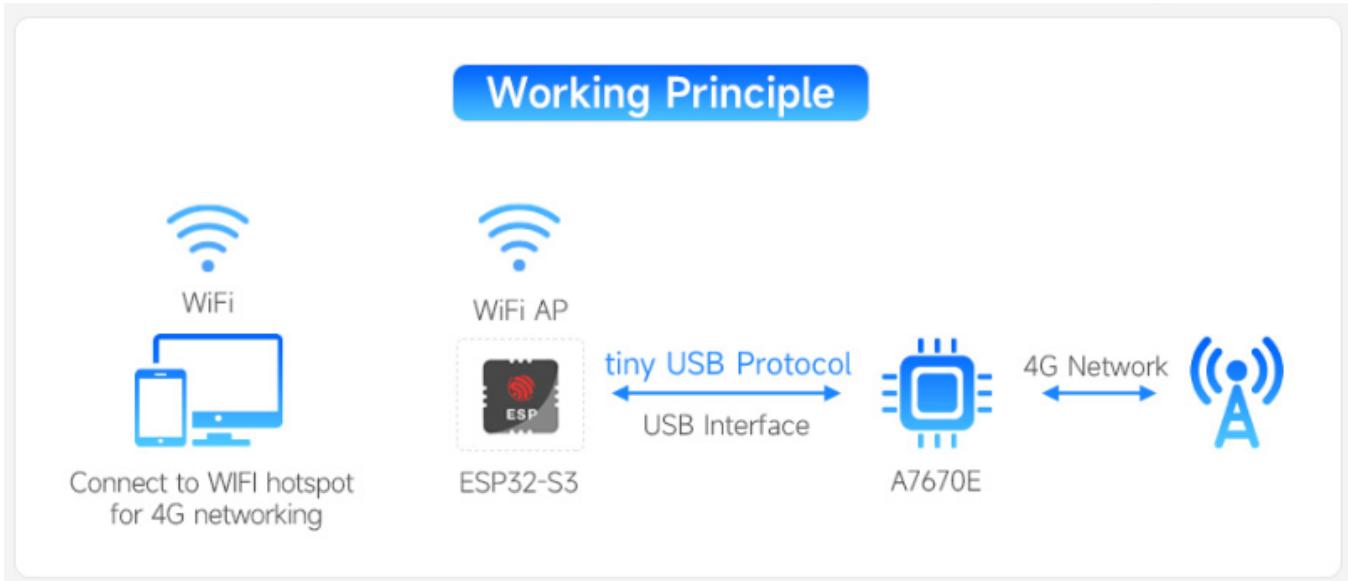


(/wiki/File:ESP32-S3-A7670E-4G-details-intro1.png)

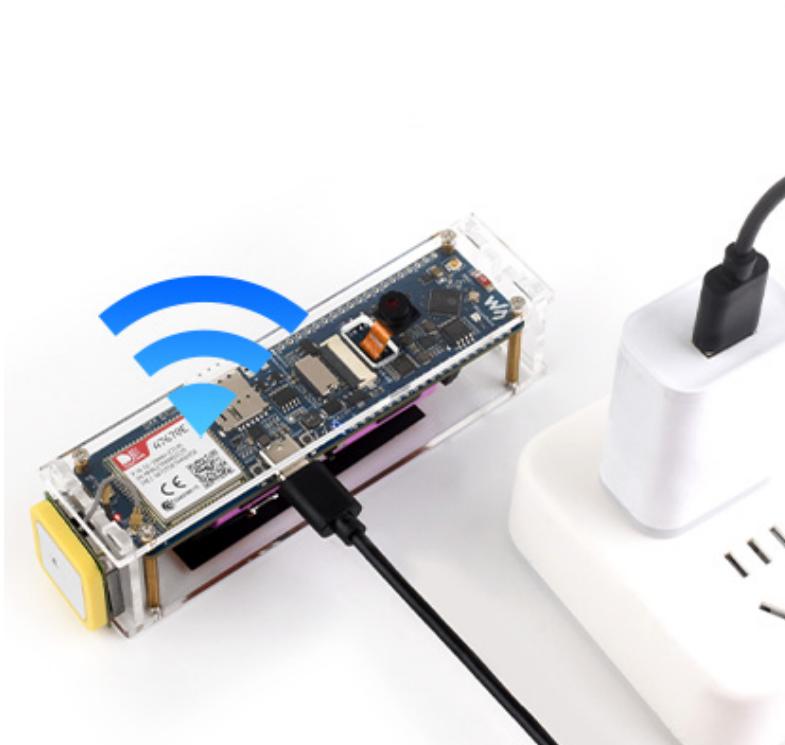
- Furthermore, there is a DIP switch onboard for convenient control of the Camera, USB HUB circuit, and 4G module power supply. It can also control the USB circuit switching of the 4G module, as depicted in figure ⑩.
- Onboard LED Description:
 - Onboard battery anti-reverse warning LED, the yellow LED is on when the batteries are reversely connected as shown in ⑪.
 - Onboard green solar panel charging LED, the LED is on when the solar panel input voltage is active, as shown in ⑫.
 - Onboard blue power indicator, lights on when the power is connected to start up, as shown in ⑬.
 - The onboard module network indicator will turn on in red once the module is powered on. After successfully registering to the network, it will flash at a frequency of 200ms, as shown in figure ⑭.

Hardware Connection

This development board's ESP32-S3 UART to USB module and 4G module USB use the same Type-C interface, and you can choose the connection between the USB interface of the 4G module and the ESP32-S3 USB connector or the Type-C interface through the USB channel of the dip switch on the back of the development board. This function is mainly used when the ESP32-S3 uses Tiny USB 4G module communication as a portable WiFi, wireless hotspot.



(/wiki/File:ESP32-S3-A7670E-4G-details-11.png)



(/wiki/File:Esp32-s3-a-

sim7670x_240124_05.png)

(/wiki/



File:Esp32-s3-a-sim7670x_240124_04.png)

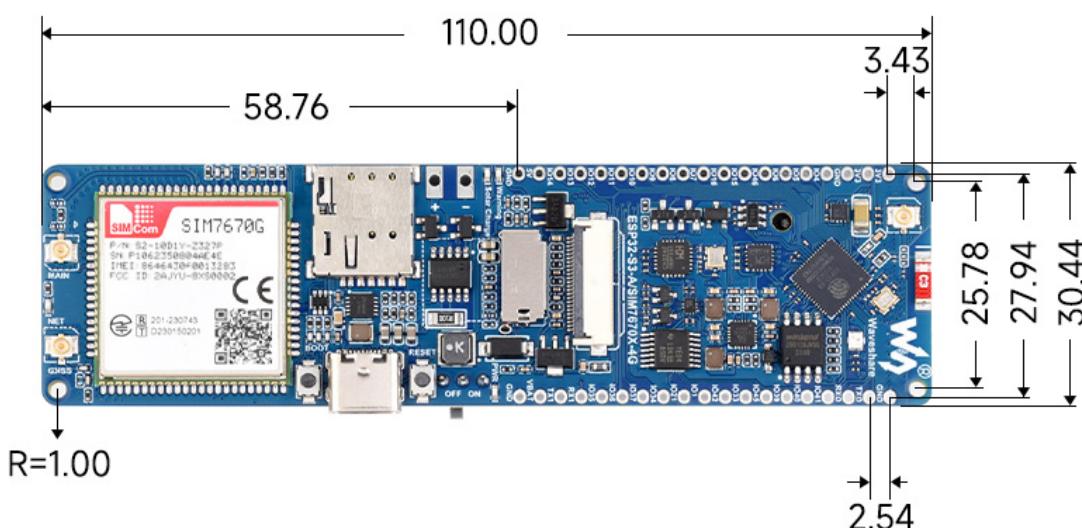
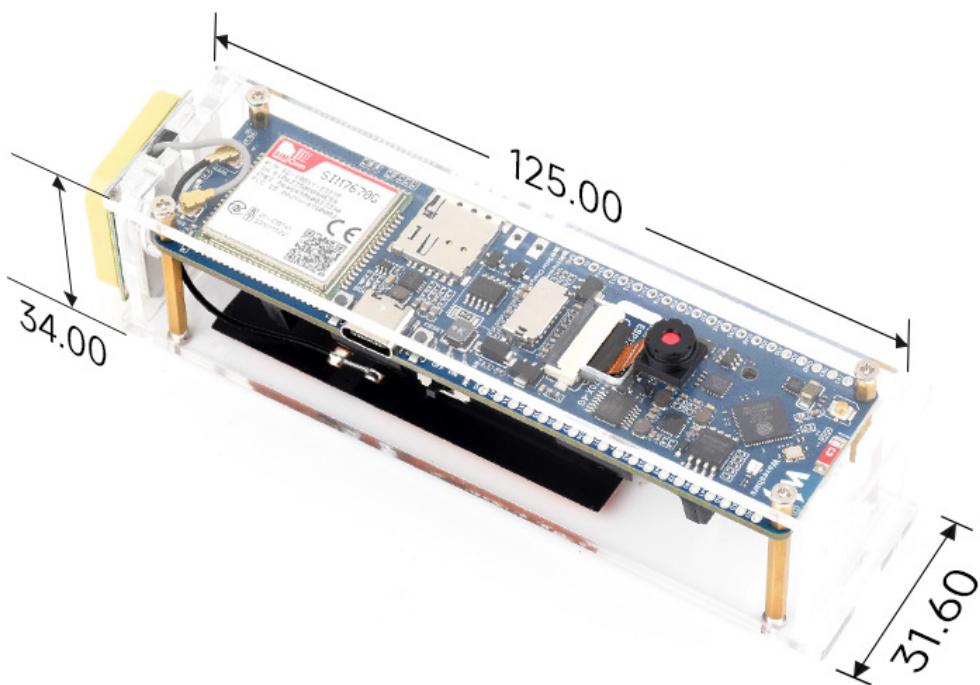
Solar Panel Charging

The solar input selection resistor on the back of the development board can switch the maximum voltage of solar input. By default, the 0-ohm resistor is used to connect to the 5V marking position, supporting solar panels with 5~6V voltage input. When using solar panels with higher voltage input, the corresponding voltage solder joints should be shorted.



(/wiki/File:ESP32-S3-A7670E-4G-Solar.jpg)

Dimensions



Unit:mm

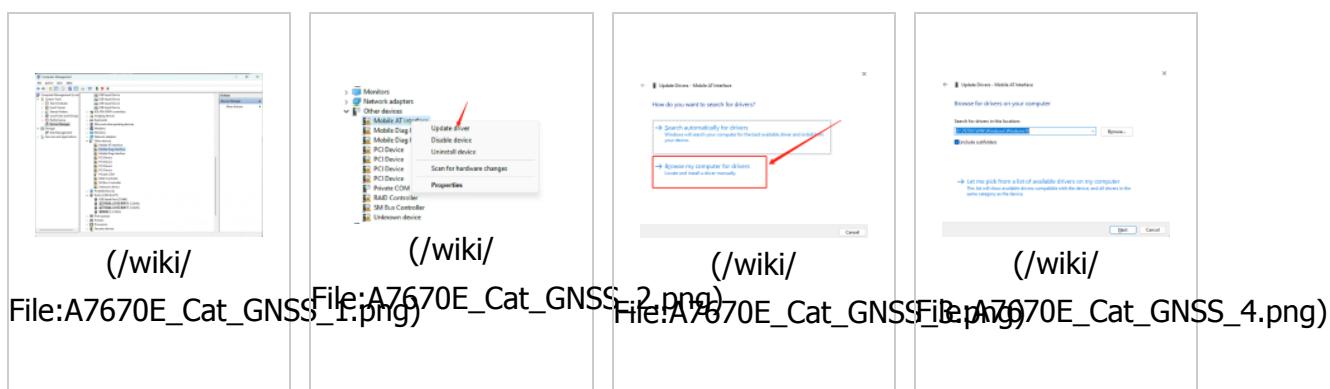
(/wiki/File:ESP32-S3-SIM7670G-4G-Dim.jpg)

Install Driver

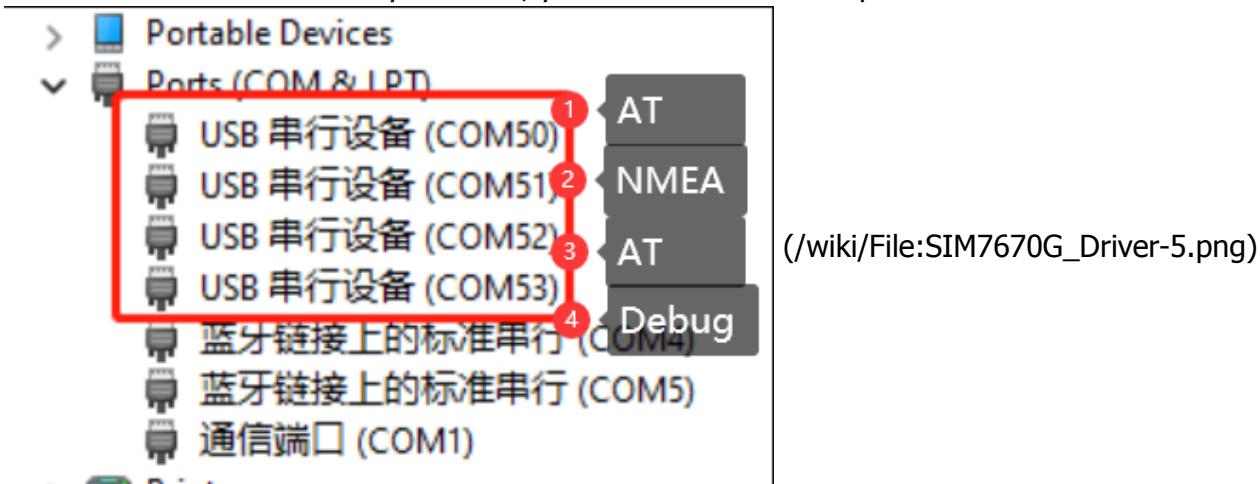
- Before using the module, you need to prepare the following things except Type C USB cable, GNSS antenna and LTE antenna:

a 4G SIM card (GPRS is available)

- In case of power failure, insert the activated 4G SIM card, and connect the USB cable to the computer.
- Connect one end of the Type C USB cable to the PC's USB port and the other end to the USB port of the ESP32-SIM7670G . The PWR light will illuminate, and wait approximately 0 to 1 second for the module to start. The NET light will remain steady, indicating successful module startup. Wait for the module to automatically search for the network; the NET light will begin to flash.
- Open the device manager; for first-time use, install the Windows driver (<https://files.waveshare.com/wiki/A7670E-Cat-1-GNSS-HAT/A7600X-Windows-Driver.7z>). Refer to the following image for installation instructions:



- After the driver is successfully installed, you can see more COM ports as shown below:



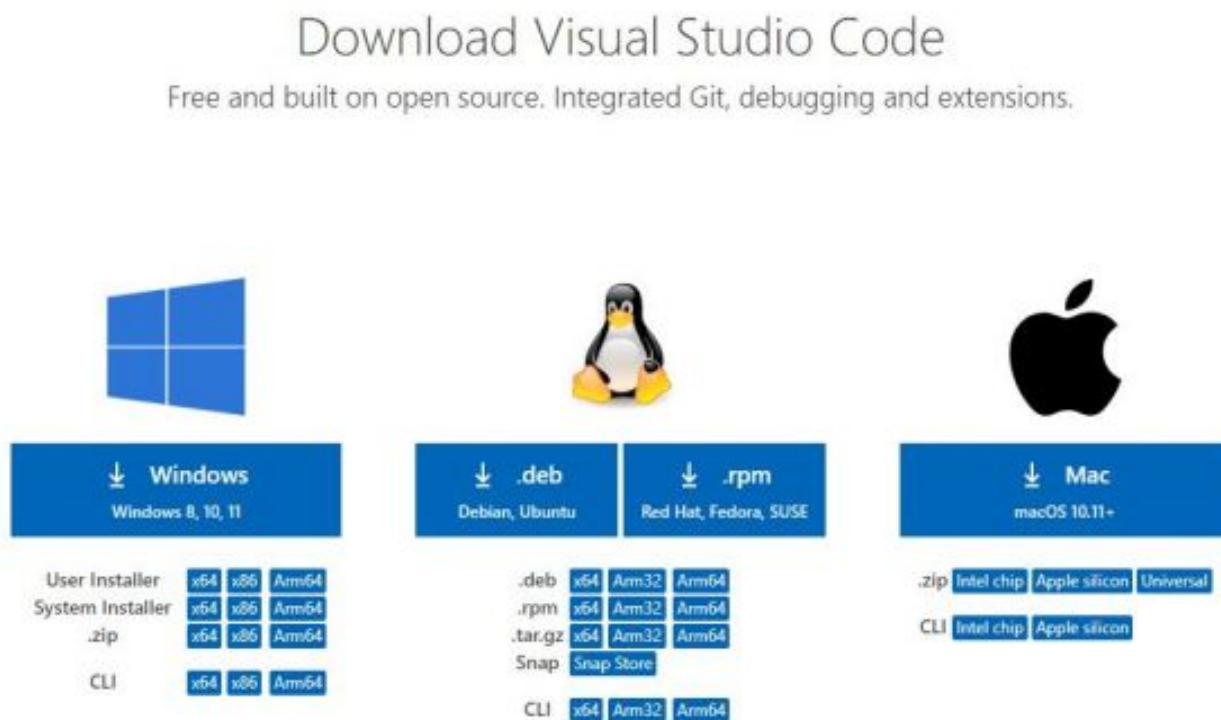
Working with ESP-IDF

This chapter introduces the setup of the ESP-IDF environment, including the installation of Visual Studio, the Espressif IDF plugin, program compilation and download, as well as testing of sample programs. It aims to help users gain proficiency with the development board for secondary development.

Environment Setup

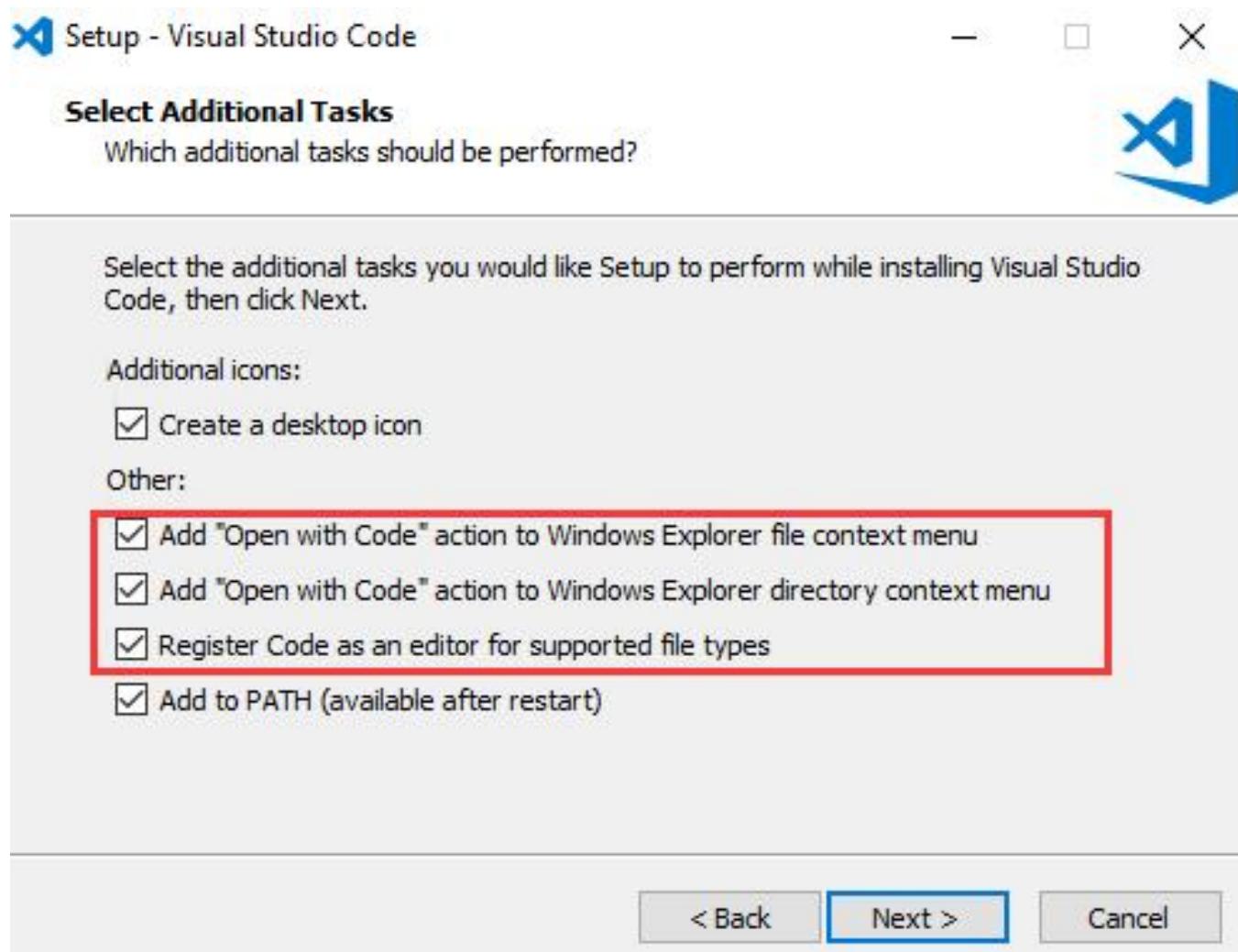
Download and Install VSCode

- Open the download page (<https://code.visualstudio.com/download>) of the official VSCode website, and select the corresponding system and system bit to download.



(/wiki/File:ESP32-S3-Pico_05.jpg)

- After running the installation package, the rest can be installed by default, but here for the subsequent experience, it is recommended to check boxes 1, 2, and 3



(/wiki/File:ESP32-S3-Pico_06.jpg)

- After the first and second items are enabled, you can open VSCode directly by right-clicking files or directories, which can improve the subsequent user experience.
- After the third item is enabled, you can select VSCode directly when you choose how to open it.

The environment setup is conducted on the Windows 10 system. Linux and Mac users can refer to the ESP-IDF Environment Setup (https://docs.espressif.com/projects/esp-idf/zh_CN/v5.1.4/esp32s3/get-started/windows-setup.html)

Install Espressif IDF Plugin

- The "**online installation**" is generally recommended for installing the Espressif IDF plugin. If online installation is not possible due to network issues, then "**offline installation**" can be used.
- For the tutorial on installing the Espressif IDF plugin, please refer to: Install Espressif IDF Plugin Tutorial (https://www.waveshare.com/wiki/Install_Espressif_IDF_Plugin_Tutorial)

Run the First ESP-IDF Demo

If you are just getting started with ESP32 and ESP-IDF, and you don't know how to create, compile, flash, and run ESP-IDF ESP32 programs, then please expand and take a look. Hope it can help you!

[\[Expand\]](#)

Demo

- ESP32-S3-GEEK Demo

Demo	Basic Description
Wireless_USB_flash_drive	The ESP32-S3-GEEK can be used as a USB disk with wireless access capabilities

Wireless_USB_flash_drive

Program Description

[\[Collapse\]](#)

- This routine allows the ESP32-S3-GEEK to be used as a USB disk with wireless access capabilities. When combined with SD card storage, it can become a high-capacity wireless storage device. Additionally, you can connect to the ESP32-S3-GEEK's hotspot to upload and download via an HTTP file server, greatly facilitating user operations. It is suitable for learning the USB MSC functionality of the ESP32, as it can act as a storage device connected to a USB host, supporting various storage methods, with callback functions to handle various events, and testing for stability and reliability
- To exit the program, simply press the boot button and power cycle the device

Hardware connection

- Insert the TF card into the development board
- Connect the development board to the computer

Code analysis

- **init_fat()**:Initializes the FAT file system. Depending on the configuration, it mounts either the internal flash memory or an external TF card
 - If using internal flash memory, a specific function is selected for mounting based on the ESP-IDF version
 - If using an external TF card, it is initialized and mounted according to the interface type

(SPI or SDIO)

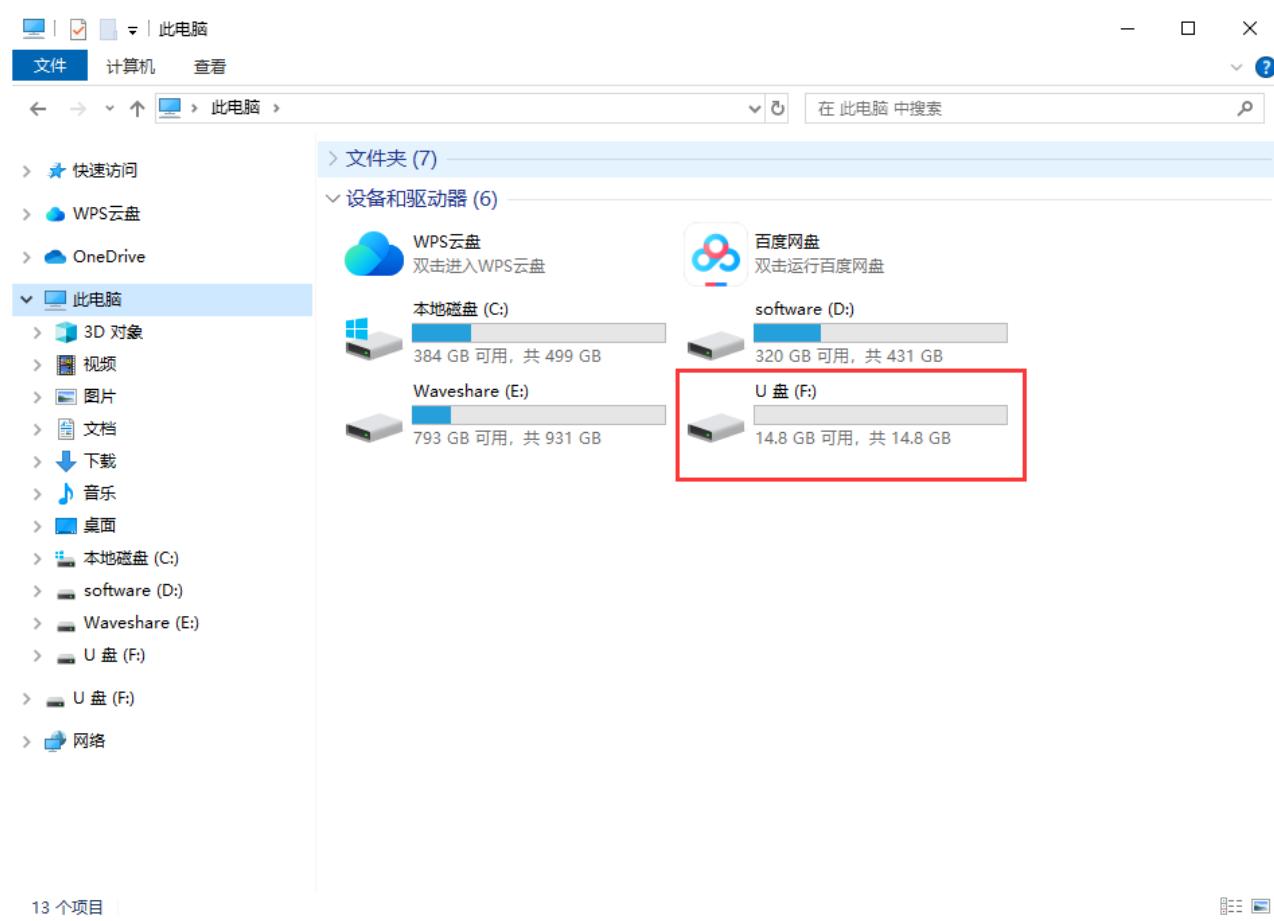
- **app_main ()**: The main function that coordinates the initialization of various parts of the program, including file storage, Wi-Fi (if configured), and USB MSC
 - Calls `init_fat` to initialize file storage
 - If Wi-Fi is configured, it initializes and starts the file server
 - Initializes tinyUSB and installs the driver

Code flashing

- Ensure the TF card is installed
- Select the model (esp32s3) and port
- Download via UART
- Flash the program

Result demonstration

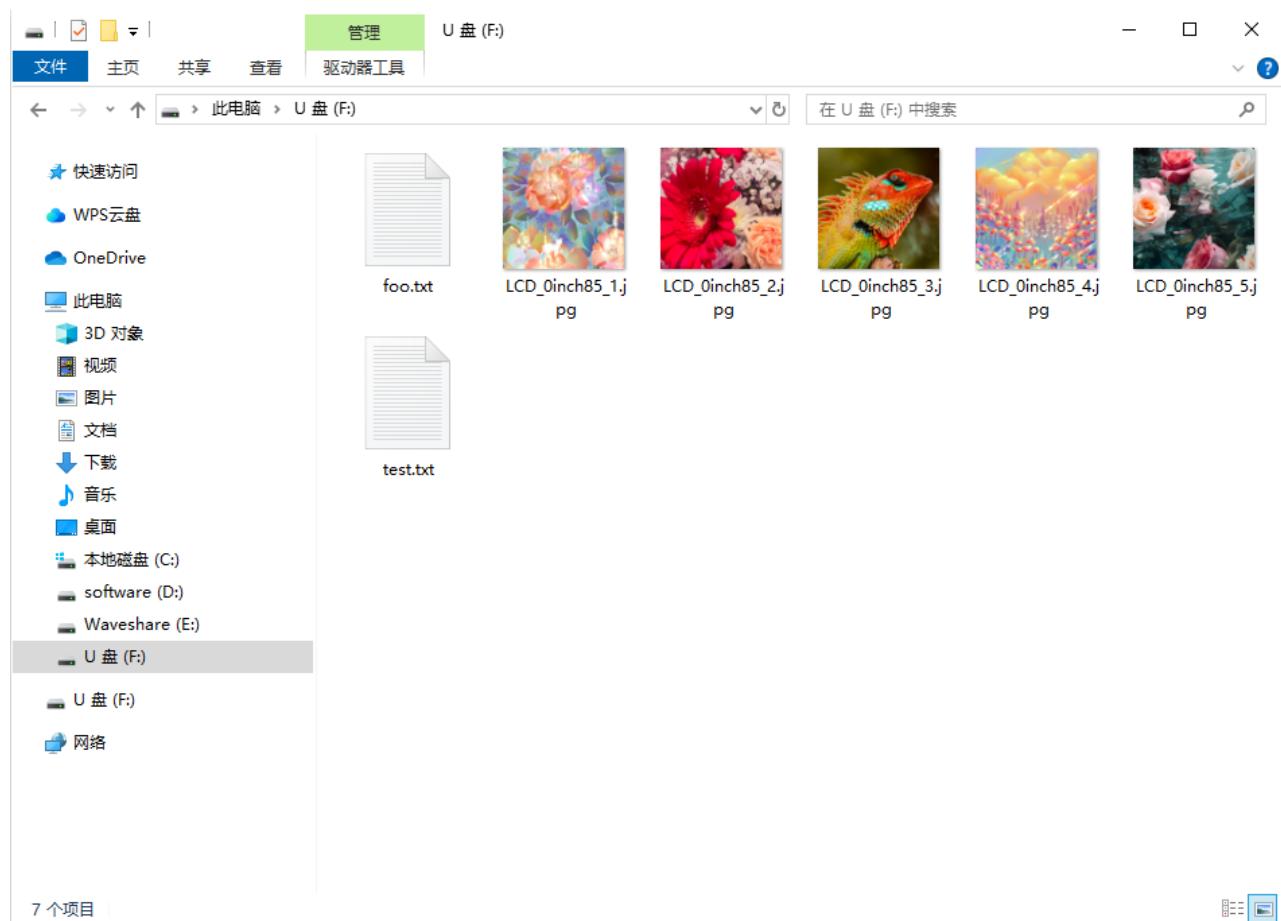
- After programming, unplug and plug the ESP32-S3-GEEK again, and you can see a new USB flash driver



(/wiki/File:ESP32-S3_wireless_disk02.png)

- You can open the USB drive to browse through the files on the TF card and perform various

operations such as adding, deleting, modifying, and checking files



(/wiki/File:ESP32-S3_wireless_disk03.png)

- Enter the PC's WIFI setting, connect to the ESP32-S3-GEEK's AP, and input the password of "Waveshare"
- After connecting successfully, open the browser, and log in to IP: 192.168.4.1



File:ESP32-S3-GEEK_Demo03.png)

- After login successfully, you can upload and download files wirelessly

Name	Type	Size (Bytes)	Delete
System Volume Information	directory	0	<button>Delete</button>
A.mp3	file	12039346	<button>Delete</button>
B.wav	file	1691180	<button>Delete</button>
Waveshare Test1.mp3	file	8997828	<button>Delete</button>
Waveshare Test2.mp3	file	9602739	<button>Delete</button>
test.txt	file	1048576	<button>Delete</button>
foo.txt	file	13	<button>Delete</button>
hello.txt	file	0	<button>Delete</button>
A	directory	0	<button>Delete</button>
1.jpg	file	10166	<button>Delete</button>
2.jpg	file	21144	<button>Delete</button>
3.jpg	file	8635	<button>Delete</button>
4.jpg	file	10000	<button>Delete</button>
5.jpg	file	23494	<button>Delete</button>
6.jpg	file	8281	<button>Delete</button>
7.jpg	file	11823	<button>Delete</button>
8.jpg	file	8775	<button>Delete</button>

(/wiki/File:ESP32-S3-GEEK-demo-10.png)

Cat-1 Module Command Set

HTTP

AT Command	Command Description	Return
AT+HTTPINIT	Open HTTP service	OK
AT+HTTPPARA="URL",https://www.waveshare.cloud/api/sample-test/ (https://www.waveshare.cloud/api/sample-test/)	Connect to the remote server	OK
AT+HTTPDATA=5,1000	Input the data	DOWNLOAD <Enter hello OK
AT+HTTPACTION=0	Open HTTP request, 0:GET; 1:POST; 2:HEAD; 3:DELETE; 4:PUT	OK +HTTPACTION: 0,200,54
AT+HTTPTERM	Close HTTP service	OK
AT+HTTPPARA	Set HTTP parameters	OK
AT+HTTPHEAD	Read the HTTP response header message	OK
AT+HTTPREAD	Read the HTTP response message	OK

SSCOM V5.13.1 Serial/Net data debugger, Author:Tintin, 2618058@qq.com

PORT COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛

```
[10:52:39.987]OUT→◇AT+HTTPINIT
□
[10:52:39.988]IN←◆AT+HTTPINIT
OK

[10:52:42.827]OUT→◇AT+HTTPPARA="URL", https://www.waveshare.cloud/api/sample-test/
□
[10:52:42.829]IN←◆AT+HTTPPARA="URL", https://www.waveshare.cloud/api/sample-test/
OK

[10:52:43.281]IN←◆
PB DONE

[10:52:44.228]OUT→◇AT+HTTPACTION=0
□
[10:52:44.229]IN←◆AT+HTTPACTION=0
OK

[10:52:44.661]IN←◆
+HTTPACTION: 0, 200, 57

[10:52:46.688]OUT→◇AT+HTTPREAD=0, 500
□
[10:52:46.690]IN←◆AT+HTTPREAD=0, 500
OK

+HTTPREAD: 57
{"value": "SIM7670G", "timestamp": "2023-11-23 10:34:19"}
+HTTPREAD: 0

[10:52:50.338]OUT→◇AT+HTTPDATA=5, 1000
□
[10:52:50.341]IN←◆AT+HTTPDATA=5, 1000
DOWNLOAD

[10:52:51.211]OUT→◇hello
□
[10:52:51.303]IN←◆
OK

[10:52:52.874]OUT→◇AT+HTTPACTION=1
□
[10:52:52.877]IN←◆AT+HTTPACTION=1
OK

[10:52:53.139]IN←◆
+HTTPACTION: 1, 200, 29

[10:52:55.396]OUT→◇AT+HTTPREAD=0, 500
□
[10:52:55.398]IN←◆AT+HTTPREAD=0, 500
OK

+HTTPREAD: 29
{"message": "data is saved!"}
+HTTPREAD: 0

[10:52:56.904]OUT→◇AT+HTTPACTION=0
□
[10:52:56.907]IN←◆AT+HTTPACTION=0
OK

[10:52:57.157]IN←◆
+HTTPACTION: 0, 200, 54

[10:52:58.198]OUT→◇AT+HTTPREAD=0, 500
□
[10:52:58.201]IN←◆AT+HTTPREAD=0, 500
OK

+HTTPREAD: 54
{"value": "hello", "timestamp": "2023-11-23 10:52:51"}
+HTTPREAD: 0

[10:52:59.864]OUT→◇AT+HTTPTERM
□
[10:52:59.868]IN←◆AT+HTTPTERM
OK
```

ClearData OpenFile SendFile Stop ClearSend OnTop English SaveConfig EXT

ComNum: COM95 Simcom HS-USB AT POR □ HEXShow SaveData □ ReceivedToFile □ SendHEX □ SendEvery: 1000 ms/Tim AddCrLf ?
 Show Time and Packe Overtime: 20 ms No: 1 BytesTo: 末尾 Verify: None
 RTS DTR BaudRate: 115200 A7670Sub



MQTT

AT Command	Command Description	Return
AT+CMQTTSTART	Open MQTT service	OK
AT+CMQTTACCQ=0,"Waveshare-7670X",0	Apply for MQTT client	OK
AT+CMQTTCONNECT=0,"tcp://mqtt.easyiothings.com",20,1	Send MQTT request, connect to the private MQTT server (MQTTS)	OK
AT+CMQTTTOPIC=0,8	Input the message to publish the topic	>A7670Pub OK
AT+CMQTPPAYLOAD=0,9	Input the message to be published	OK >waveshare
AT+CMQTPUB=0,0,60	Publish the message	OK +CMQTPUB: 0,0
AT+CMQTTSUB=0,8,1	Subscribe to message topic	>A7670Sub OK +CMQTTSUBTOPIC: 0,0 [10:03:39.665]Receive←◆ +CMQTRXSTART: 0,8,15 +CMQTRXTOPIC: 0,8 A7670Sub +CMQTRXPAYLOAD: 0,15 {"data":"test"} +CMQTRXEND: 0
AT+CMQTTSTOP	Stop MQTT service	OK
AT+CMQTTREL	Release the client	OK
AT+CMQTTUNSUBTOPIC	Unsubscribe the topic	OK
AT+CMQTTUNSUB	Release subscription	OK

SSCOM V5.13.1 Serial/Net data debugger, Author:Tintin,2618058@qq.com

POR COM_Settings Display Send_Data Multi_Strings Tools Help 联系作者 大虾论坛

```
[10:54:45.978]IN<-◆AT+CMQTTACQ=0, "Waveshare-7670X", 0
OK

[10:54:47.051]OUT->◇AT+CMQTTCONNECT=0, "tcp://mqtt.easyiothings.com", 20, 1
>
[10:54:47.053]IN<-◆AT+CMQTTCONNECT=0, "tcp://mqtt.easyiothings.com", 20, 1
[10:54:47.249]IN<-◆
OK

*CMQTTCONNECT: 0, 0

[10:54:48.036]OUT->◇AT+CMQTTTOPIC=0, 8
>
[10:54:48.039]IN<-◆AT+CMQTTTOPIC=0, 8
>
[10:54:49.119]OUT->◇A7670Pub
>
[10:54:49.122]IN<-◆
OK

[10:54:49.993]OUT->◇AT+CMQTTPAYLOAD=0, 9
>
[10:54:49.996]IN<-◆AT+CMQTTPAYLOAD=0, 9
>
[10:54:50.533]OUT->◇waveshare
>
[10:54:50.536]IN<-◆
OK

[10:54:51.334]OUT->◇AT+CMQTTPUB=0, 0, 60
>
[10:54:51.337]IN<-◆AT+CMQTTPUB=0, 0, 60
OK

*CMQTTPUB: 0, 0

[10:54:53.884]OUT->◇AT+CMQTTSUB=0, 8, 1
>
[10:54:53.887]IN<-◆AT+CMQTTSUB=0, 8, 1
>
[10:54:54.474]OUT->◇A7670Sub
>
[10:54:54.478]IN<-◆
OK

[10:54:54.539]IN<-◆
*CMQTTSUB: 0, 0

*CMQTTRXSTART: 0, 8, 15
*CMQTTRXTOPIC: 0, 8
A7670Sub
*CMQTTRXPAYLOAD: 0, 15
{"data": "test"}
*CMQTTRXEND: 0

[10:54:56.450]IN<-◆
*CMQTTRXSTART: 0, 8, 15
*CMQTTRXTOPIC: 0, 8
A7670Sub
*CMQTTRXPAYLOAD: 0, 15
{"data": "test"}
*CMQTTRXEND: 0

[10:54:59.385]OUT->◇AT+CMQTTUNSUBTOPIC=0, 8
>
[10:54:59.389]IN<-◆AT+CMQTTUNSUBTOPIC=0, 8
>
[10:55:00.135]OUT->◇A7670Sub
>
[10:55:00.139]IN<-◆
OK

[10:55:01.102]OUT->◇AT+CMQTTUNSUB=0, 1
>
[10:55:01.105]IN<-◆AT+CMQTTUNSUB=0, 1
OK

[10:55:01.220]IN<-◆
*CMQTTUNSUB: 0, 0

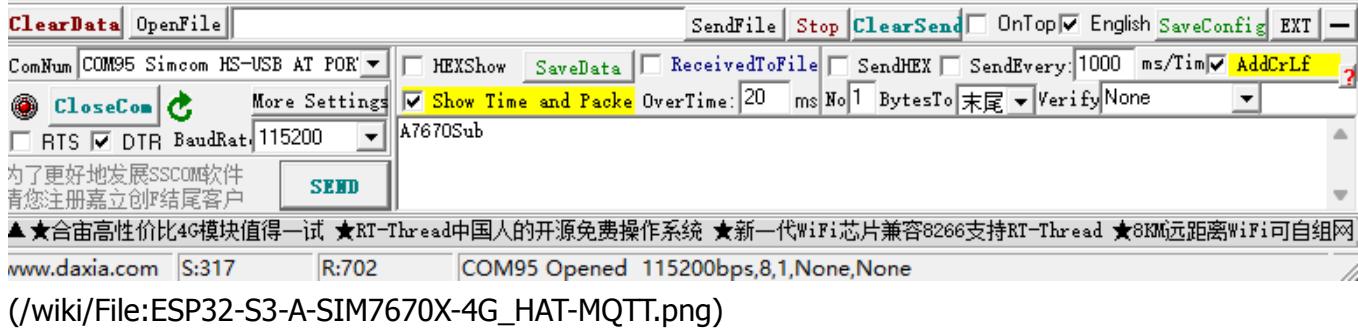
[10:55:01.855]OUT->◇AT+CMQTTDISC=0, 120
>
[10:55:01.858]IN<-◆AT+CMQTTDISC=0, 120
[10:55:01.939]IN<-◆
OK

*CMQTTDISC: 0, 0
```

```
[10:55:02.428]OUT→◇AT+CMQTTREL=0
[10:55:02.437]IN←◆AT+CMQTTREL=0
OK

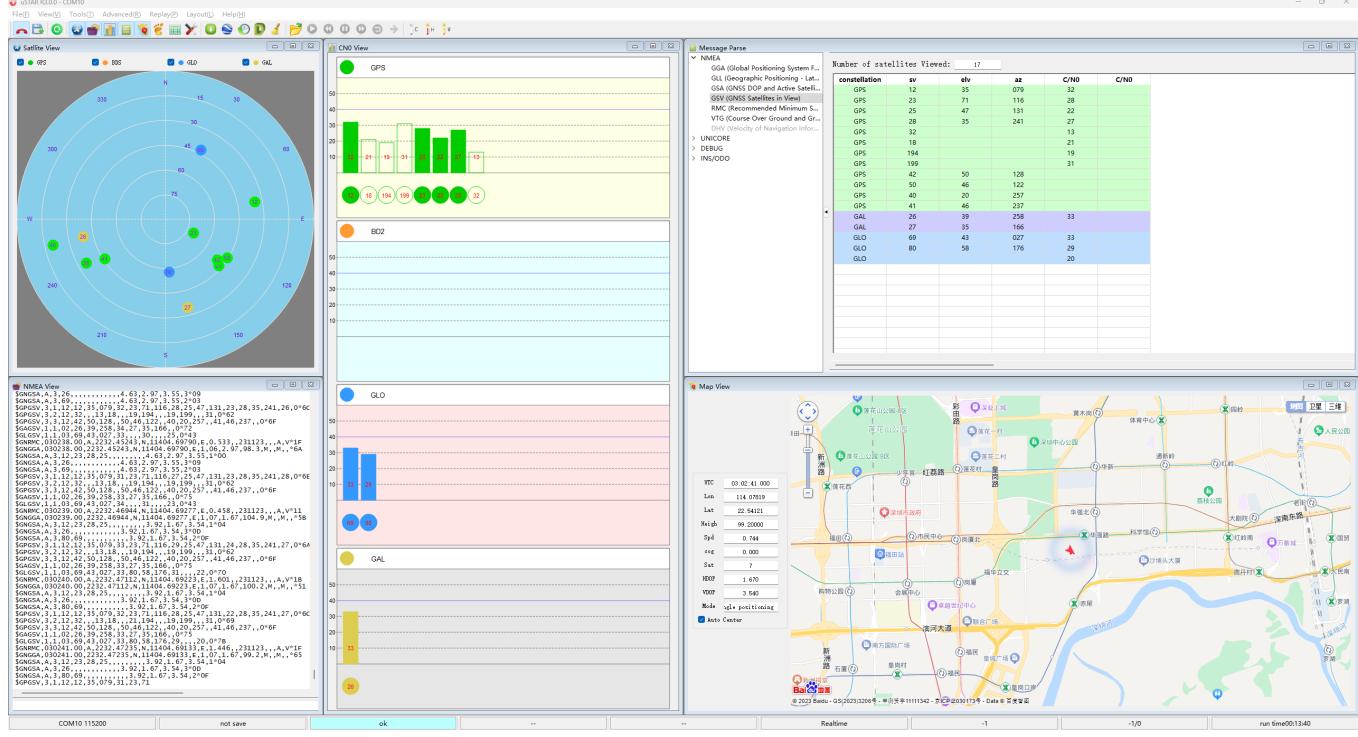
[10:55:02.994]OUT→◇AT+CMQTTSTOP
[10:55:02.998]IN←◆AT+CMQTTSTOP
OK

+CMQTTSTOP: 0
```



GNSS

AT+CGNNSPWR=1	Open GNSS	+CGNNSPWR: READY!
AT+CGNSSTST=1	Open GNSS data output	OK



(/wiki/File:ESP32-S3-A-SIM7670X-4G_HAT-GNSS.png)

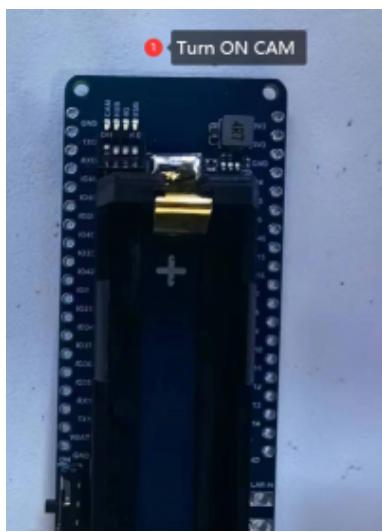
Demo Explanation

ESP32-S3 Application

Camera

This demo is based on the CameraWebServer demo of the ESP32.

- Firstly, you need to set the WiFi name and password and switch the hardware to ESP32S3 by default.
- Please turn on the CAM of the DIP switch on the back of the development board, and connect to the supported cameras.
- Please check CameraPins.



(/wiki/File:Esp32-s3-a-sim7670x_240124_01.png)

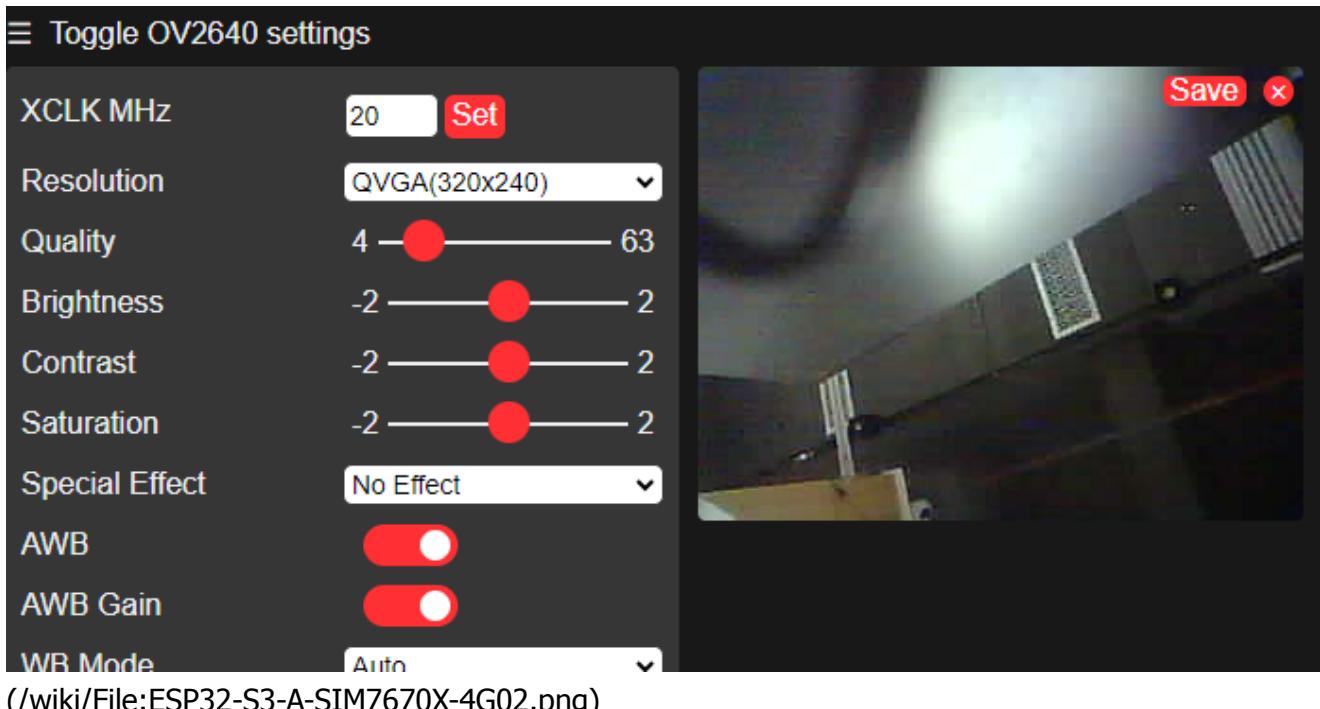


Y2	GPIO 7
Y3	GPIO 8
Y4	GPIO 9
Y5	GPIO 10
Y6	GPIO 11
Y7	GPIO 12
Y8	GPIO 13
Y9	GPIO 14
PCLK	GPIO 57
VSYNC	GPIO 36
HREF	GPIO 35
XCLK	GPIO 34
SIO_CLK	GPIO 16
SIO_DAT	GPIO 15
RESET	CHIP_PU

(/wiki/File:Esp32-s3-a-sim7670x_240124_02.png)

```
#define PWDN_GPIO_NUM      -1
#define RESET_GPIO_NUM     -1
#define XCLK_GPIO_NUM       34
#define SIOD_GPIO_NUM      15
#define SIOC_GPIO_NUM       16
#define Y9_GPIO_NUM         14
#define Y8_GPIO_NUM         13
#define Y7_GPIO_NUM         12
#define Y6_GPIO_NUM         11
#define Y5_GPIO_NUM         10
#define Y4_GPIO_NUM         9
#define Y3_GPIO_NUM         8
#define Y2_GPIO_NUM         7
#define VSYNC_GPIO_NUM      36
#define HREF_GPIO_NUM       35
#define PCLK_GPIO_NUM       37
```

- Program the demo, and open the terminal to access the prompted IP.



TF-Card

- Insert the TF-Card into the TF card slot.
- Pinout definition:

```
const int SDMMC_C  
LK = 5;  
const int SDMMC_C  
MD = 4;  
const int SDMMC_D  
ATA = 6;  
const int SD_CD_P  
IN = 46;
```



(/wiki/File:Esp32-s3-a-sim7670x_240124_03.jpg)

- Program the demo, and open the terminal to display the file content:

```
11:39:04.960 -> SD_MMC Card Type: SDHC  
11:39:04.960 -> SD_MMC Card Size: 59640MB  
11:39:04.960 -> Listing directory: /  
11:39:04.960 -> DIR : System Volume Information  
11:39:04.960 -> DIR : bat  
11:39:04.960 -> DIR : CameraWebServer  
11:39:04.960 -> DIR : RGB  
11:39:04.960 -> DIR : SD  
11:39:04.960 -> FILE: ut.h SIZE: 3953  
11:39:04.960 -> DIR : AP
```

(/wiki/File:ESP32-S3-A-SIM7670X-4G03.png)

RGB

Onboard a WS2812b RGB LED, and the signal pin is 38.

After programming the sample demo, the LED light is expected to display a gradient color.



wiki/File:ESP32-S3-A-SIM7670X-4G-5.gif)

BAT

This development board utilizes the MAX17048 as the battery charge measurement IC.

- First, confirm the I2C pin.

```
#include <Wire.h>

#define MAX17048_I2C_ADDRESS 0x36

void setup() {
    Wire.begin(3, 2);
    Serial.begin(9600);
}
```

(/wiki/File:ESP32-S3-A-SIM7670X-4G-08.png)

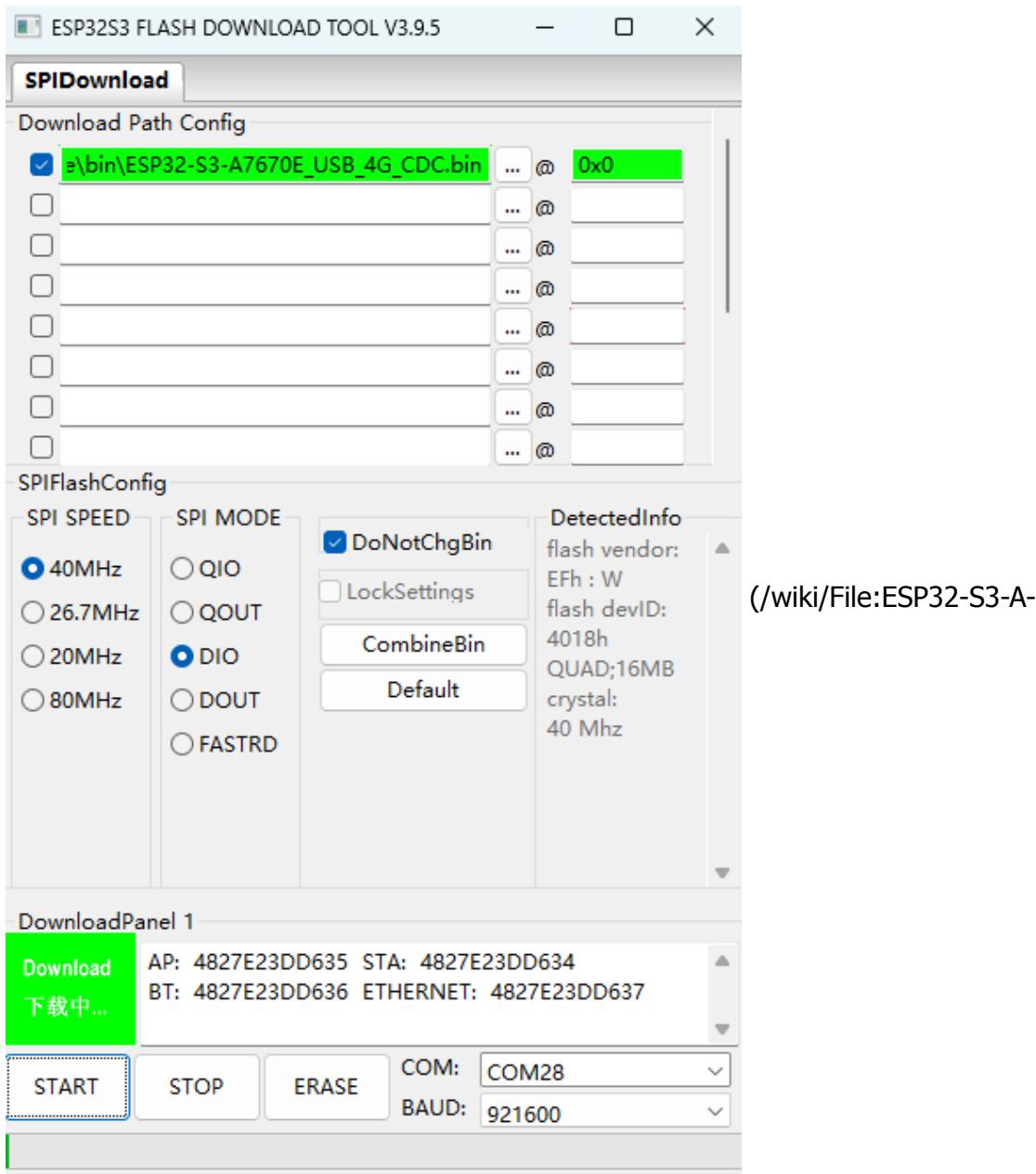
- Program the code and change the threshold.

```
11:55:10.316 -> Battery Level: 83.06%
11:55:11.282 -> Battery Level: 83.06%
11:55:12.322 -> Battery Level: 83.11%
11:55:13.303 -> Battery Level: 83.11%
11:55:14.324 -> Battery Level: 83.11%
11:55:15.289 -> Battery Level: 83.11%
11:55:16.318 -> Battery Level: 83.11%
11:55:17.332 -> Battery Level: 83.11%
11:55:18.327 -> Battery Level: 83.11%
11:55:19.325 -> Battery Level: 83.11%
```

(/wiki/File:ESP32-S3-A-SIM7670X-4G-09.png)

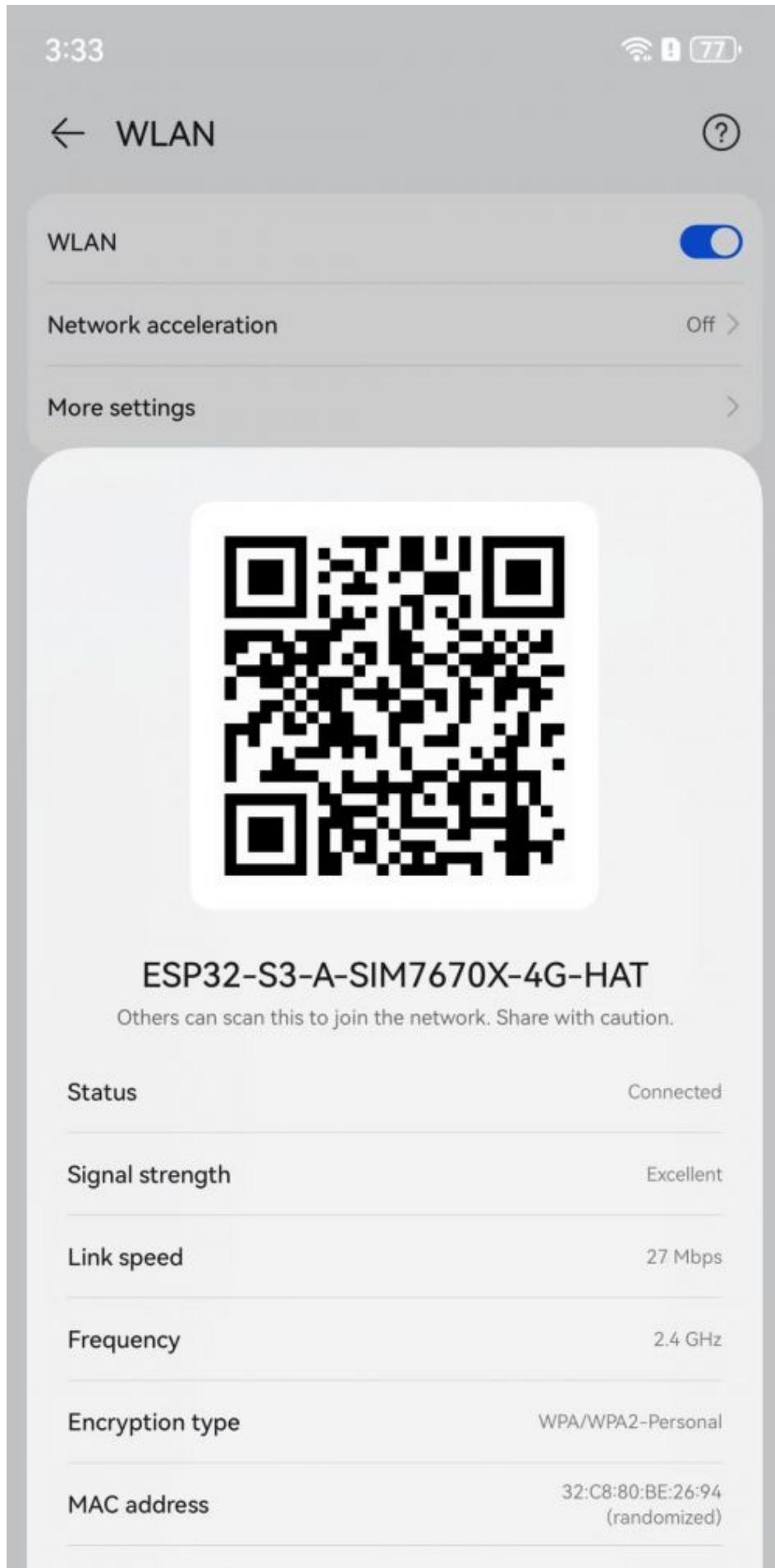
Portable WIFI Demo

- This demo uses the TinyUSB protocol stack to communicate with a 4G Cat-1 module, using ppp dial-up to provide the network to the ESP32-S3.
- This demo uses the compiled firmware, please download Flash Tools (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/flash_download_tool_3.9.5_0.zip) first.
- Download the firmware:
 - [ESP32-S3-A7670E_USB_4G_CDC.zip](https://files.waveshare.com/wiki/ESP32-S3-A7670E_USB_4G_CDC.zip) (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/ESP32-S3-A7670E_USB_4G_CDC.bin)
 - [ESP32-S3-SIM7670G_USB_4G_CDC.zip](https://files.waveshare.com/wiki/ESP32-S3-SIM7670G_USB_4G_CDC.zip) (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/ESP32-S3-SIM7670G_USB_4G_CDC.bin)
- Open the **Flash Tools**, select the development mode, choose the firmware, with the address set to "0x0", as shown in the diagram, and insert the SIM card to download and start the program.



SIM7670X-4G_HAT_wIFI.png)

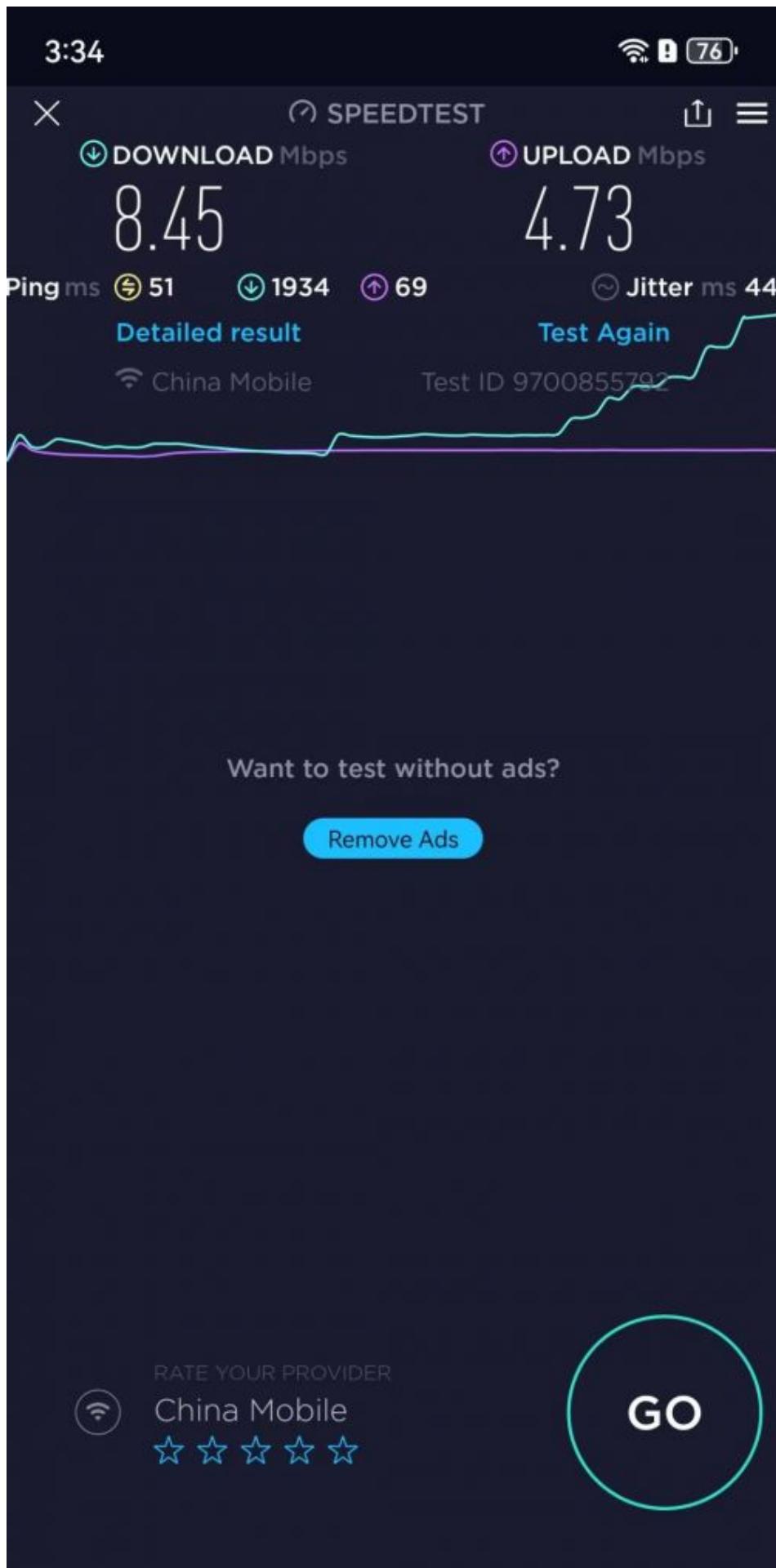
- The back of the development board toggle switch 4G on, USB off, re-power on the development board, wait for the LED display red, open the phone to connect to WIFI: ESP32-S3-A-SIM7670X-4G-HAT, password: 12345678 to access the Internet.



(/wiki/File:ESP32-S3-A-



SIM7670X-4G_HAT_wIfl-3.jpg)



(/wiki/File:ESP32-S3-A-



SIM7670X-4G_HAT_wIfI04.jpg)



(/wiki/File:Esp32-s3-a7670e-wifi-comanded.png)

- For more details, you can refer to sample demo (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/ESP32-S3-A-SIM7670X_4G.zip).

Waveshare Cloud Application

Please download the demo (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/ESP32-S3-A-SIM7670X_4G.zip), and open the **GNSS-With-WaveshareCloud** sample demo.

In this application, communication between ESP32-S3 and A7670E-FASE is established using ESP32-S3's software serial port. By sending AT commands, the GNSS (Global Navigation Satellite System) is activated, and NMEA GNSS data is parsed and uploaded to the Waveshare Cloud. The specific location of the development board is then displayed on a web view map page.

Here, we take the map service provided by Waveshare Cloud (<https://waveshare.cloud/#/yourlocation>) as an example:

1. Through **Device | Attribute** to create any devices, and obtain the MQTT connection data.

id	devicesType	Device name	大屏路由器地址	Last up time	Operation		
1	Modbus	Modbus POE Relay	2023-12-08 14:31:23				
2	NB-IoT	SIM7028 NB-IoT AT	sim7028	2023-12-08 17:23:10			
3	ESP32-4G	ESP32-S3-A-SIM70X-4G		2023-12-13 16:40:26			
4	Raspberry Pi	Raspberry Pi 5		2023-12-06 19:52:56			
5	Relay	ESP32-S3-Relay H					
6	ESP32-4G	ESP32		2023-12-08 17:28:26			
7	GPS	GPS_test		2023-12-13 16:41:26			

Total 7 20/page < 1 > Go to 1

(/wiki/File:A7670E_Cat-1-GNSS-HAT_faq.png)

2. Enter the parameters in the **GNSS-With-WaveshareCloud** demo.

```
#define MSG_BUFFER_SIZE (50)
#define STASSID "yourWifi ssid"
#define STAPSK "yourwifi passwd"

#define MAX17048_I2C_ADDRESS 0x36
const char *clientID = ""; // Client ID
char sub[] = ""; // Sub Topic
char pub[] = ""; // Pub Topic
const char *mqtt_server = "mqtt.waveshare.cloud";
```

File:GNSS_with_wavesharecloud.png



(/wiki/File:Gnss_esp32-s3-a7670E-waveshareCloud.jpg)

Resource

Document

- Schematic (<https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/ESP32-S3-A-SIM7670X-4G-Sch.pdf>)
- MicroPython Development Document (<https://docs.micropython.org/en/latest/>)
- ESP32 Arduino Core's documentation (<https://docs.espressif.com/projects/arduino-esp32/en/latest/index.html>)
- Arduino-Esp32 (<https://github.com/espressif/arduino-esp32>)
- ESP-IDF (<https://github.com/espressif/esp-idf>)
- 3D drawing (<https://files.waveshare.com/wiki/ESP32-S3-A-SIM7670X-4G-HAT/3D/esp32-s3-a-sim7670x-4g-pcba.zip>)

Demo

- Demo_V1 (https://files.waveshare.com/wiki/ESP32-S3-A-SIM7670X-4G-HAT/Demo/ESP32-S3-A-SIM7670X-4G-HAT-Demo_V1)

7670X_4G.zip)

- Demo_V2 (<https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/demo/ESP32-S3-A-SIM7670X-4G-V2.zip>)

Tools & Driver

- SSCom tool (https://files.waveshare.com/wiki/ESP32-S3-A7670E-4G/Sscom_7670X.zip)
- Thonny Python IDE (<https://thonny.org/>)
- Arduino IDE (<https://www.arduino.cc/en/software/>)
- GPS debug tool (<https://files.waveshare.com/wiki/A7670E-Cat-1-GNSS-HAT/uSTAR3.zip>)
- A7600X Windows Driver (<https://files.waveshare.com/upload/5/59/A7600X-Windows-Driver.7z>)

Application Note

- ESP32-S3 Datasheet (https://www.espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=842)
- SIM7672X SIM7652X Series FTP(S) Application Note V1.00 ([https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_FTP\(S\)_Application_Note_V1.00.pdf](https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_FTP(S)_Application_Note_V1.00.pdf))
- SIM7672X SIM7652X Series HTTP(S) Application Note V1.00 ([https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_HTTP\(S\)_Application_Note_V1.00.pdf](https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_HTTP(S)_Application_Note_V1.00.pdf))
- SIM7672X SIM7652X Series MQTT(S) Application Note V1.00 ([https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_MQTT\(S\)_Application_Note_V1.00.pdf](https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_MQTT(S)_Application_Note_V1.00.pdf))
- SIM7672X SIM7652X Series NETWORK Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_NETWORK_Application_Note_V1.00.pdf)
- SIM7672X SIM7652X Series SSL Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_SSL_Application_Note_V1.00.pdf)
- SIM7672X SIM7652X Series Sleep Mode Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_Sleep_Mode_Application_Note_V1.00.pdf)
- SIM7672X SIM7652X Series TCPIP Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_TCPIP_Application_Note_V1.00.pdf)
- SIM7672X SIM7652X Series UART Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_UART_Application_Note_V1.00.pdf)
- SIM7672X SIM7652X Series UIM HOT SWAP Application Note V1.00 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_SIM7652X_Series_UIM_HOT_SWAP_Application_Note_V1.00.pdf)
- SIM7672X Series Hardware Design V1.01 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672X_Series_Hardware_Design_V1.01.pdf)
- SIM7672 Series SPEC Preliminary 230518 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672_Series_SPEC_Preliminary_230518.pdf)
- SIM7672 Series Spec 231205 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM7672_Series_Spec_231205.pdf)

- SIM767XX Series AT Command Manual V1.01 (https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM767XX_Series_AT_Command_Manual_V1.01.pdf)
- SIM767XX V1.01 KDL(230309) ([https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM767XX_V1.01_KDL\(230309\).pdf](https://files.waveshare.com/wiki/ESP32-S3-SIM7670G-4G/SIM767XX_V1.01_KDL(230309).pdf))

FAQ

Question: What are the functions of each dip switch?

Answer:

The functions of each dip switch are as follows:

Function	Description	Notes
CAM	Control the camera on and off function	Set to OFF to turn off the camera
HUB	Control the USB HUB drive circuit power supply	Use battery, set to OFF to enable low power mode
4G	Control the A7670E module power supply	Set to OFF to shut down the cellular module
USB	Control the A7670E module USB channel selection	Set to OFF, the module cannot be accessed through USB, and the hotspot function is turned on

(/wiki/File:Esp32-dip.png)

Question: How does ESP32-S3 dial-up internet using the 4G module's USB in the example?

Answer:

ESP32-S3 can perform PPP dial-up internet using both serial port and USB. In this example, TinyUSB protocol is utilized, and USB enumeration addresses are used for dial-up internet.

Question: How can the ESP32-S3 dial-up internet code be modified?

Answer:

The code in this example is compiled and flashed using esp-idf. If using Arduino IDE, porting of

TinyUSB and handling of PPP packets will be necessary.

Question:How to modify the ESP32-S3 dial-up code?

Answer:

The code in this example is compiled and flashed by esp-idf. To modify ESP32-S3 dial-up code, Arduino IDE needs to be ported tinyUSB and ppp packet processing, etc.

Question:Can you assist with code modification?

Answer:

We do not assist in modifying the code, please do it yourself.

Question: Can the module itself connect to WaveshareCloud and report latitude and longitude data?

Answer:

Currently, the development board connects to the A/SIM7670X 4G module's serial port using software serial port. After enabling GNSS functionality with AT commands, the module outputs satellite signal data through its serial port. AT command can also be sent to the serial port at the moment. Execute the Publish command to publish the required data and filtering of NMEA signal data is required for listening to the returned values from the platform.

Question:Does the camera included in the package support auto-focus?

Answer:

No.

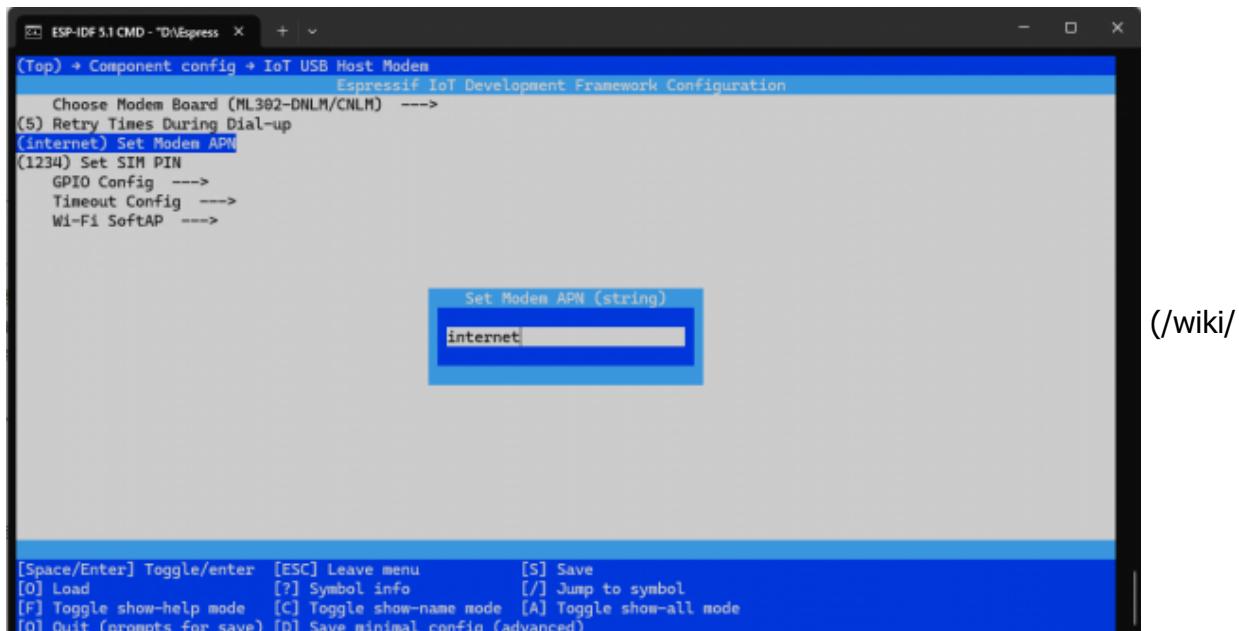
Question:For the portable WiFi demo, how to set module APN manually by selecting different operators?

Answer:

The default setting is null for the portable WiFi demo, if the module can't recognize the SIM card APN operator automatically, you need to modify the provided source code, refer to README.md in the directory of ESP32-S3-A-SIM7670X-4G-example.

The specific steps are as follows:

1. Refer to the ESP-IDF chapter of the development environment configuration, install the ESP-IDF development environment and vscode programming tools.
2. Use vscode to open the example program and enter menuconfig to set APN manually.



File:ESP32-S3-A7670E.png)

3. Upload the demo to the development board, power off, and reboot the development board.

Question:Why is it common to encounter situations where header files are missing when using the provided dmeos? Can library files be provided?

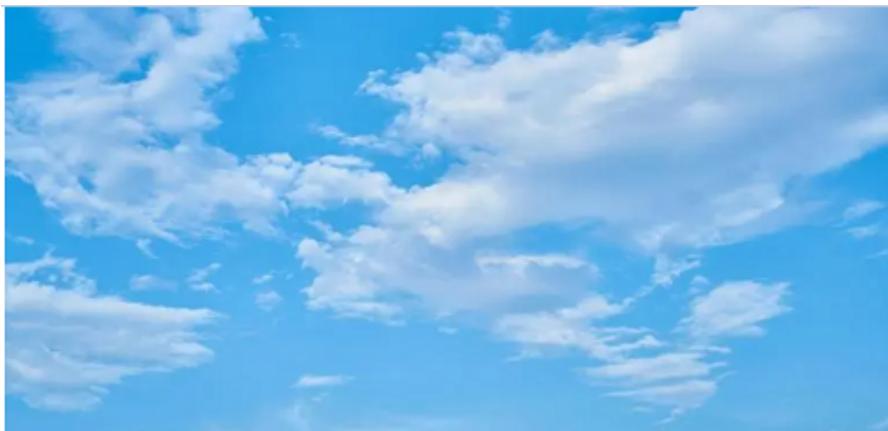
Answer:

All the example demos provided use libraries downloaded and installed from the Arduino IDE library. These library files are constantly being updated and iterated upon, so often a library might be missing and can be directly downloaded and installed in the Arduino IDE.

Question:What if I can't receive the GPS signal and don't get the location information?**Answer:**

Plug in the GPS antenna to the GNSS antenna socket and place the receiver label face down in an open outdoor area (note that it can't be tested in cloudy and rainy weather), and you need to wait for about 1 minute to receive the positioning signal when powering on;

As GPS indoor search is not stable, please put the module or antenna next to the balcony or window, or directly test it under the outdoor visible sky.



(/wiki/File:Visible-sky.png)

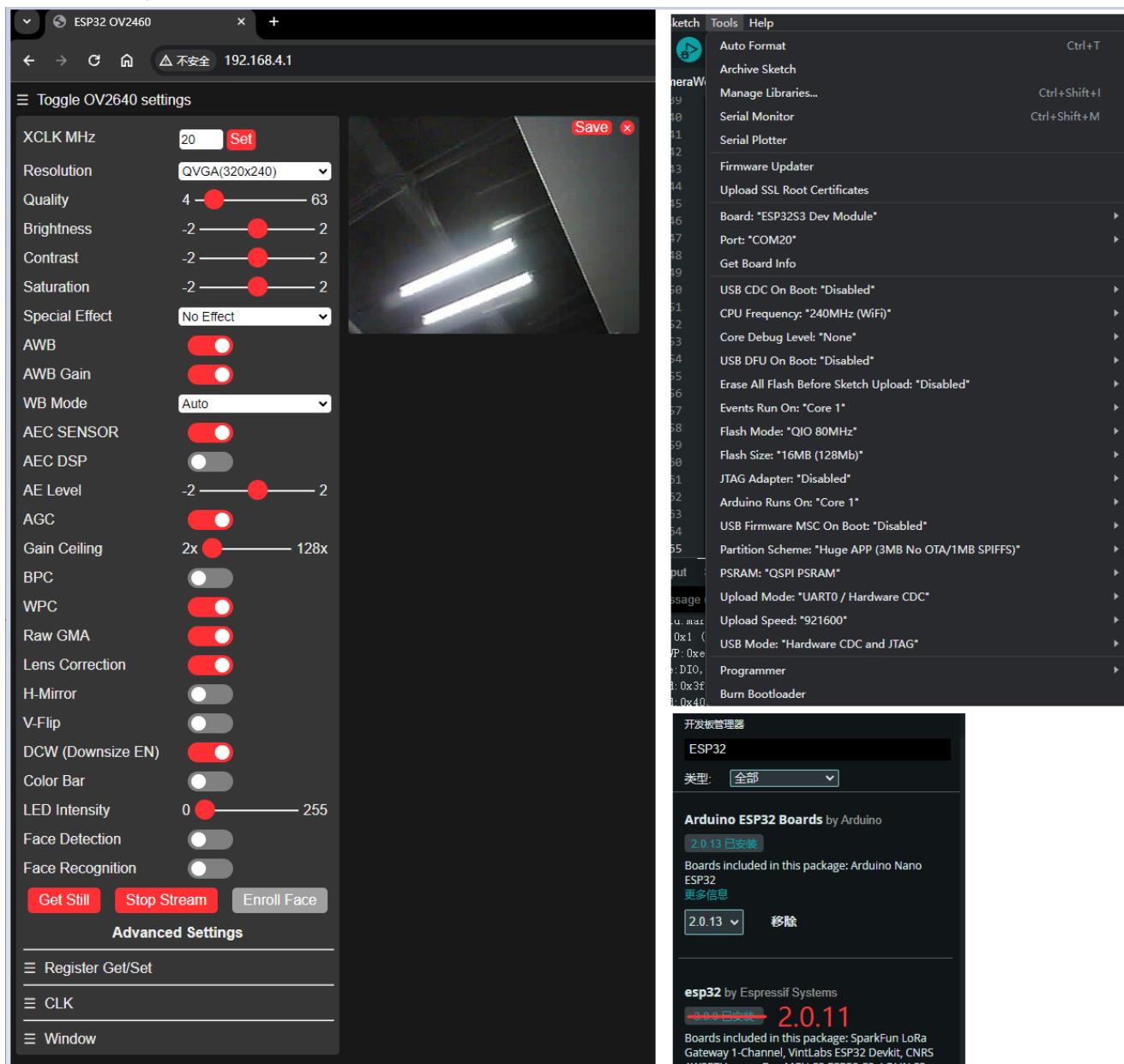
Question:Does this development board have any demos to remotely transmit the video from the camera over 4g?

Answer:

- This is not readily available sample demo, please program yourself and achieve it by secondary development.
- This Module only provides data connection, ESP32 capture camera side is a video stream, this video stream is temporarily displayed in html, that is, directly displayed on the web page!
- If the video stream data through the module can be mapped out to achieve the function of map transmission; the theory is no problem; need to build a public IP server and so on.

Question:Why is the camera screen black?**Answer:**

Please configure as shown below:



(/wiki/File:7670-CAM.png)

Question:Why does the corresponding hotspot not appear after flashing the AP sample program?

Answer:

To realize the hotspot function:

- 0) X7670X to register to the network and successfully dial up to the Internet
- 1) Turn on the 4G dip switch on the back of the board, turn off the USB, and re-power on the board.
- 2) Correctly download the corresponding firmware, don't confuse the A7670 and SIM7670 firmware, remember to check the box.

Question:Can the ESP32-S3-SIM7670G-4G be turned on with the battery when it is not connected to the power supply?

Answer:

- (1) the first time on the battery (that is, after the installation of the battery) need to be connected to the power supply to activate the protection mechanism (this mechanism is to prevent the reverse connection), the battery is fully charged, you can not need to connect the power supply!
- (2) In addition, it can also be discharged to activate, typec interface in addition to charging, but also for external equipment power supply, so that the module to external equipment power supply to achieve the purpose of discharging, but also activated!

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 PM GMT+8 (Monday to Friday)

Submit Now (<https://service.waveshare.com/>)

Retrieved from "<https://www.waveshare.com/w/index.php?title=ESP32-S3-SIM7670G-4G&oldid=109082>" (<https://www.waveshare.com/w/index.php?title=ESP32-S3-SIM7670G-4G&oldid=109082>)"
