# Arduino develops the general environment configuration

## ➢ Environment Setting

The software framework of ESP32 series development board is mature, and it can use C/C++(Arduino,ESP-IDF), MicroPython, etc., to rapidly develop product prototypes. The following is a brief introduction：

Espressif's official C/C++ library is easy and quick to install, please refer to the FAQ to solve download problems encountered by users in Chinese mainland✧ Arduino development manual for ESP32 series

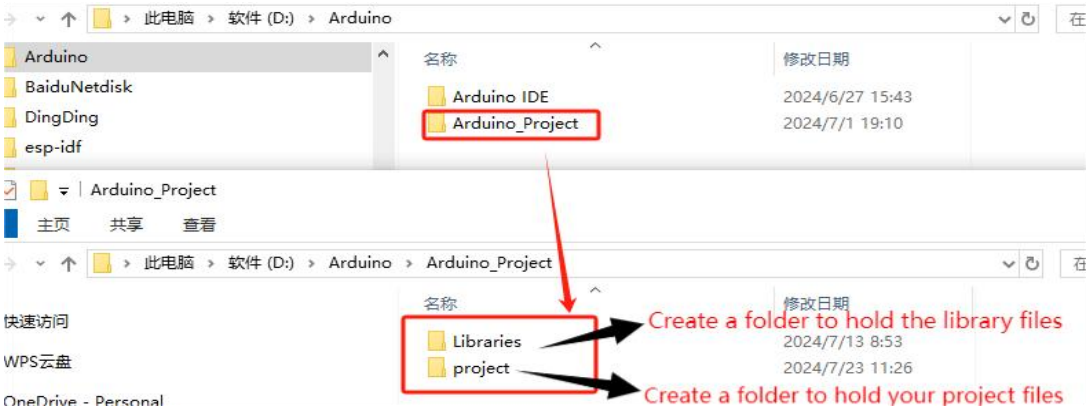
- ✧ ESP-IDF Development Manual for ESP32 series
- ✧ Arduino development manual for ESP32 series

- Environment Settings are carried out under the Windows 10 system, users can choose to use Arduino or Visual Studio Code(ESP-IDF) as an IDE for development, Mac/Linux operating system users please refer to the official instructions
- If you choose to use Visual Studio Code(ESP-IDF) as your IDE, please refer to Configuration Instructions
- Here we mainly explain the development with Arduino, including the installation of Arduino, installation of libraries and development boards, configuration of compilation environment,

compilation, upload, debugging and other aspects in detail. Just follow the steps

## ◆ Arduino

- Download and install Arduino IDE.
  - ■ Choose your own installation path. If I installed on disk D, I will create an “Arduino” folder under disk D, where I will install the Arduino IDE, and create an “Arduino_Project” folder under the Arduino folder to store the library and project code.




- Install ESP32 on the Arduino IDE as shown below, and you can refer to this link.

### ■ Once installed, proceed to the following steps

(1) Open the Arduino IDE.

(2) Click on the File menu on the top menu bar.

(3) Click on the Preferences menu item. This will open a Preferences dialog box.

(4) You should be on the Settings tab in the Preferences dialog box by default.

(5) Look for the textbox labeled “Additional Boards Manager URLs ”.

(6) If there is already text in this box add a coma at the end of it, then follow the next step.

(7) Paste the following link into the text box ：

Stable release link:

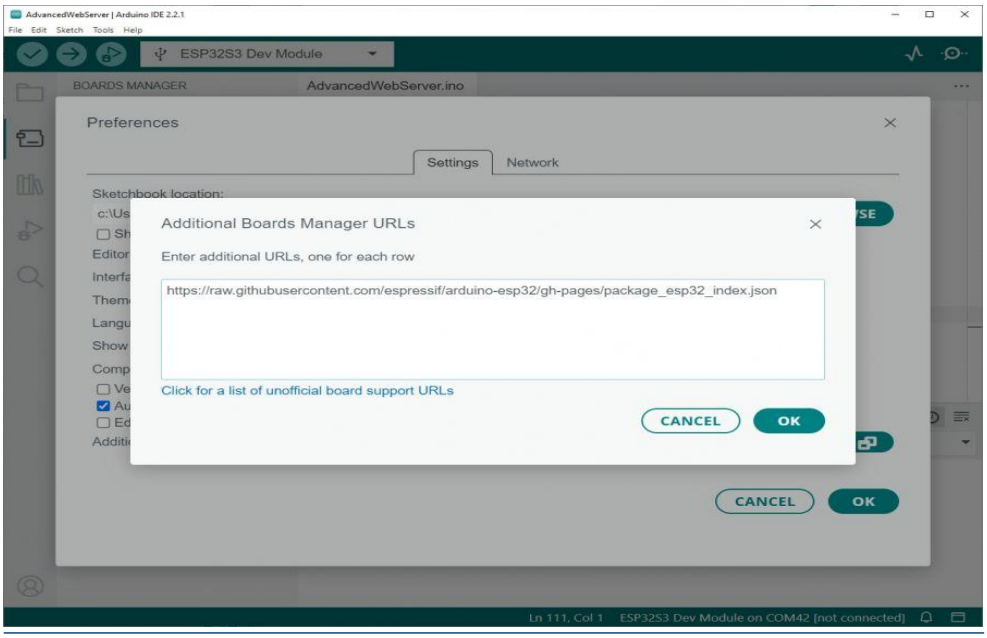
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Development release link:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

(8) Click the OK button to save the setting.

The textbox with the JSON linkin it is illustrated here:




Note: we should keep in mind that a development board should not make changes easily after determining the version of the library and development board, and hastily changing the version of the development board and library will lead to compilation errors and other problems

(9) In the Arduino I DE click on the Tools menu on the top menu bar.

(10) Scroll down to the Board: entry

(11) A sub menu will open when you highlight the Board: entry.

(12) At the top of the sub menu is Boards Manager. Click on it to open the Boards Manager dialog box.

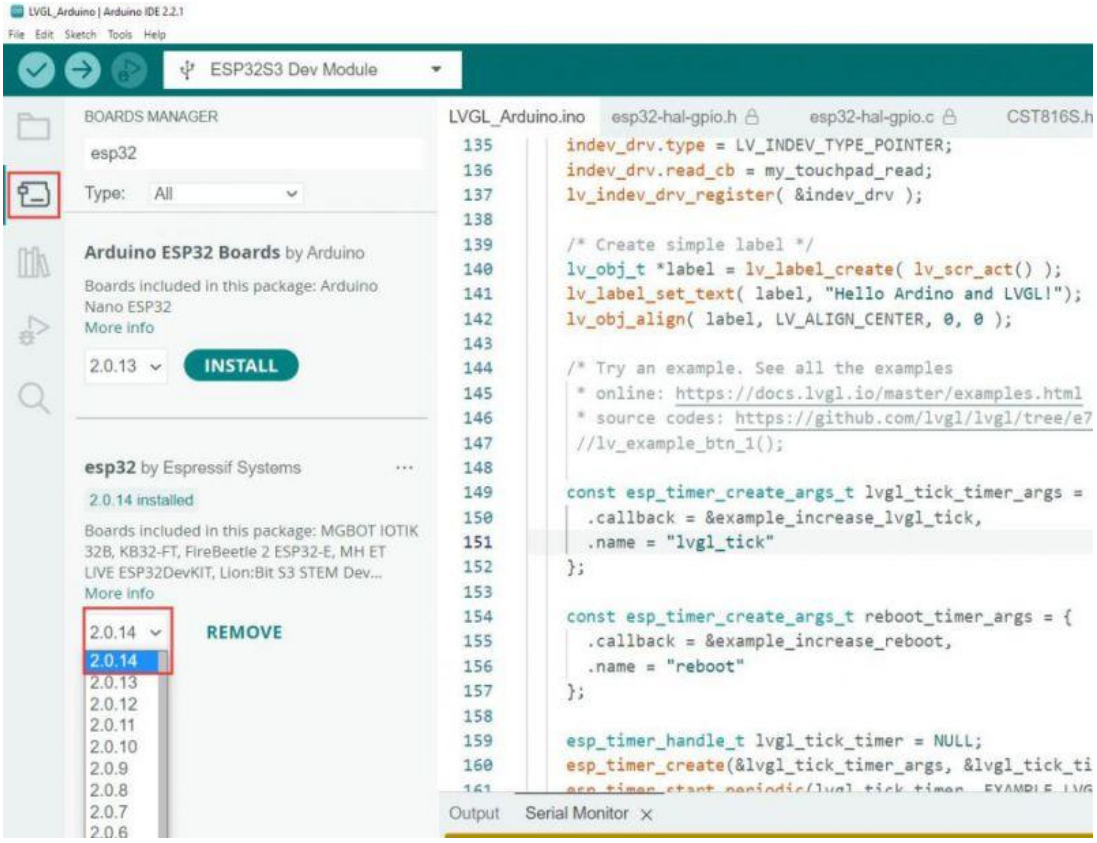
(13) In the search box in the Boards Manager enter “ESP32 ”.

(14) You should see an entry for “esp32 by Espressif Systems ”.

Highlight this entry and click on the Install button.

Note: The development board required for 1.3inch (1.28inch), 1.9inch, 2.1inch, 4.0inch, 4.3inch, 5.0inch in the current product is :esp32 2.0.17; For 7.0inch, you need an esp32 3.0.0 or higher. We use esp32 3.0.2, you can keep the same with us (if there is a change, we will even change it)

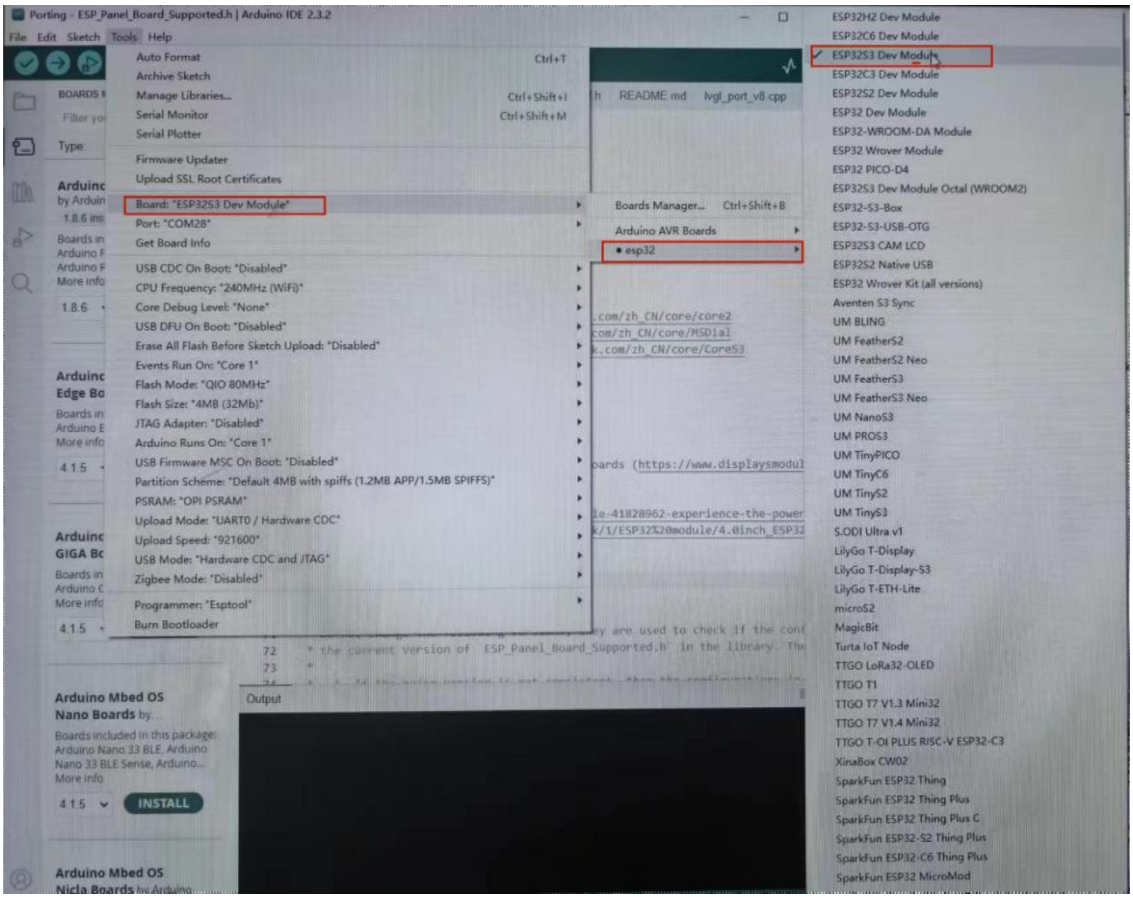
This will install the ESP32 boards into your Arduino IDE




(15) Click tools and then click “Board”

(16) Let's go to “esp32”

(17) Select the appropriate development board here select “ESP32S3 Dev Module”

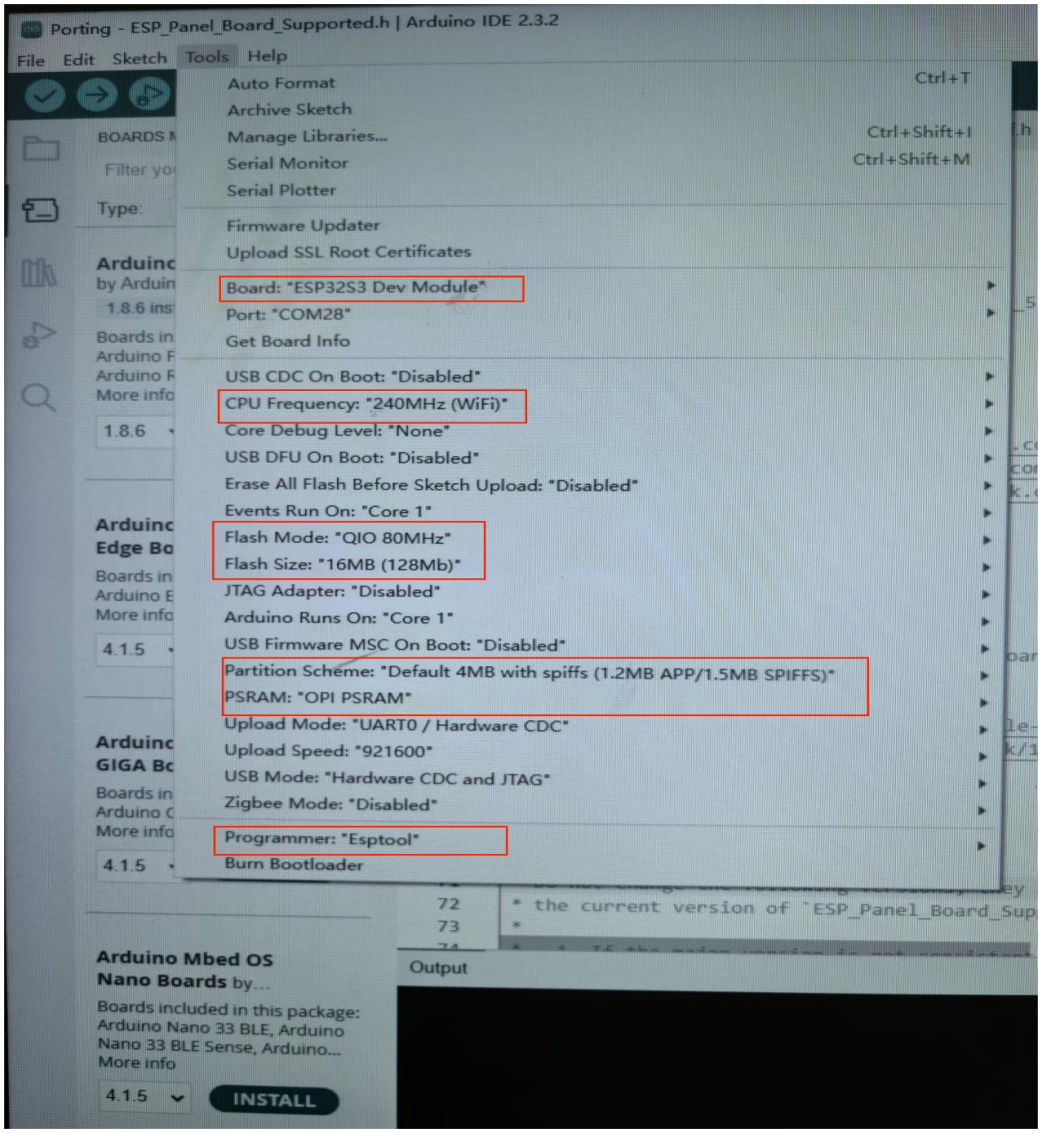



(18) CPU, Flash Mode, Flash Size, Partition Scheme, PSRAM.(For these configurations, please check the corresponding specifications of the product. It is detailed in 2.5 of the product specification)

(19) Set and In the programmer entry of the same tab, we choose "esptool ".

Note: These configuration must be set according to the specifications, otherwise your program may not run properly and the screen will not display normally.

Are configured as follows



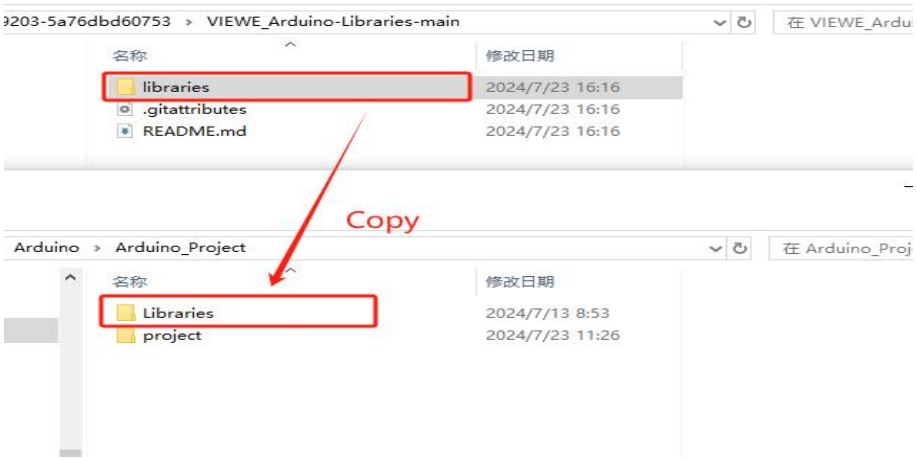

## ■ Library Installation

- Downloading library files:

① The lvgl library and other libraries require configuration files after installation. It is recommended to use our Libraries directly to avoid compilation failures.Please

refer to the last page of the specification and download it directly by clicking the link under 5.2.

② After unzipped, copy all the files in the libraries folder into the Libraries folder you created At installation time.



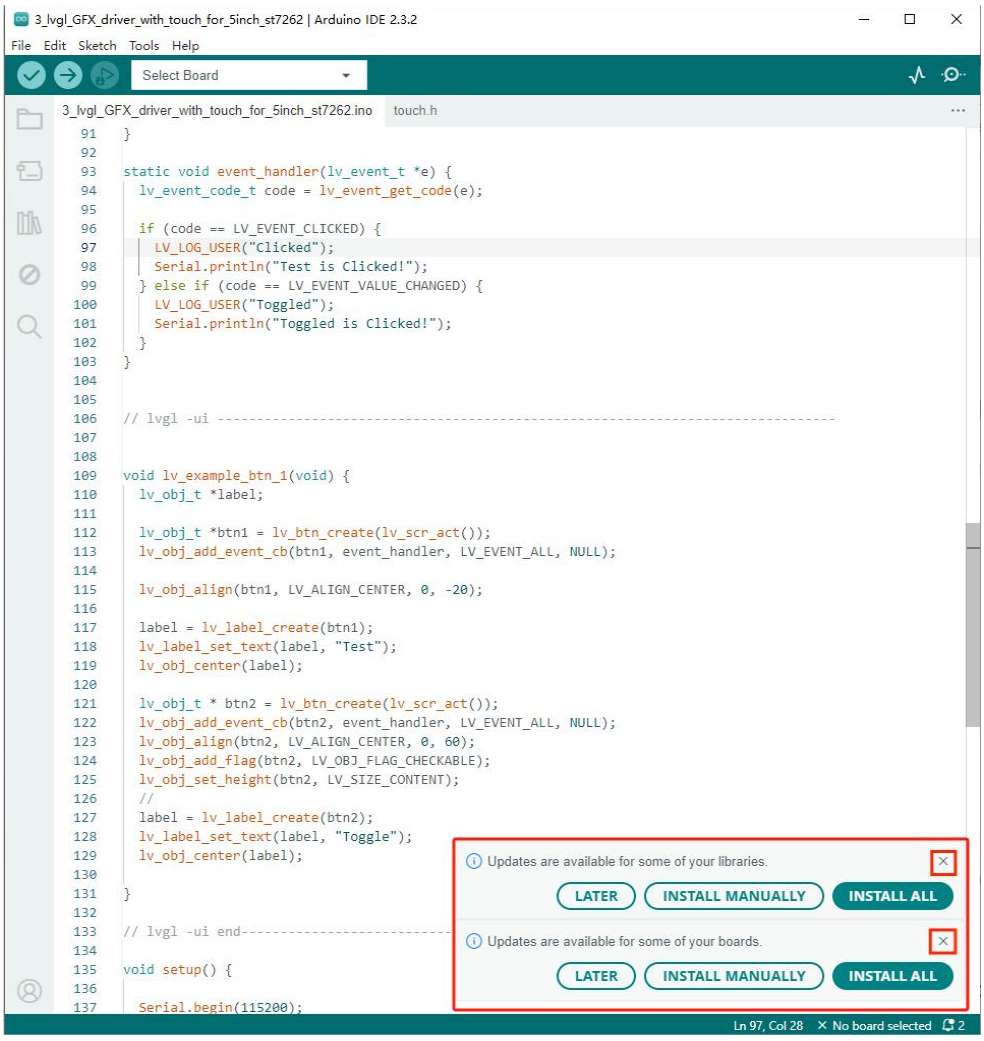

## ■ Arduino Sample Demo (here directly demonstrate the examples provided by the company)

Note: You need to ensure that the above steps to install Arduino have been completed before starting the following test demo

- Open Arduino IDE

Note: Every time you open the Arduino IDE, you will be prompted to update some library or development board. Once you've decided on a version, don't rush to update, because the functions and arguments supported by your repository and board versions can vary wildly from

minor versions to completely incompatible versions.




- Download Demo,Click here.

  Note: Please download the SDK according to the corresponding Product size and model, which can be downloaded according to the link under 5.1 on the last page of the product-specification provided to you.

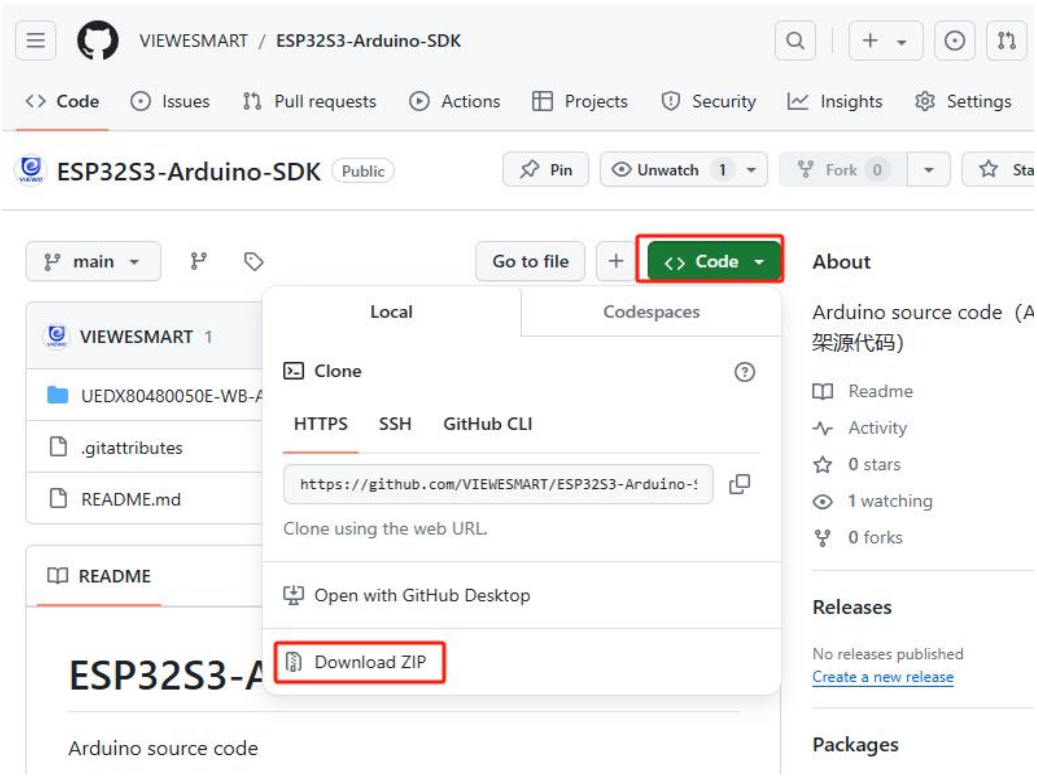
  - To download a specific folder from the GitHub web page, use the following method:

    Method 1: Directly download the entire repository and

extract the folders

1. Download the entire repository:

- ✧ Open the GitHub repo page where you want to download the folder.
- ✧ Click the green "Code" button, then select "Download ZIP".
- ✧ This will download a ZIP file of the entire repository.




2. Extract the file:

- ✧ Unzip the downloaded ZIP file.
- ✧ Find the folder you need and copy it to the place you want to save it.


Method 2: Use third-party tools to download the specified

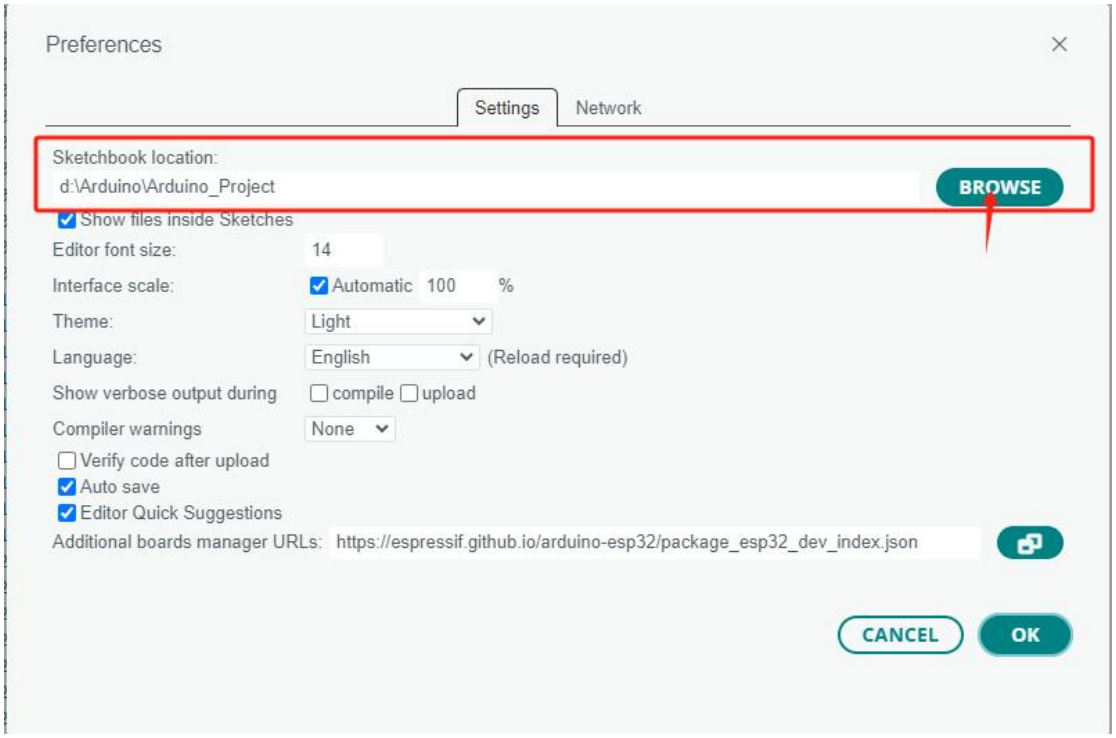
folder

If you only want to download specific folders and don't want to download the entire repository, you can use a third-party tool like DownGit:

1. Access DownGit:
   - ✧ Open the DownGit website.Click here.
2. Enter the folder URL:
   - ✧ Go back to the GitHub repo page and navigate to the folder you want to download.
   - ✧ Copy the URL from your browser's address bar.
3. Generate the download link:
   - ✧ Go back to the DownGit page and paste the copied URL into the input field.
   - ✧ Click the "Download" button.
4. Download the folder:
   - ✧ DownGit will generate a link that you can click to download directly to the folder you specify.

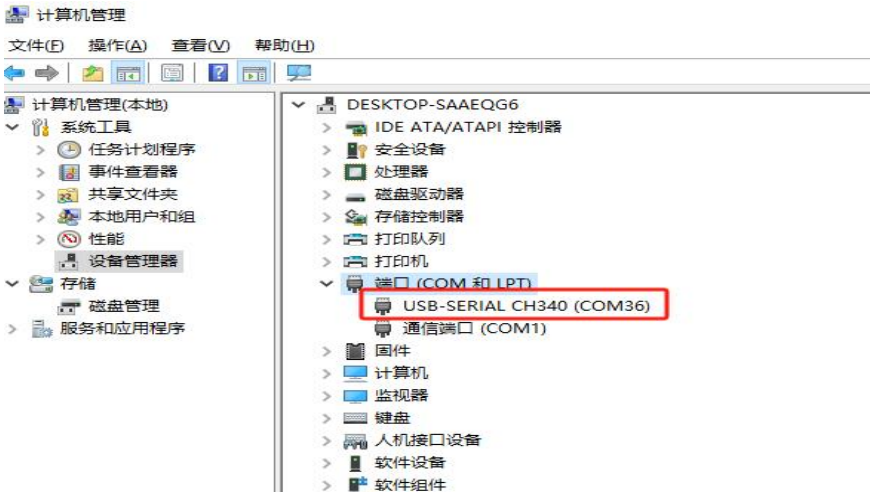

- Setting the project path
  - ■ Earlier, we created a folder called Arduino_Project, which contains libraries and projects. When we set the project

path to Arduino_Project, the IDE will automatically retrieve the libraries in our folder.Go to File->Preferences.

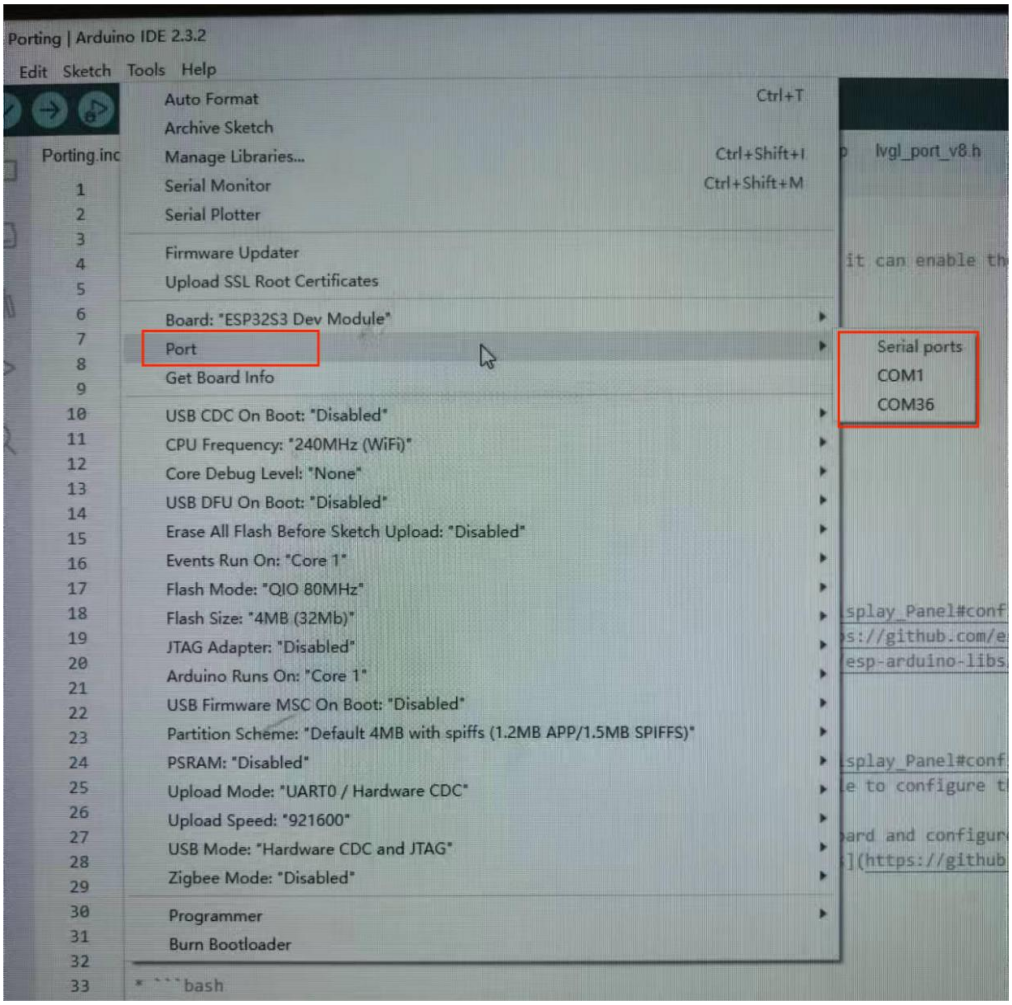



Note: If the project path is wrong, the required libraries in the application will not be retrieved, resulting in an error

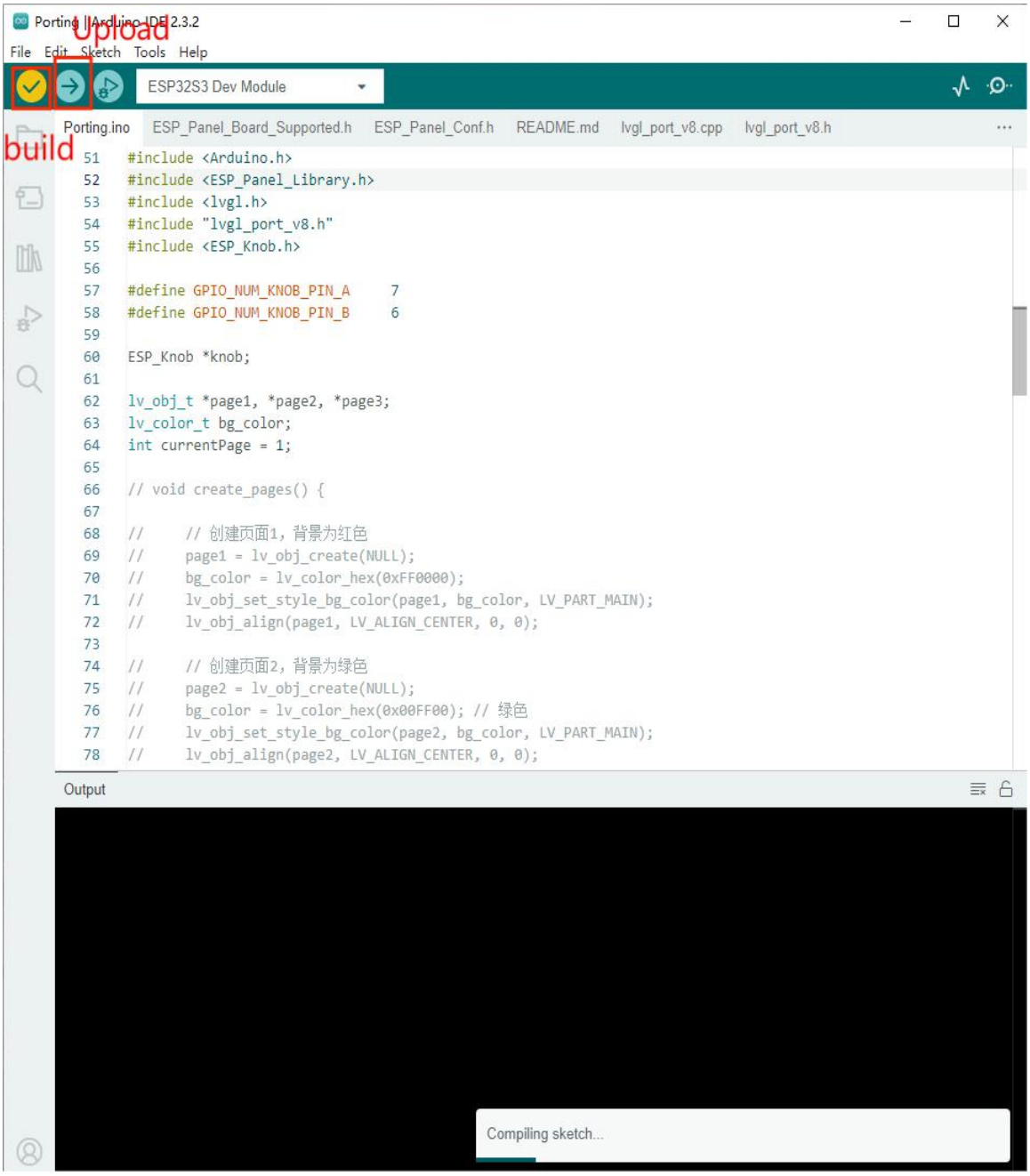
- Hardware link method (Install USB to serial port tool before this.Select the appropriate version according to your system.Click here)
  - After installation, when you plug in the board and look at the device manager in computer management, the COM and LPT ports appear as follows



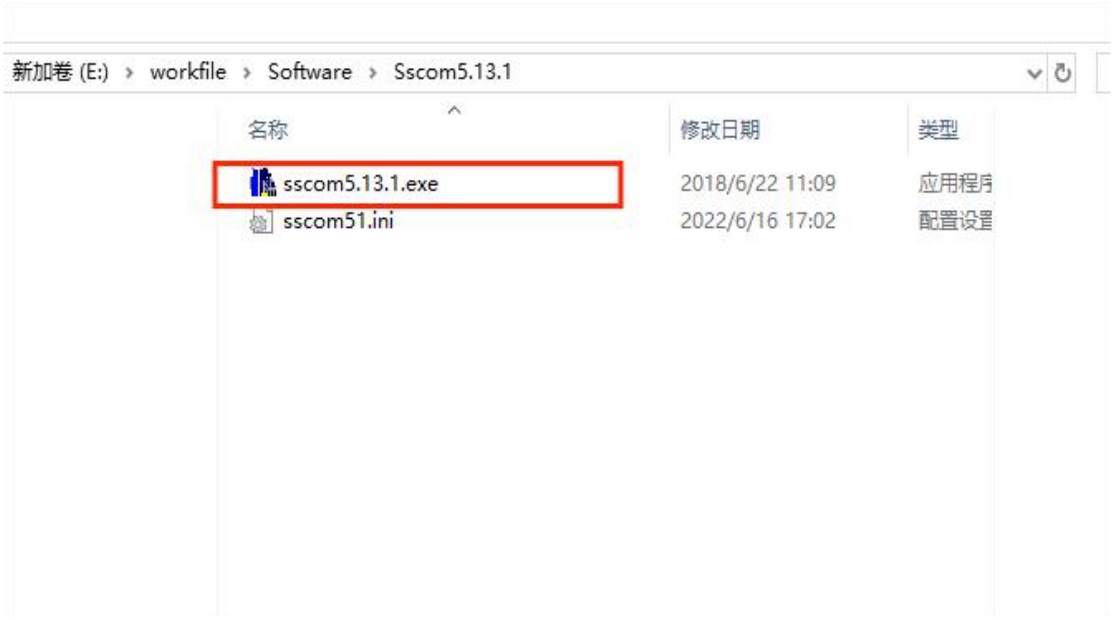

- Configuring Ports.(Select the port number corresponding to the above plug in development board, such as the above appears to be USB-SERIAL 340 (COM36), so select COM36.Corresponds to its own port)



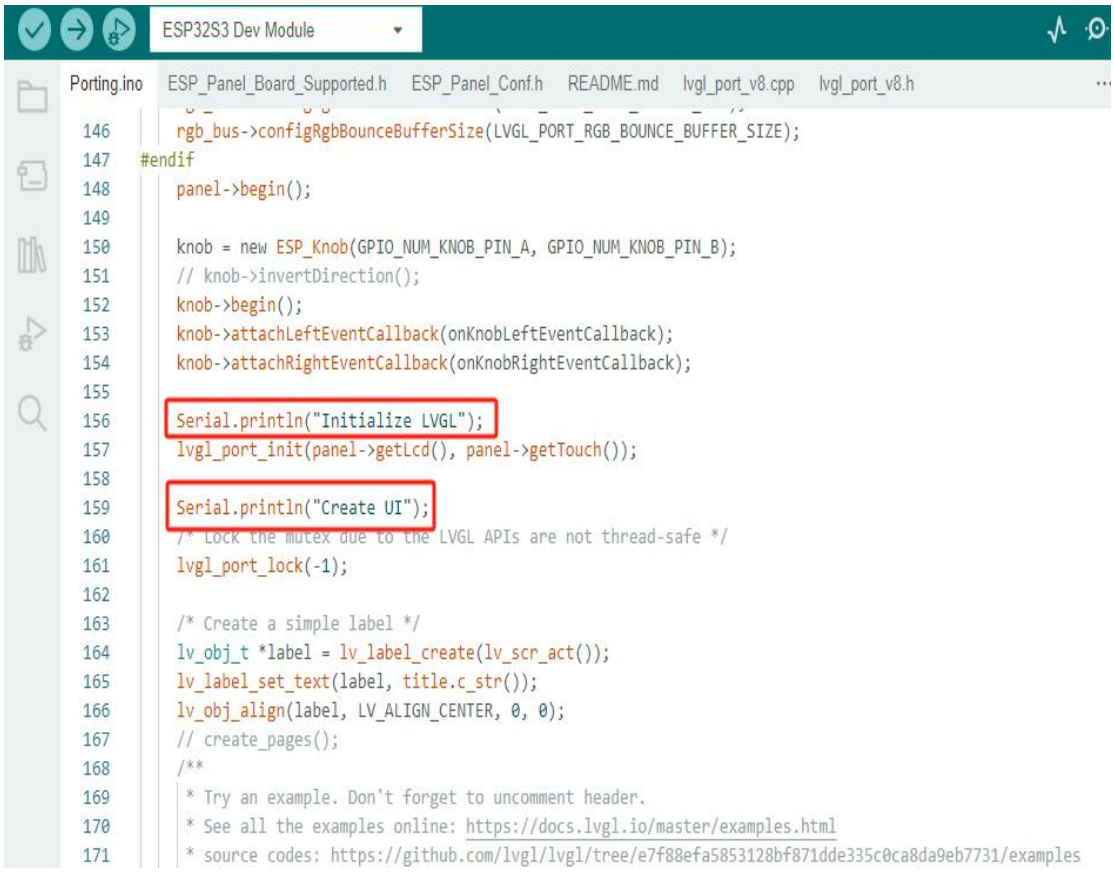

- Configuring CPU, Flash Mode, Flash Size, Partition Scheme, PSRAM, Programmer. See(18) (19) here.
- Build and Upload(Once this is done, you can compile and upload it, and it will be compiled again, but it should be compiled before uploading, just to be sure)



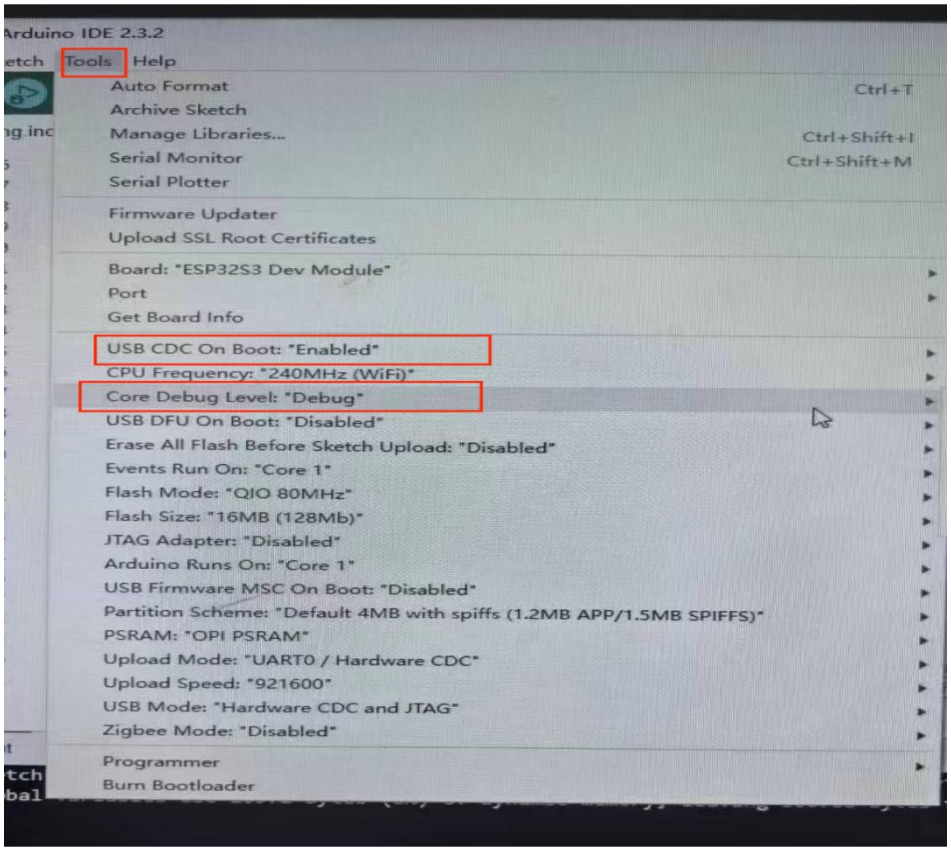

- Debug (if you encounter problems and need to use the serial port to debug: debugging tool)



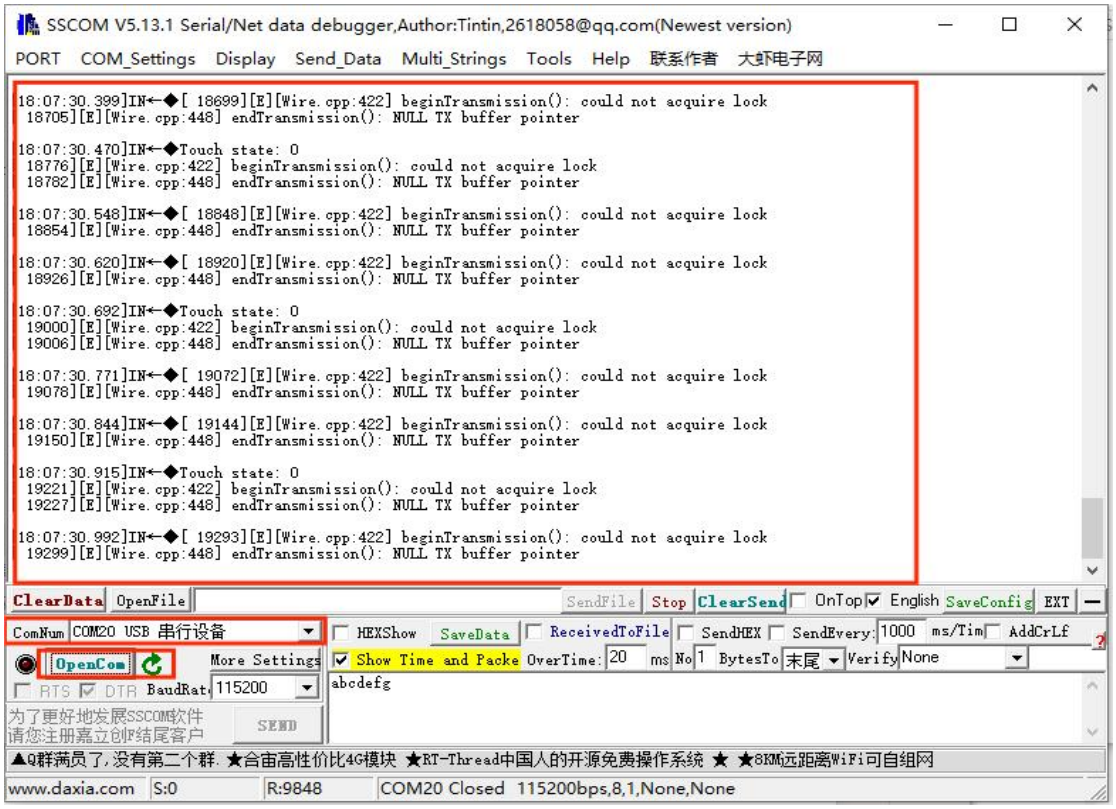

- Add debug statements where you need them, as follows




- Enable Debug




- Open the debug tool, select the appropriate COM, and click OpenCOM to see the information you want to print




# ➢ Resource

## ◆ GitHub

- VIEWESMART (github.com)

## ◆ Software

- Sscom5.13.1
- VSCode
- Arduino IDE
- USB to Serial Port Tool
  - Windows

- LINUX
- Android
- MacOS

- ESP32_S3_flash_download_tool

## FAQ

- **Question**:ESP32-S3 CAN reception failure?

**Answer:**

① Restart the COM port of the serial debugging Assistant and check tools->core Debug level: Debug in the Arduino IDE

② Uncheck DTR and RTS in the serial port debugging assistant.

- **Question**：The screen display shows no reaction after burning Arduino program？

**Answer:**

If the screen does not respond after burning the code, check whether it is correctly configured in Arduino IDE -> Tools: select the corresponding Flash(16MB) and enable PSRAM(8MB OPI).