



Welcome to the JCZN Workshop!

.....Table of contents.....

一、 Introduction.....2

二、 Installing using Arduino IDE.....2

三、 sample program usage.....11



Getting Started

Introduction

The objective of this post is to explain how to upload an Arduino program to the ESP32-1732S019 module, from JCZN .

<http://www.jczn1688.com/zlxz>

The ESP32-S3 WiFi and Bluetooth chip is the latest generation of Espressif products. It has a dual-core 32-bit MCU, which integrates WiFi HT40 and Bluetooth/BLE 4.2 technology inside.

ESP32-S3-wroom-1 has a significant performance improvement. It is equipped with a high-performance dual-core Tensilica LX7 MCU. One core handles high speed connection and the other for standalone application development. The dual-core MCU has a 240 MHz frequency and a computing power of 600 DMIPS.

In addition, it supports Wi-Fi HT40, Classic Bluetooth/BLE 4.2, and more GPIO resources.

Installing using Arduino IDE

Programming the ESP32-S3

An easy way to get started is by using the familiar Arduino IDE. While this is not necessarily the best environment for working with the ESP32-S3, it has the advantage of being a familiar application, so the learning curve is flattened.

We will be using the Arduino IDE for our experiments.

1, Installing using Arduino IDE

we first need to install version 1.8.19 of the Arduino IDE (or greater),for example, the Arduino installation was in "C:/Programs(x86)/Arduino".

download release link:

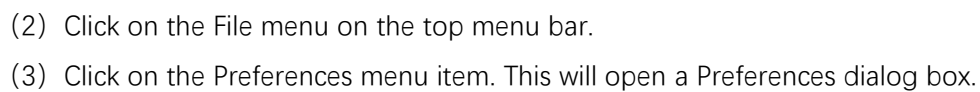
<https://downloads.arduino.cc/arduino-1.8.19-windows.exe>

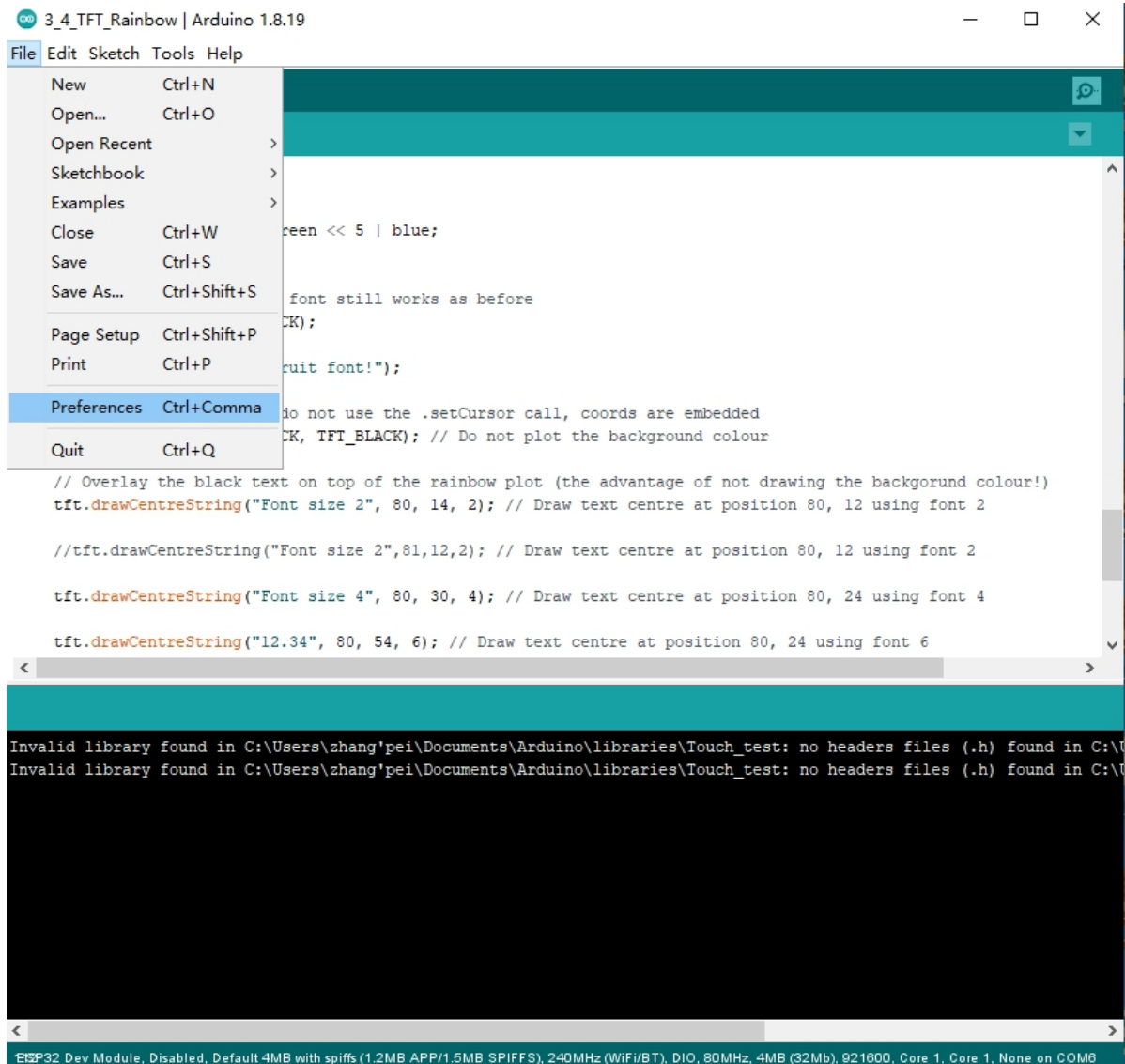
2, This is the way to install Arduino-ESP32 directly from the Arduino IDE.

Add Boards Manager Entry

Here is what you need to do to install the ESP32 boards into the Arduino IDE:

- (1) Open the Arduino IDE.





(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 comma at the end of it, then follow the next step.

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

Stable release link:

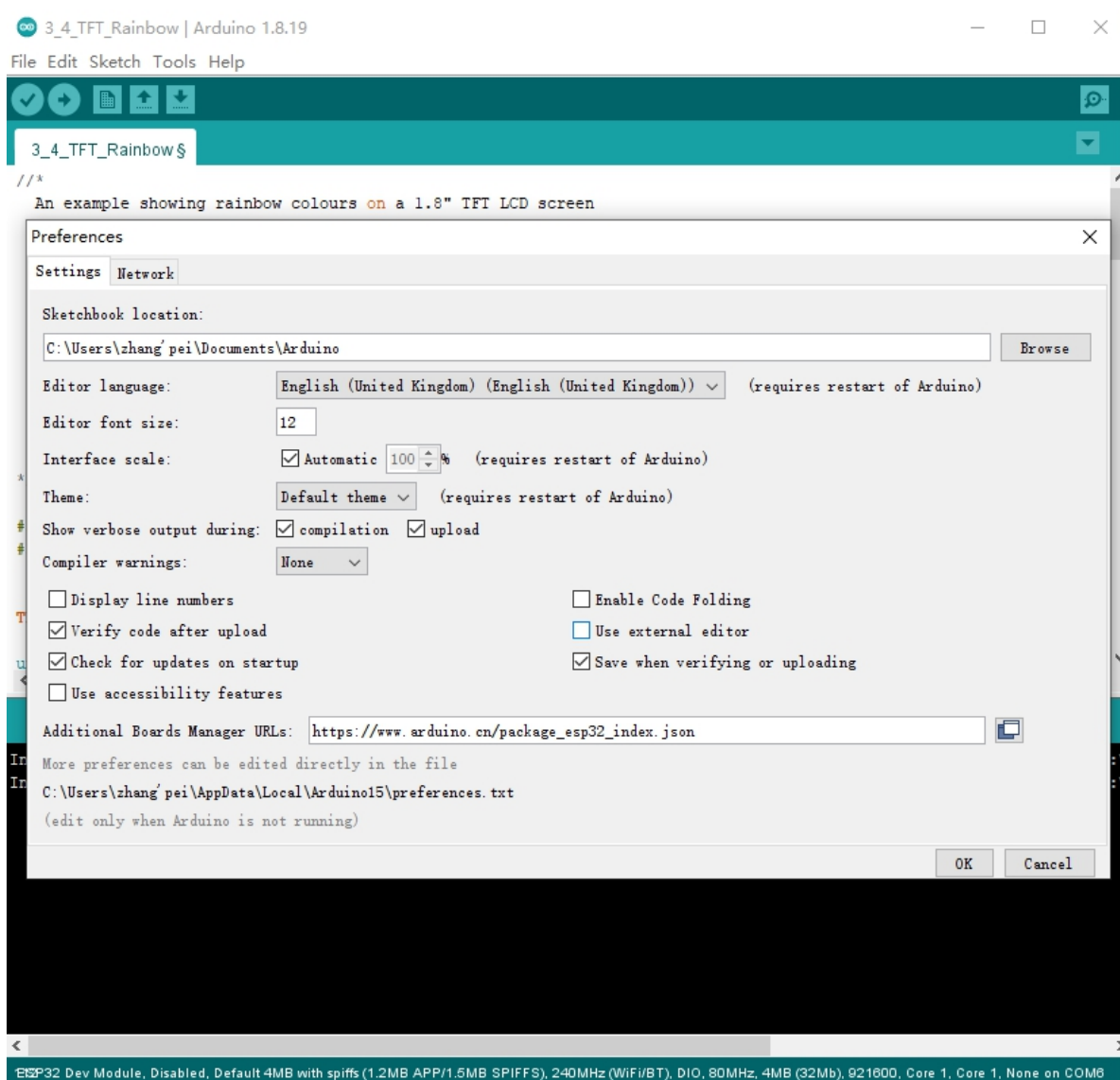
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

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 link in it is illustrated here:



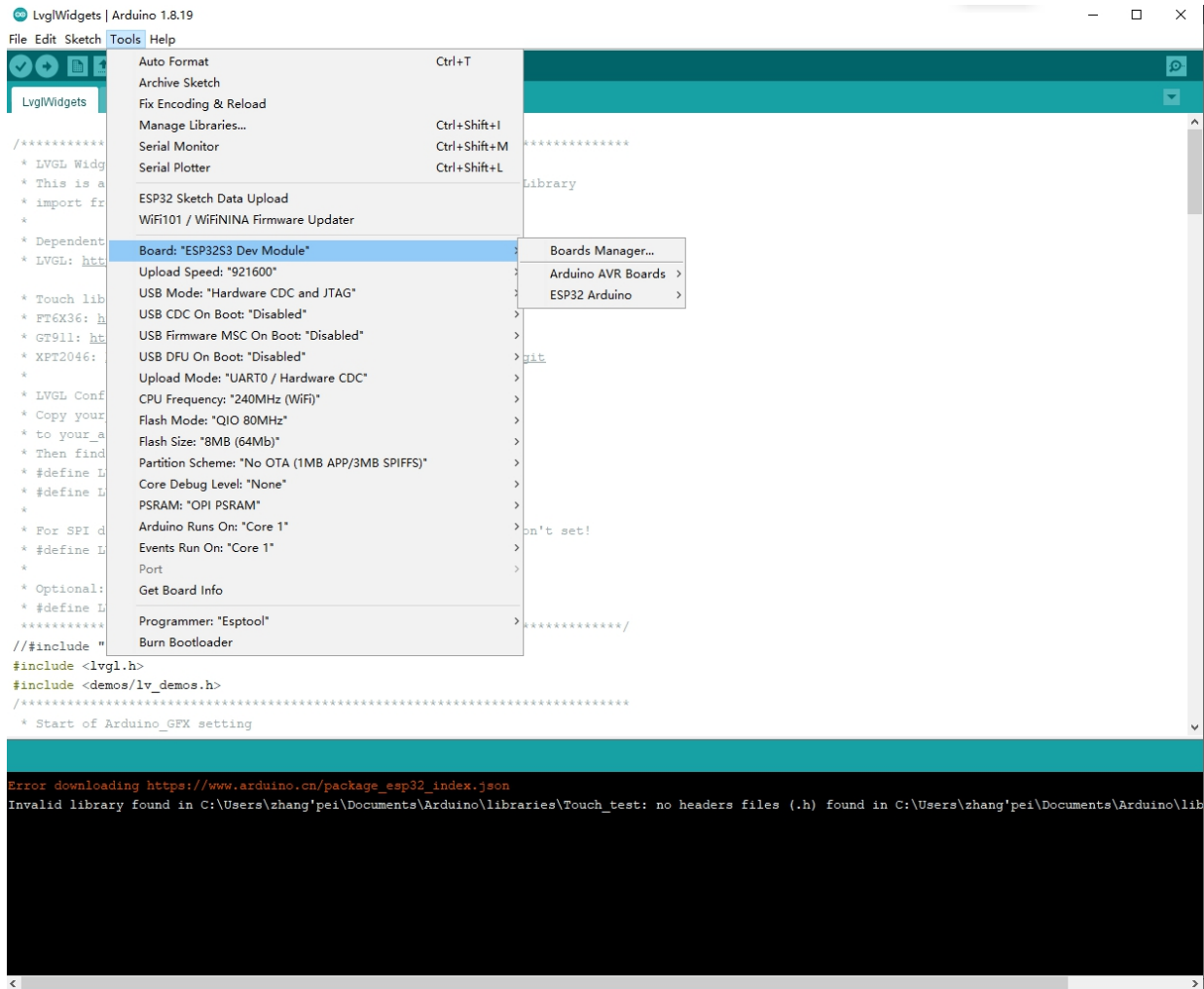
(9) In the Arduino IDE click on the Tools menu on the top menu bar.

(10) Scroll down to the Board: entry

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

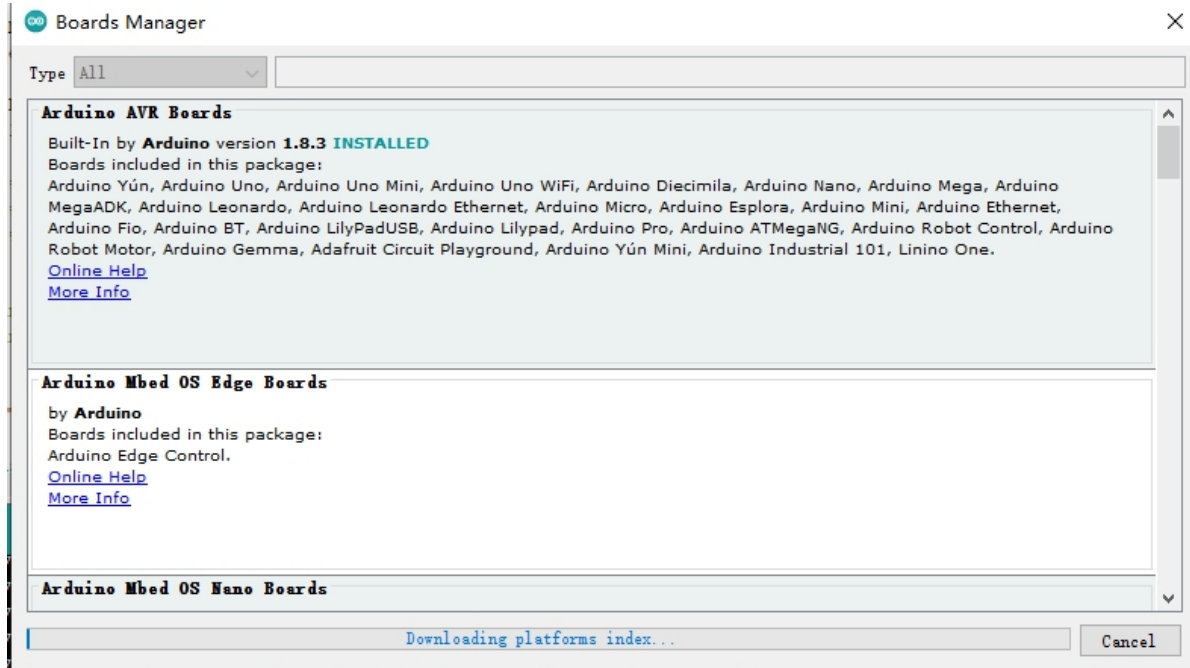
(12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.

(13) In the search box in the Boards Manager enter "esp32".

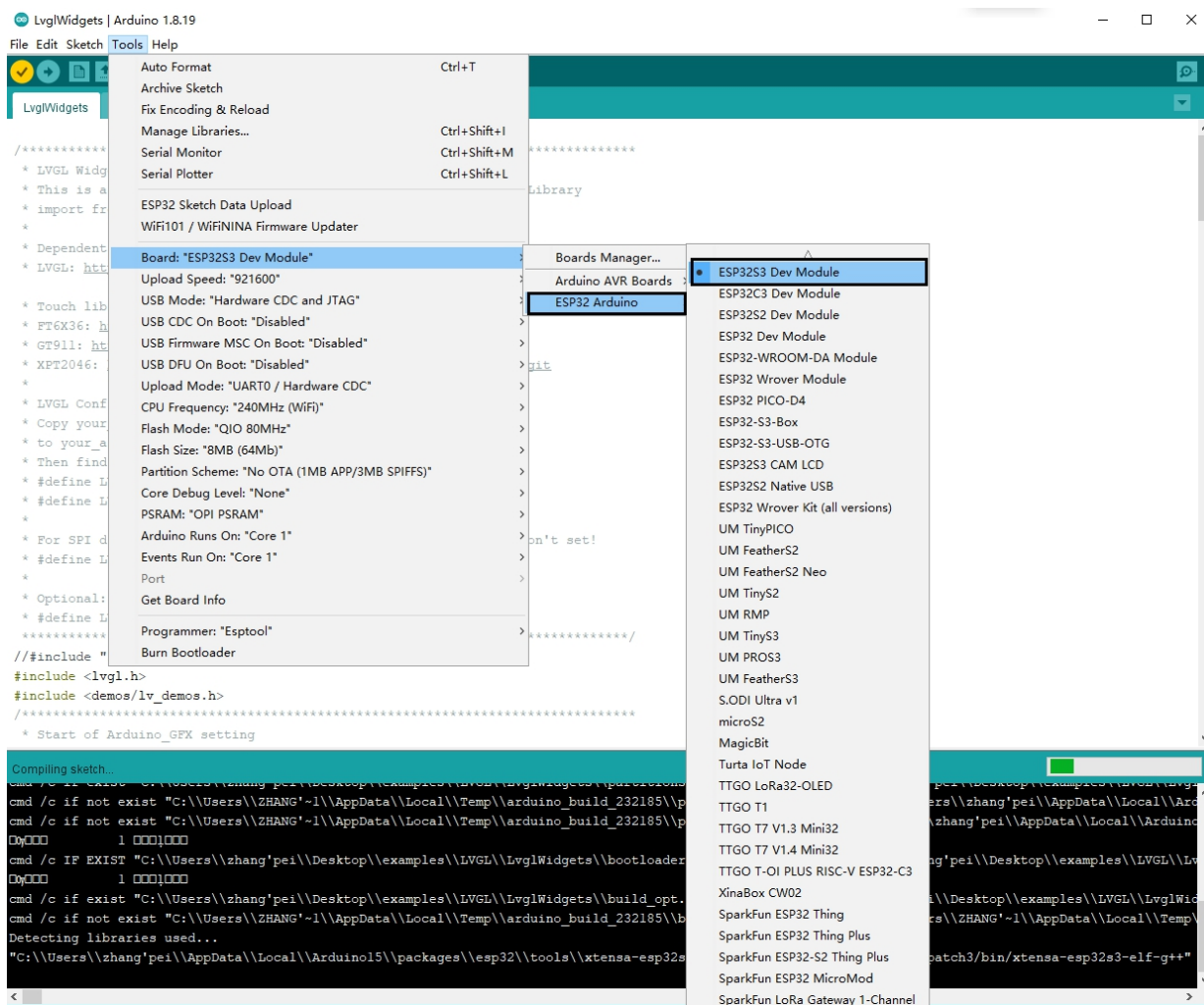


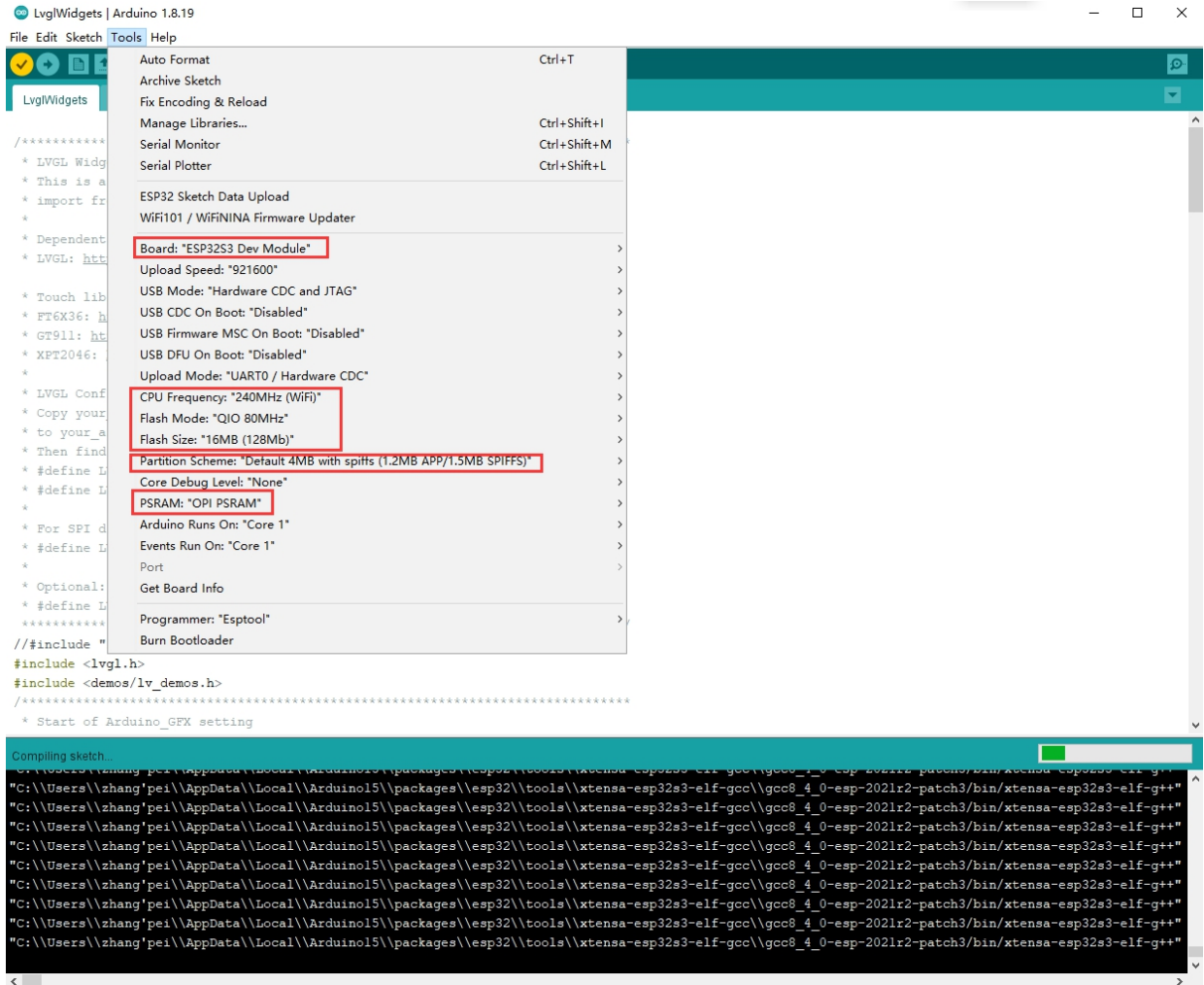
(14) You should see an entry for “esp32 by Espressif Systems”. Highlight this entry and click on the Install button.

This will install the ESP32 boards into your Arduino IDE

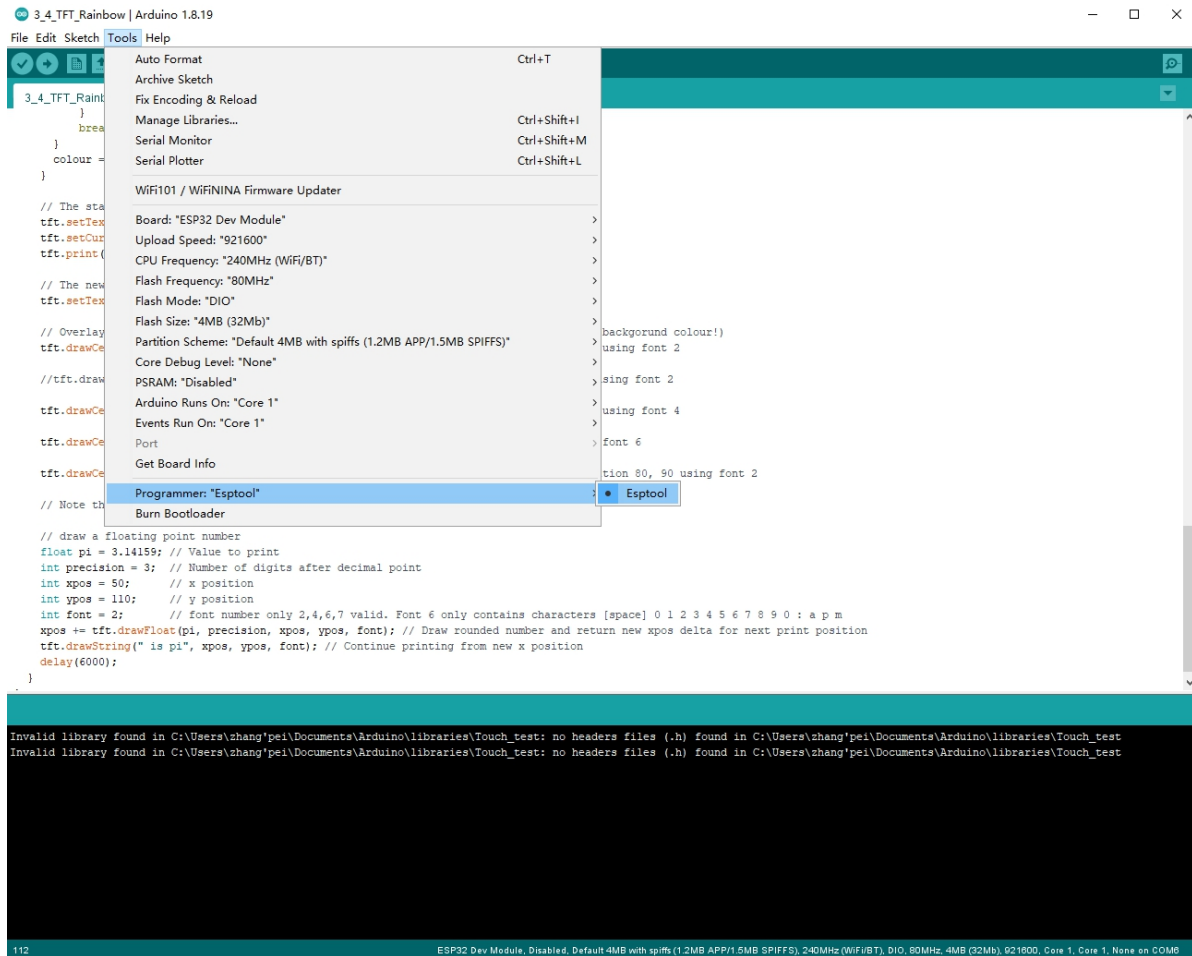


Once the installation completes, we need to select the correct board options for the "ESP32 Arduino" board. In the board type, in the tools tab, we choose "ESP32S3 Dev Module".





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



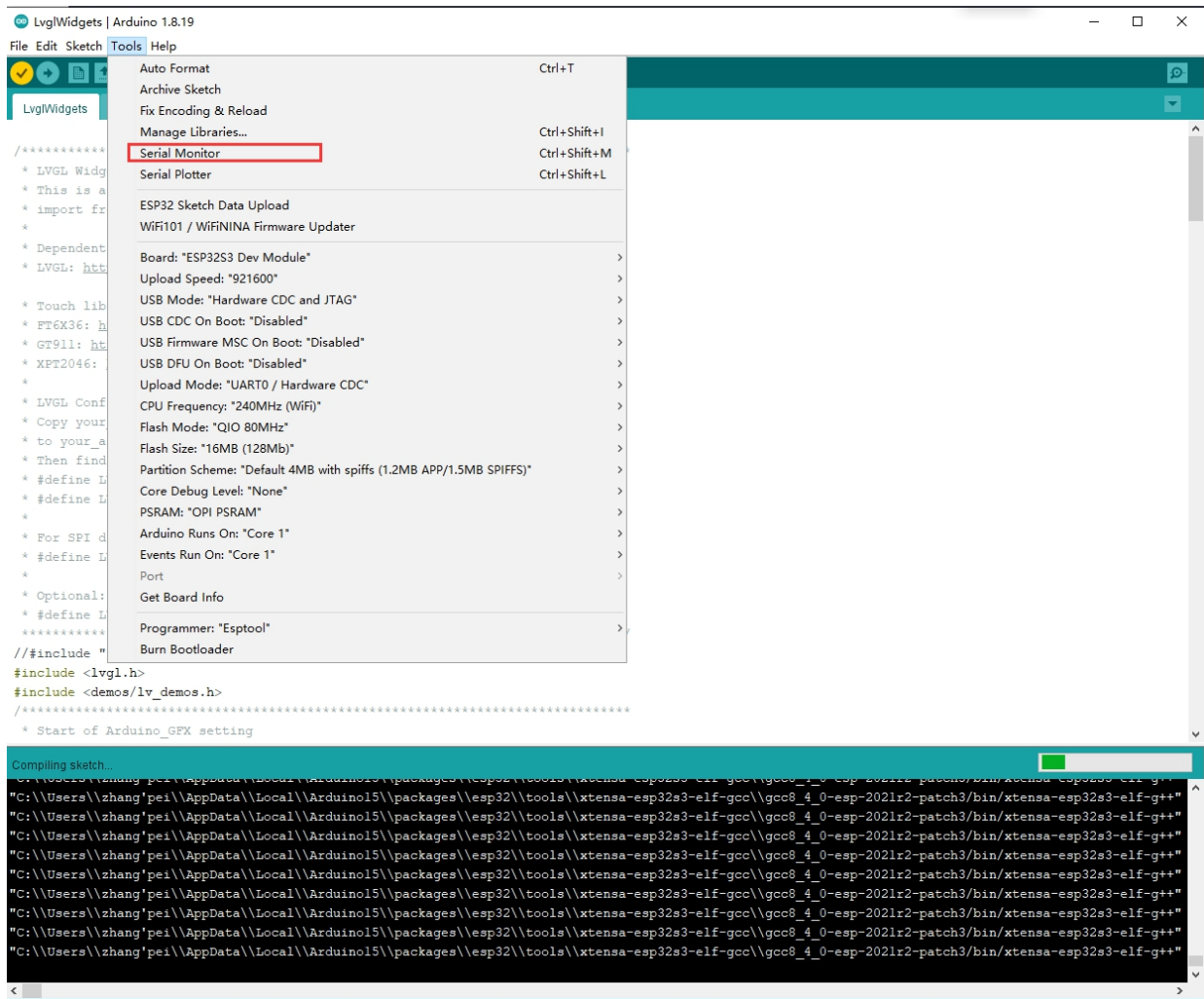
It's important to note that after the code is uploaded, the device will start to run it. So, if we want to upload a new program, we need to reset the power of the device, in order to guarantee that it enters flashing mode again.

First program

Since this platform is based on Arduino, we can use many of the usual functions. As an example for the first program, the code below starts the Serial port and prints "hello from ESP32" every second.

```
void setup() {  
  Serial.begin(115200);  
}  
  
void loop() {  
  Serial.println("hello from ESP32");  
  delay(1000);  
}
```

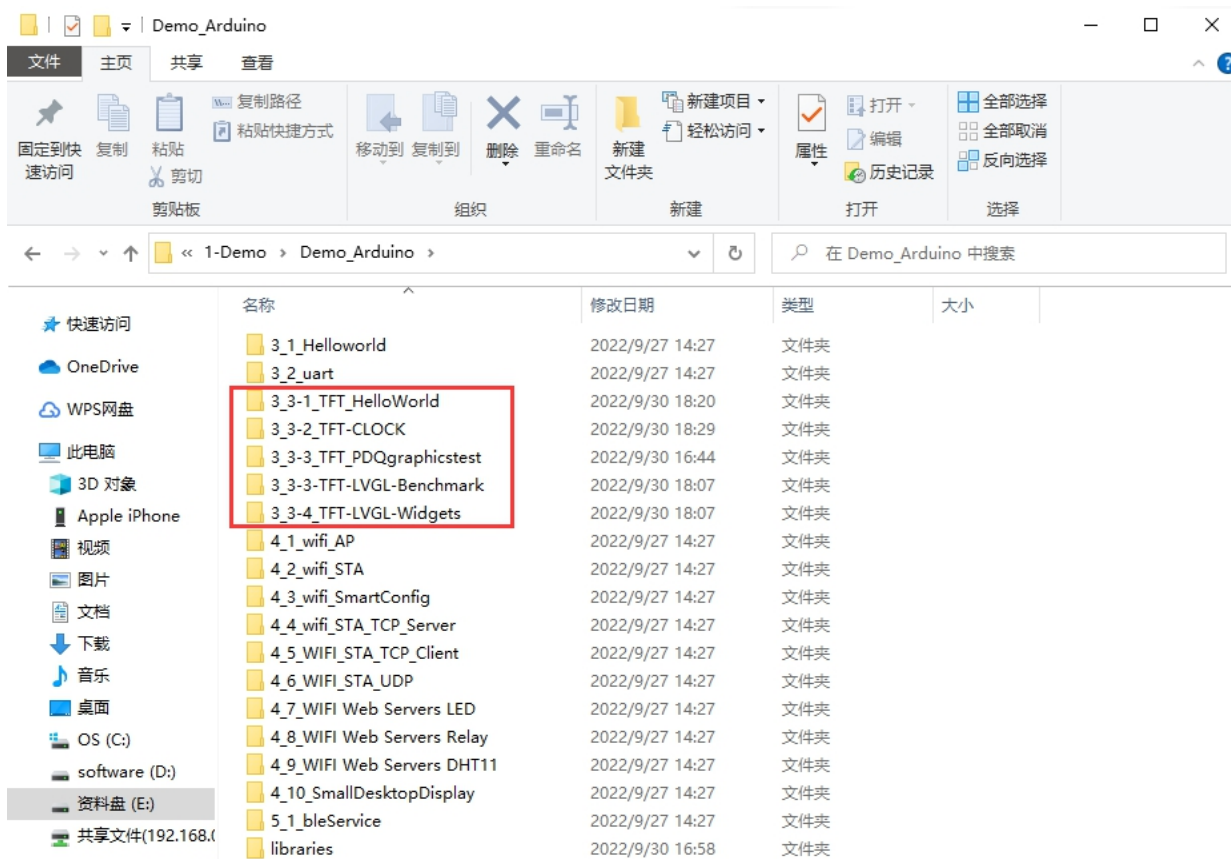
If everything is working fine, we will see the output in the serial console shown.



Again thank you for so much concern.. Hopefully, it's the beginning of a wonderful relationship!

Sample program usage

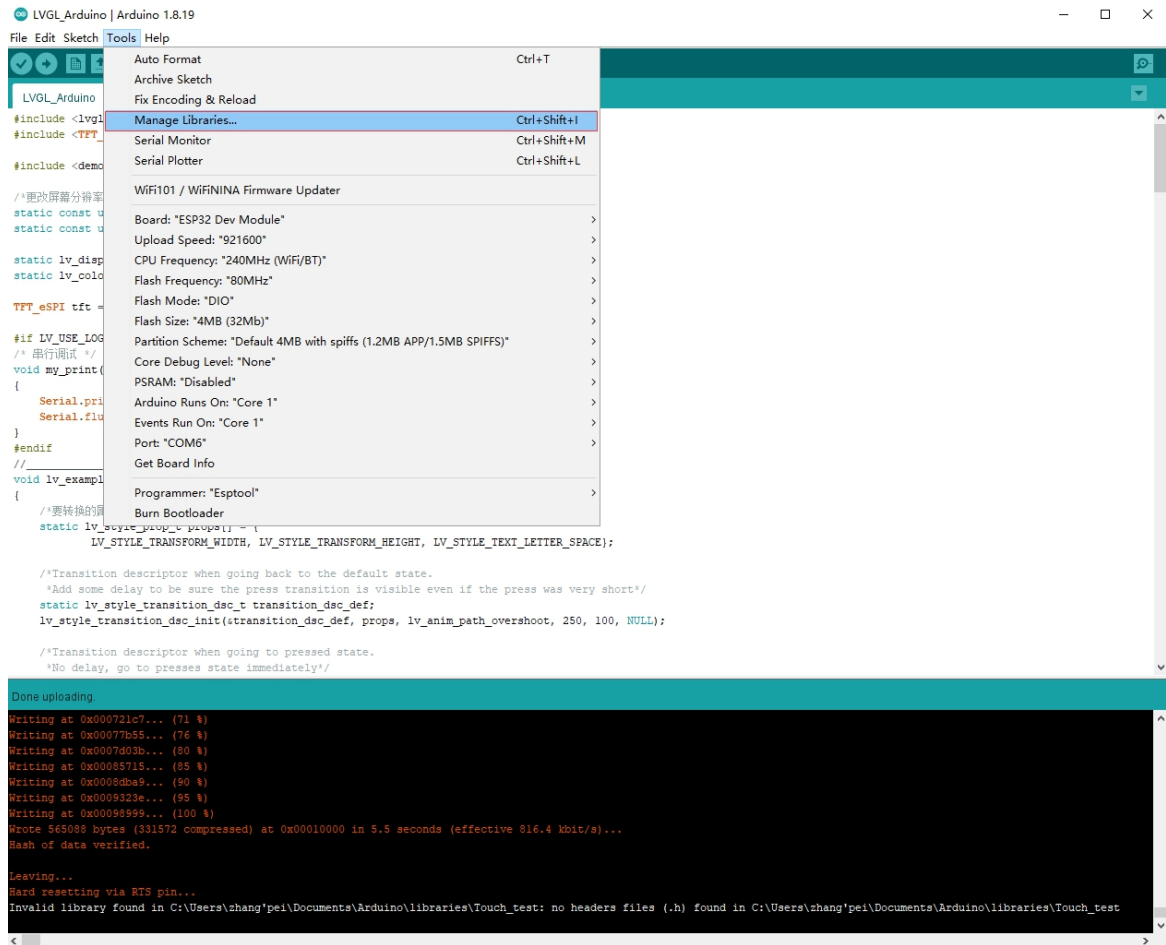
At present, only a preliminary explanation and introductory use are given to the samples displayed on the screen, and the corresponding examples in the data center are found, as shown in the figure:

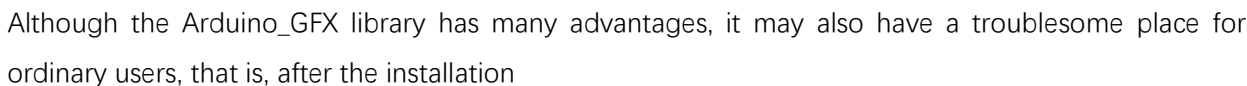


The examples in the red circle are all based on the Arduino_GFX library as the basic application. This library supports various commonly used driver chips, such as ST7735, ST7789, ILI9341, etc., and has good compatibility.

Arduino_GFX library file installation:

Open the library manager in Arduino, search for Arduino_GFX, and click instal .





Find the data center 3_3-3_TFT_PDQgraphicstest

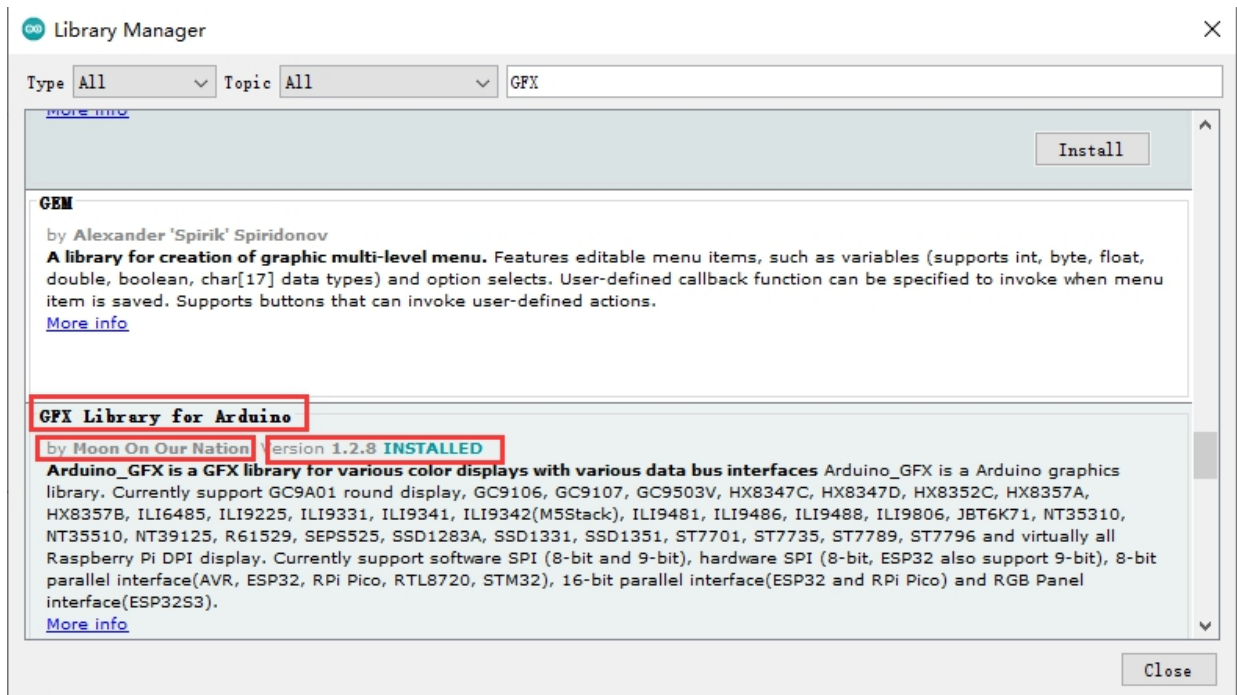
As shown:



File Explorer window showing the directory structure of the Demo_Arduino project. The path is: 1.9inch_ESP32-1732S019 > 1-Demo > Demo_Arduino.

名称	修改日期	类型
3_1_Helloworld	2022/12/6 13:01	文件夹
3_2_Uart	2022/12/6 13:01	文件夹
3_3-1_TFT_HelloWorld	2022/12/6 13:01	文件夹
3_3-2_TFT-CLOCK	2022/12/6 13:01	文件夹
3_3-3_TFT_PDQgraphicstest	2022/12/13 9:06	文件夹
3_3-3-TFT-LVGL-Benchmark	2022/12/6 13:01	文件夹
3_3-4_TFT-LVGL-Widgets	2022/12/6 13:01	文件夹
4_1_Wifi_AP	2022/12/6 13:01	文件夹
4_2_Wifi_STA	2022/12/6 13:01	文件夹
4_3_Wifi_SmartConfig	2022/12/6 13:01	文件夹
4_4_Wifi_STA_TCP_Server	2022/12/6 13:01	文件夹
4_5_WIFI_STA_TCP_Client	2022/12/6 13:01	文件夹
4_6_WIFI_STA_UDP	2022/12/6 13:01	文件夹
4_7_WIFI Web Servers LED	2022/12/6 13:01	文件夹
4_8_WIFI Web Servers Relay	2022/12/6 13:01	文件夹
4_9_WIFI Web Servers DHT11	2022/12/6 13:01	文件夹
4_10_SmallDesktopDisplay	2022/12/6 13:01	文件夹
5_1_BleService	2022/12/6 13:01	文件夹
6_1_Audio_test.ino	2022/12/6 13:01	文件夹
Libraries	2022/12/6 13:01	文件夹

Download library files .
Arduino_GFX library



After compiling, you can run PDQgraphicstest normally.