



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 JC3636W518 module, from JCZN .

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

The ESP32 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

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, 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.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** 3_4_TFT_Rainbow | Arduino 1.8.19
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Standard icons for file operations.
- Sketch Name:** 3_4_TFT_Rainbow
- Code Content:**

```
/*
An example showing rainbow colours on a 1.8" TFT LCD screen
and to show a basic example of font use.

Make sure all the display driver and pin connections are correct by
editing the User_Setup.h file in the TFT_eSPI library folder.

Note that yield() or delay(0) must be called in long duration for/while
loops to stop the ESP8266 watchdog triggering.

#####
##### DON'T FORGET TO UPDATE THE User_Setup.h FILE IN THE LIBRARY #####
#####

*/
#include <TFT_eSPI.h> // Graphics and font library for ST7735 driver chip
#include <SPI.h>

TFT_eSPI tft = TFT_eSPI(); // Invoke library, pins defined in User_Setup.h

unsigned long targetTime = 0;
```
- Serial Monitor:** Shows two error messages:
Invalid library found in C:\Users\zhang\Documents\Arduino\libraries\Touch_test: no headers files (.h) found in C:\U
Invalid library found in C:\Users\zhang\Documents\Arduino\libraries\Touch_test: no headers files (.h) found in C:\U
- Bottom Status Bar:** ESP32 Dev Module, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), DIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM6

- (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.



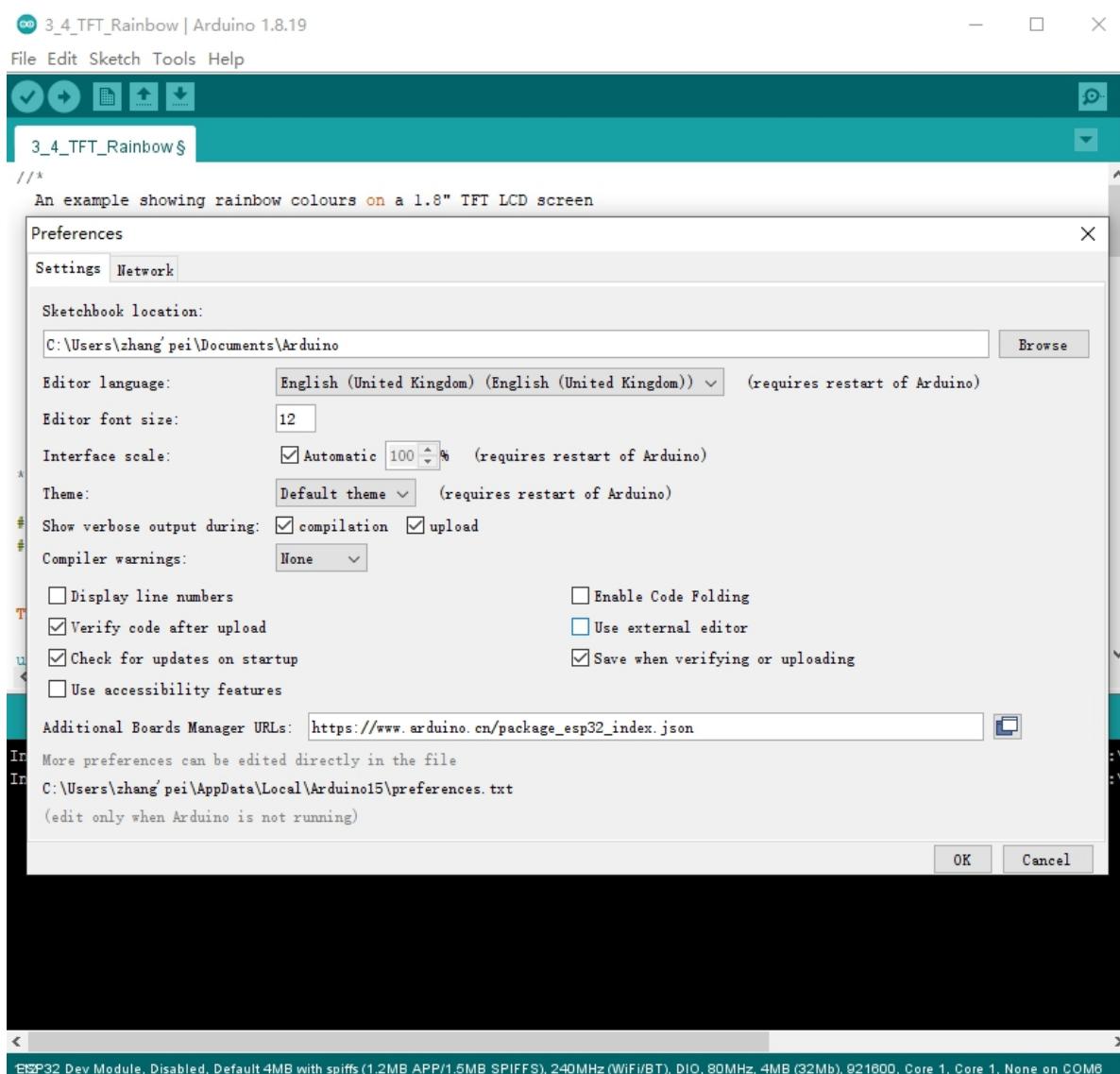
The screenshot shows the Arduino IDE interface with the title bar "3_4_TFT_Rainbow | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. A context menu is open over some code, with options like New, Open..., Open Recent, Sketchbook, Examples, Close, Save, Save As..., Page Setup, Print, Preferences (which is highlighted in blue), and Quit. The main code area contains several lines of TFT library code, including font definitions and drawCentreString() calls. At the bottom of the code area, there are two error messages: "Invalid library found in C:\Users\zhang'pei\Documents\Arduino\libraries\Touch_test: no headers files (.h) found in C:\U" repeated twice. The status bar at the bottom right indicates "ESP32 Dev Module, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), DIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM6".

- (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:

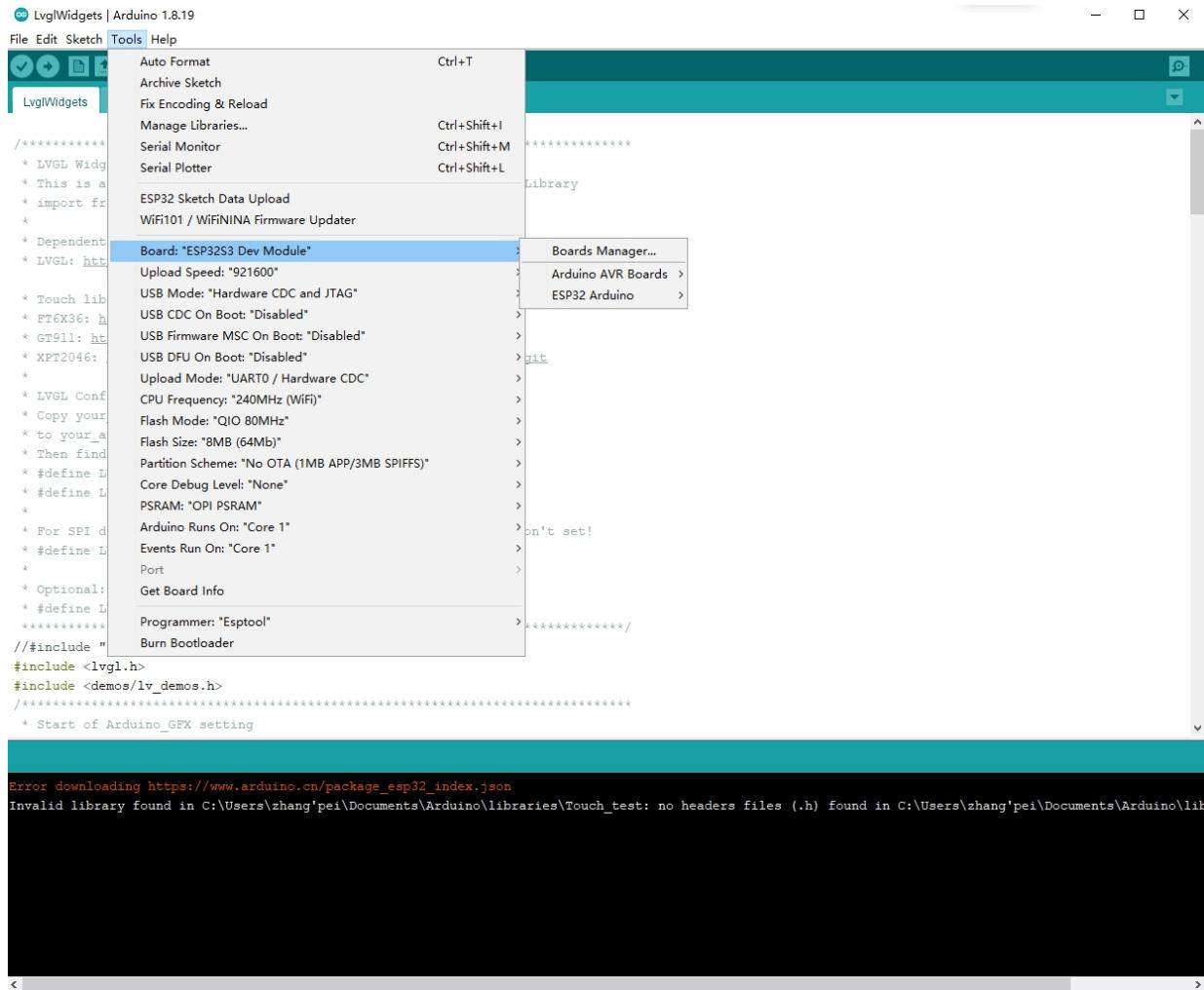
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

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

The textbox with the JSON link in it is illustrated here:

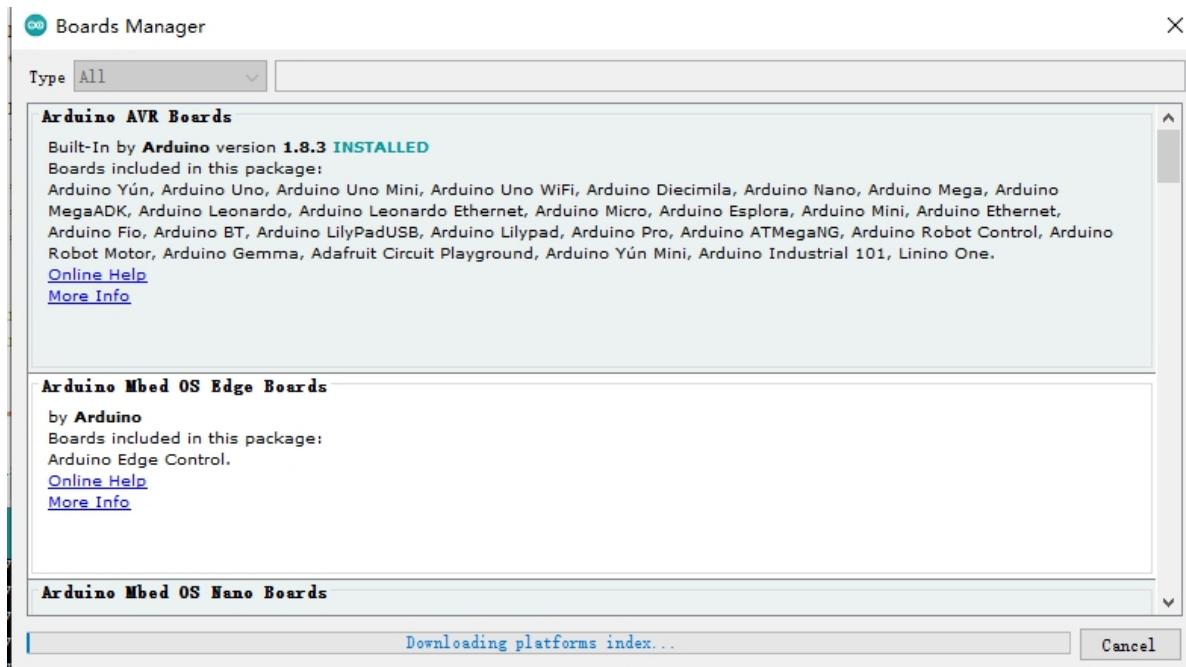


- (9) In the Arduino IDE click on the Tools menu on the top menu bar.
- (10) Scroll down to the Board: entry
- (11) A submenu will open when you highlight the Board: entry.
- (12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.
- (13) In the search box in the Boards Manager enter "esp32".

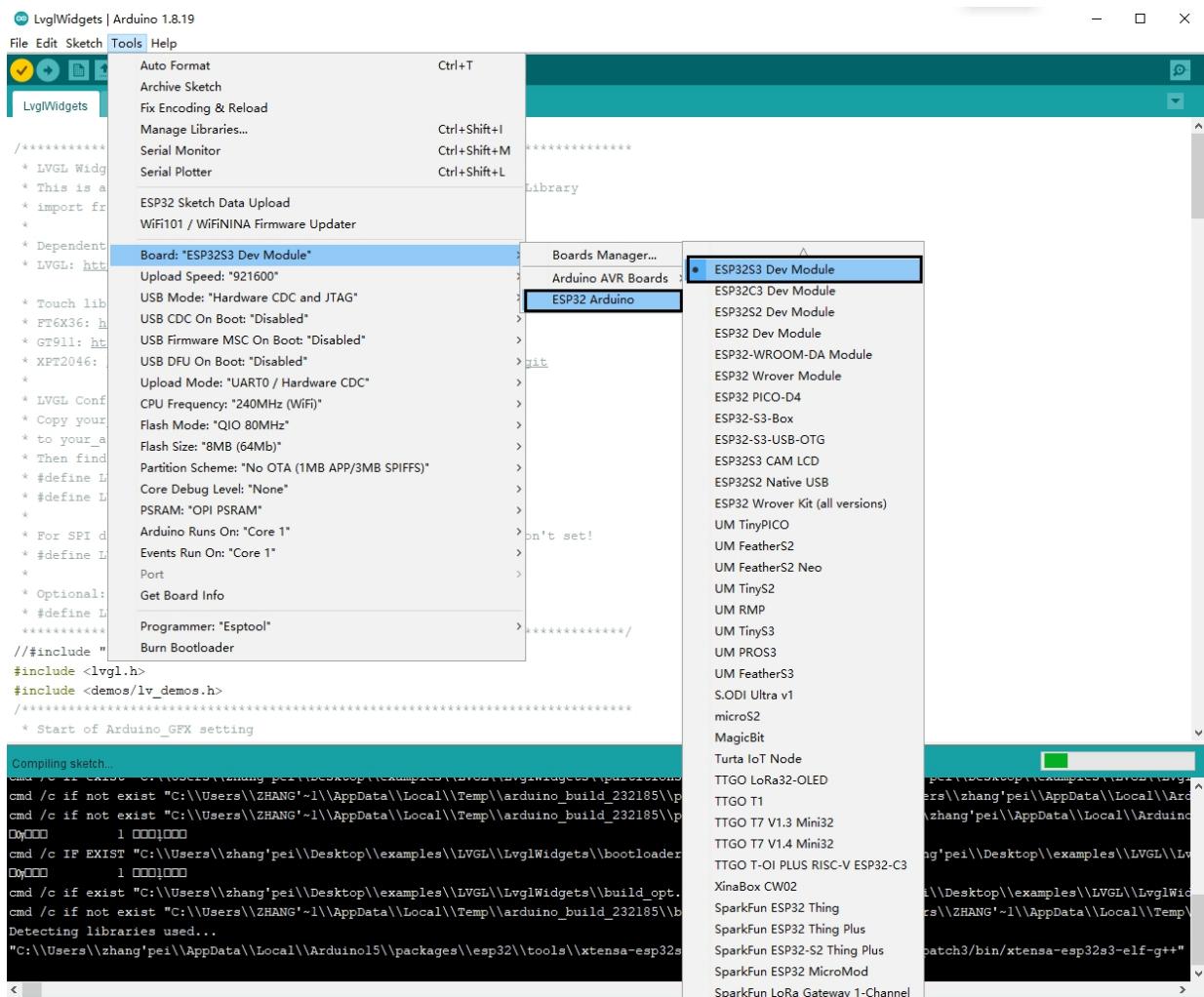


(14) You should see an entry for “esp32 by Espressif Systems”. Highlight this entry and click on the Install button.

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".





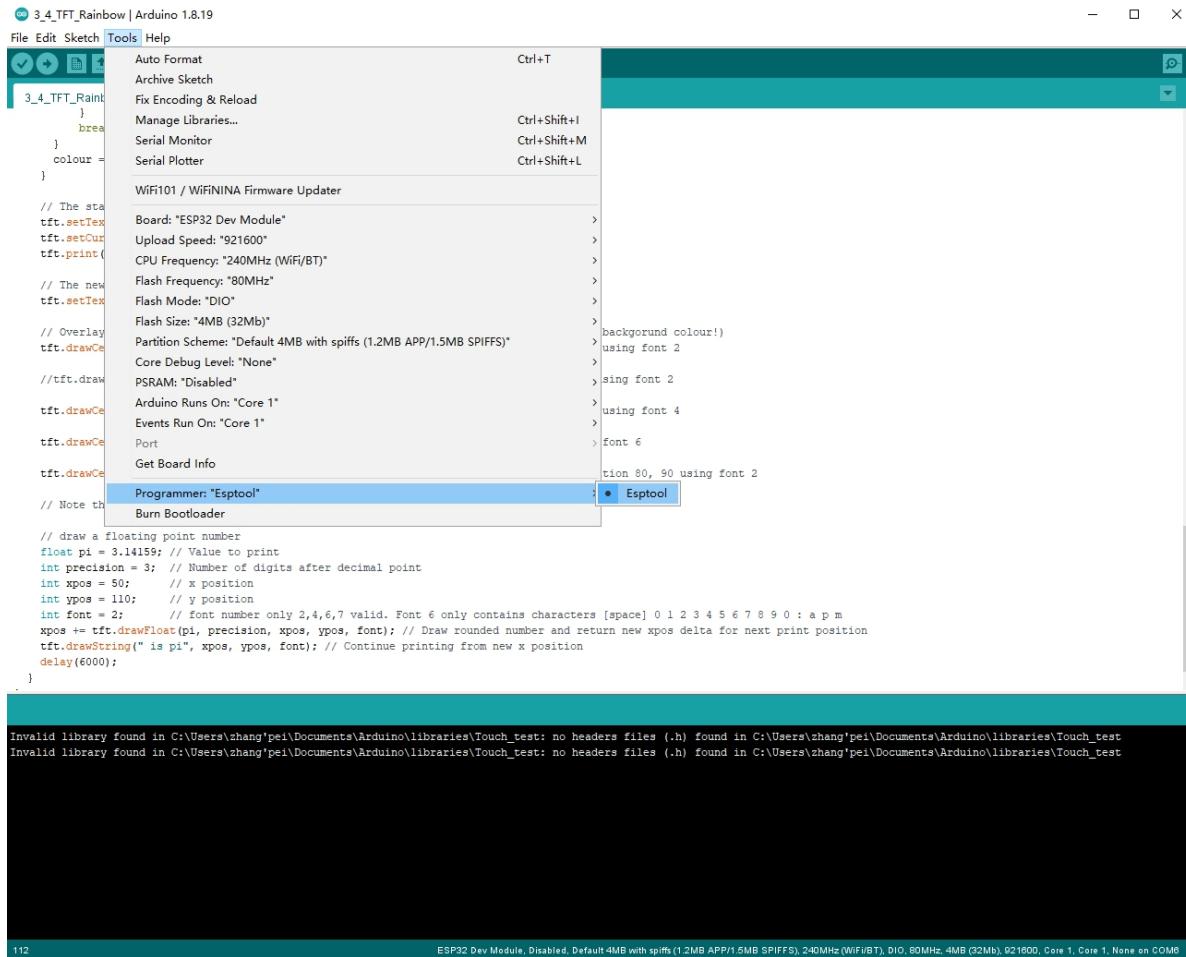
ST77916_LVGL_DEMO - scr_st77916.h | Arduino 1.8.19

File Edit Sketch Tools Help

```
ST77916_LVGL
} // 输入值为0-1
void set_bri
{
    if (NULL =
        return;
    backlight-
}
static void
{
    ESP_PanelT
    ESP_PanelT
    int read_t
    if (read_t
    {
        data->po
        data->po
        data->st
    }
    else
    {
        data->st
    }
}
static lv_in
Done uploading.
Writing at 0x
Writing at 0x
```

Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiINA Firmware Updater
Board: "ESP32S3 Dev Module"
Upload Speed: "921600"
USB Mode: "Hardware CDC and JTAG"
USB CDC On Boot: "Enabled"
USB Firmware MSC On Boot: "Disabled"
USB DFU On Boot: "Disabled"
Upload Mode: "UART0 / Hardware CDC"
CPU Frequency: "240MHz (WiFi)"
Flash Mode: "QIO 120MHz"
Flash Size: "16MB (128Mb)"
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "OPI PSRAM"
Arduino Runs On: "Core 1"
Events Run On: "Core 1"
Erase All Flash Before Sketch Upload: "Disabled"
JTAG Adapter: "Disabled"
Zigbee Mode: "Disabled"
Port: "COM1318 (ESP32 Family Device)"
Get Board Info
Programmer
Burn Bootloader

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.

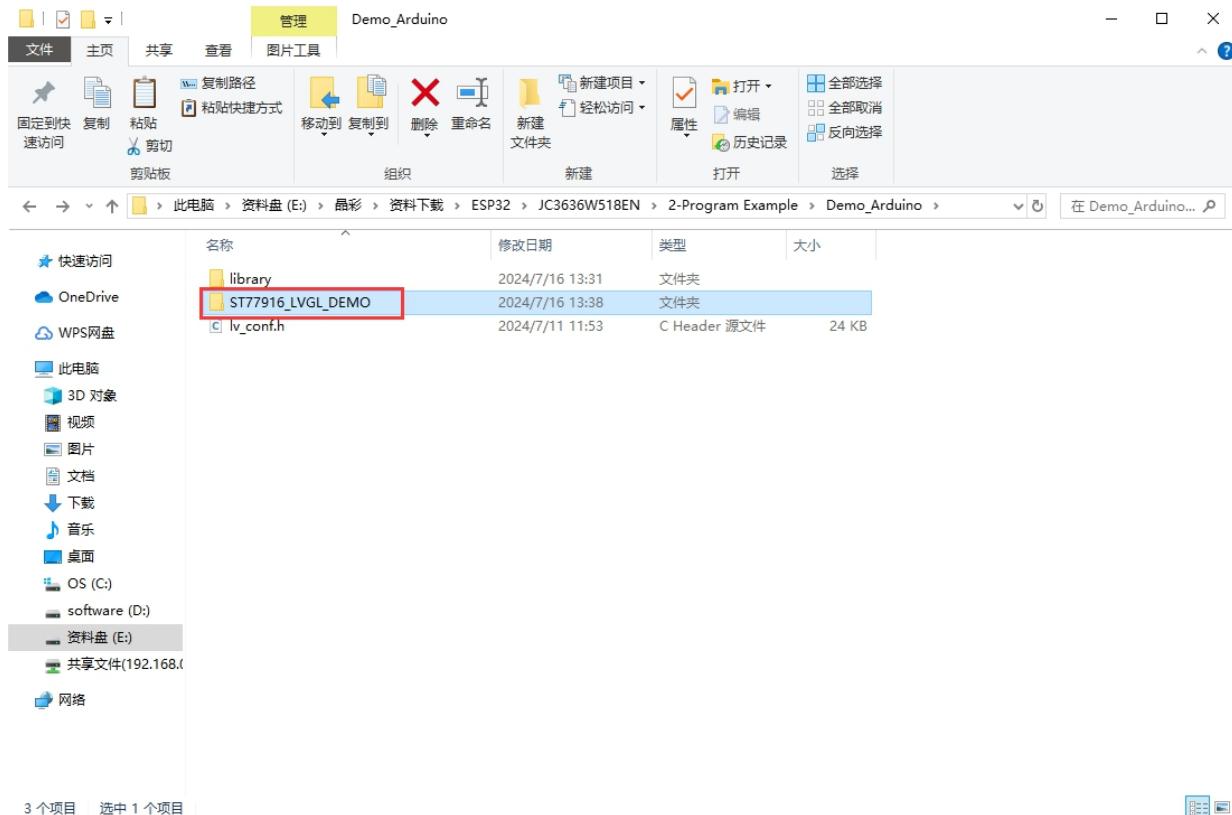


The screenshot shows the Arduino IDE interface with the title bar "LvglWidgets | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. The Tools menu is open, displaying various options like Auto Format, Archive Sketch, Fix Encoding & Reload, Manage Libraries..., Serial Monitor (which is highlighted with a red box), Serial Plotter, ESP32 Sketch Data Upload, WiFi101 / WiFiNINA Firmware Updater, Board: "ESP32S3 Dev Module", Upload Speed: "921600", USB Mode: "Hardware CDC and JTAG", USB CDC On Boot: "Disabled", USB Firmware MSC On Boot: "Disabled", USB DFU On Boot: "Disabled", Upload Mode: "UART0 / Hardware CDC", CPU Frequency: "240MHz (WiFi)", Flash Mode: "QIO 80MHz", Flash Size: "16MB (128Mb)", Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)", Core Debug Level: "None", PSRAM: "OPI PSRAM", Arduino Runs On: "Core 1", Events Run On: "Core 1", Port, Get Board Info, Programmer: "Esptool", and Burn Bootloader. Below the menu, there is a code editor window containing C++ code related to LvglWidgets. The status bar at the bottom shows "Compiling sketch..." and a progress bar.

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:

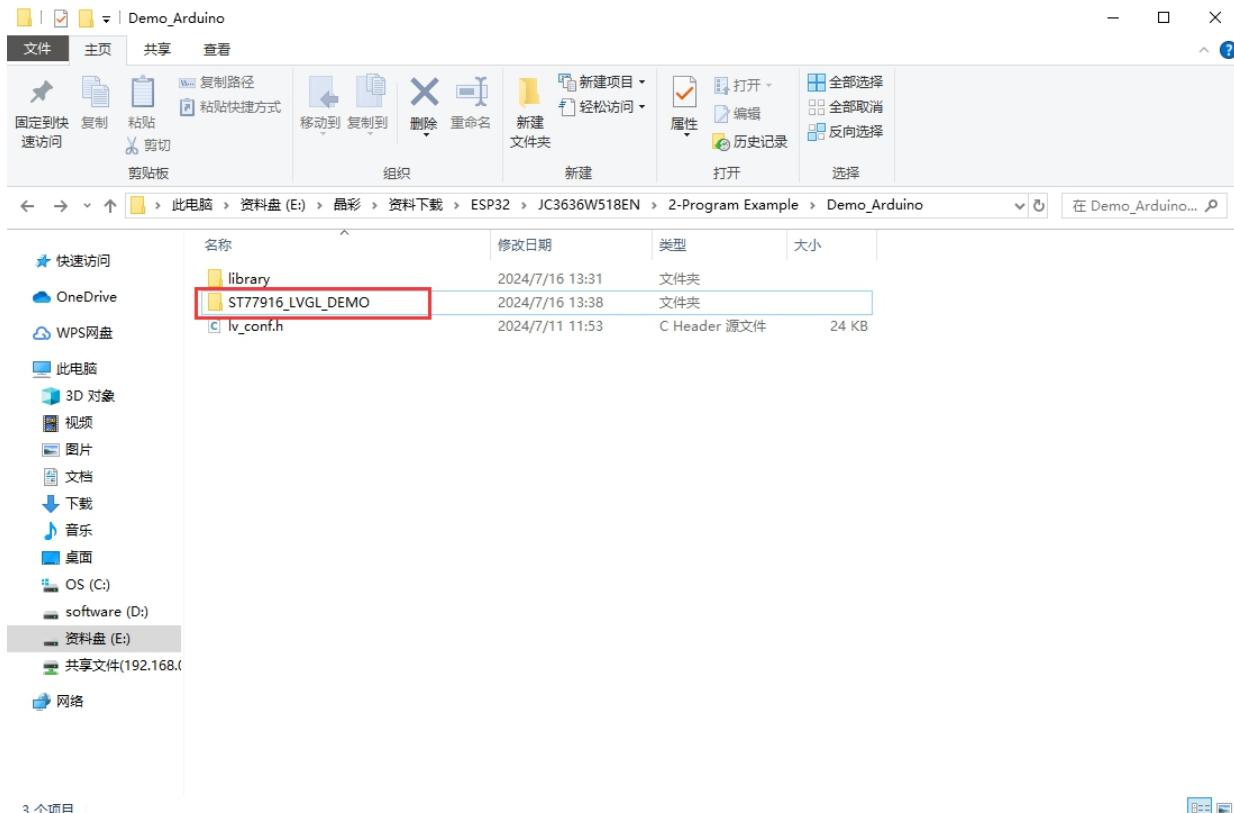


The examples in the red circle are all based on the ESP32_Display_Panel library and ESP32_IO_Expander library and LVGL library.

About the use of touch and LVGL:

Find the data center ST77916_LVGL_DEMO

As shown:



3 个项目

Download three library files .

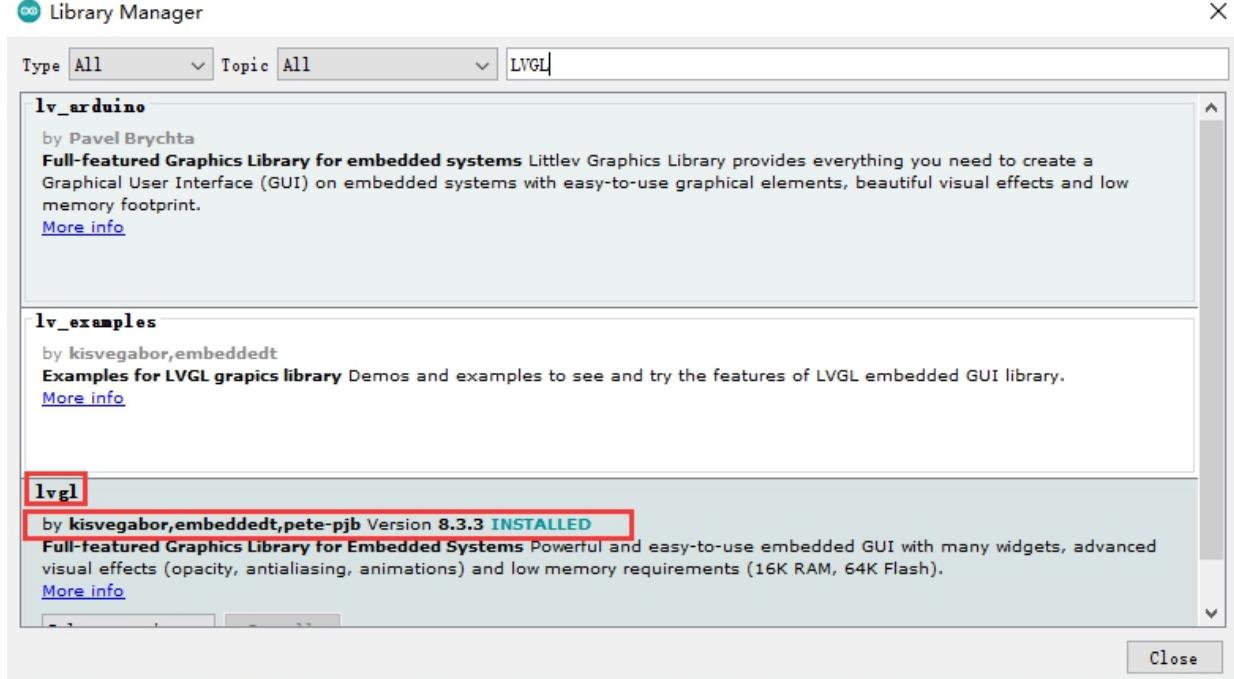
Requires Arduino 3.0.1 and above

Install the ESP32_Screen (0.1.4) library and

ESP32_IO_Expander(0.0.2)library

LVGL installation library 8.4.0 and below

One -Lvgl



Two -ESP32_Display_Panel



Library Manager

Type All Topic All ESP32_Display_Panel

uses DMA for ultra-fast refresh rates and therefore very low CPU usage.

ESP32_Display_Panel
by espressif Version 0.1.4 **INSTALLED**
ESP32_Display_Panel is an Arduino library designed for ESP SoCs to drive display panels and facilitate rapid GUI development. Currently support boards: ESP32-C3-LCDkit, ESP32-S3-Box, ESP32-S3-Box-3, ESP32-S3-Box-3(beta), ESP32-S3-Box-Lite, ESP32-S3-EYE, ESP32-S3-Korvc devices: Bus, LCD, Touch, Backlight, IO expander. Currently supported Bus: I2C, SPI, QSPI, 3-wire SPI + RGB. Currently supported LCDs: CST816S, FT5x06, GT1151, GT911, ST7123, TT21100, XPT2046.

GFX Library for Arduino
by Moon On Our Nation Version 1.2.9 **INSTALLED**
Arduino_GFX is a GFX library for various color displays with various data bus interfaces. Arduino_GFX is a Arduino graphics library. Currently support GC9A01 round display, GC9106, GC9107, HX8347C, HX8347D, HX8352C, HX8357A, HX8357B, HX8369A, ILI6122, ILI9225, ILI9331, ILI9341, ILI9342(M5Stack, ESP32-S3-BOX), ILI9481, ILI9486, ILI9488, ILI9806, JBT6K71, NT35310, NT35510, NT39125, NV3041A, OTM8009A, R61529, RM67162, SEPPS525, SSD1283A, SSD1331, SSD1351, ST7735, ST7789, ST7796 and virtually all Raspberry Pi DPI (RGB) display. Tested RGB display: GC9503V, ILI6485, ST7262, ST7701. Currently support software SPI (8-bit and 9-bit), hardware SPI (8-bit, ESP32 also support 9-bit), 8-bit parallel interface(AVR, ESP32, RPi Pico, RTL8720, STM32), 16-bit parallel interface(ESP32 and RPi Pico) and RGB Panel

Close

Three -ESP32_IO_Expander

Library Manager

Type All Topic All ESP32_IO_Expander

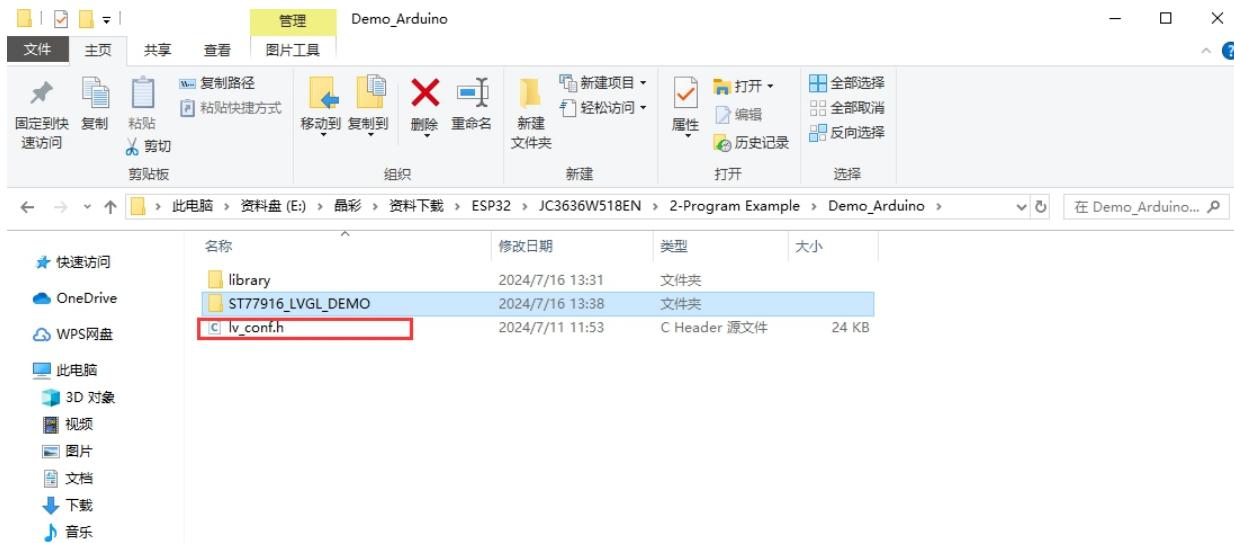
ESP32_Display_Panel
by espressif Version 0.1.4 **INSTALLED**
ESP32_Display_Panel is an Arduino library designed for ESP SoCs to drive display panels and facilitate rapid GUI development. Currently support boards: ESP32-C3-LCDkit, ESP32-S3-Box, ESP32-S3-Box-3, ESP32-S3-Box-3(beta), ESP32-S3-Box-Lite, ESP32-S3-EYE, ESP32-S3-Korvc devices: Bus, LCD, Touch, Backlight, IO expander. Currently supported Bus: I2C, SPI, QSPI, 3-wire SPI + RGB. Currently supported LCDs: CST816S, FT5x06, GT1151, GT911, ST7123, TT21100, XPT2046.

ESP32_IO_Expander
by espressif Version 0.0.2 **INSTALLED**
ESP32_IO_Expander is a library designed for driving IO expander chips using ESP32 SoCs. Currently support TCA95xx(8bit), TCA95xx(16bit), HT8574, CH422G

More info Select version Install Update Close

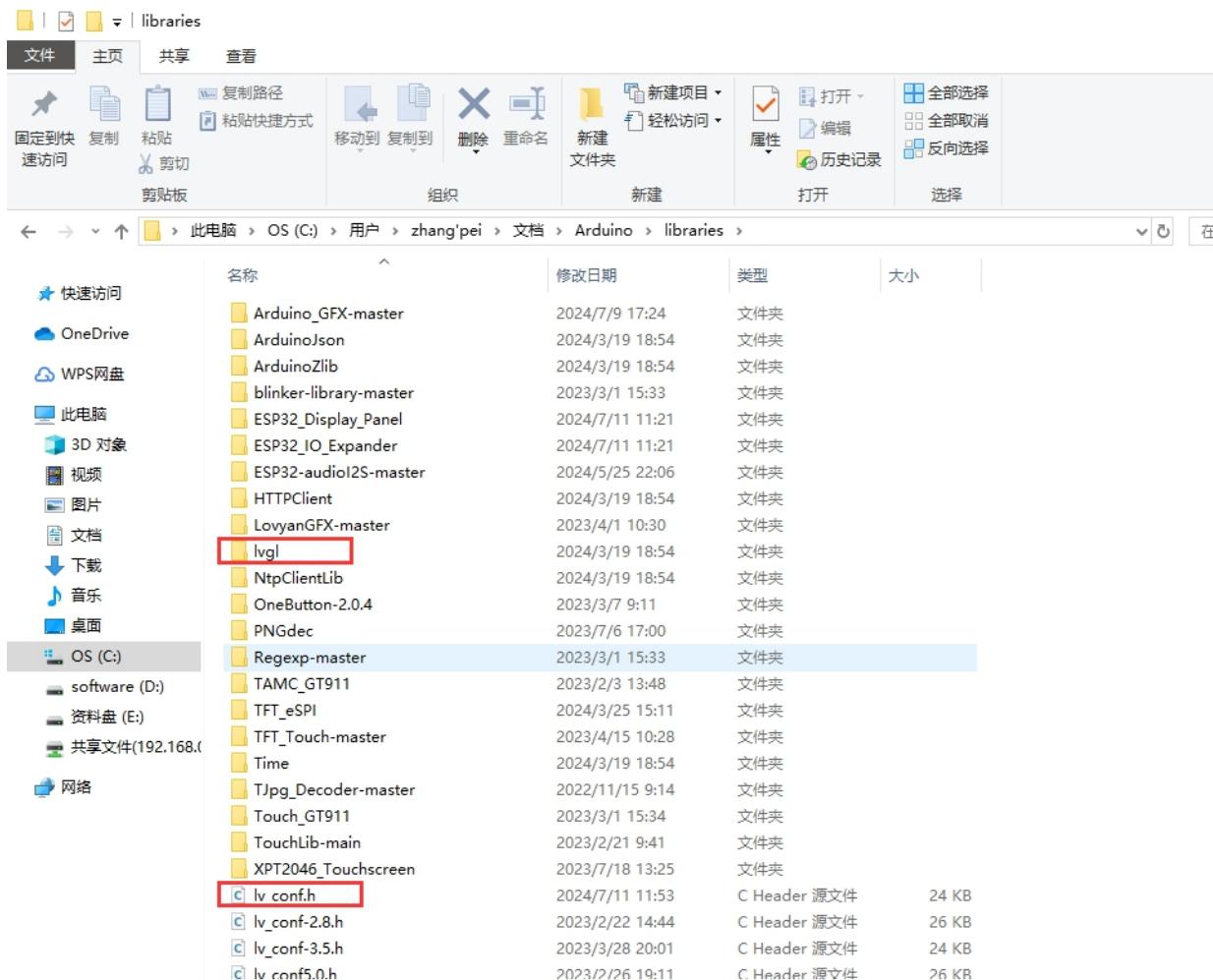
Copy the lv_conf.h of the data center .

As shown:



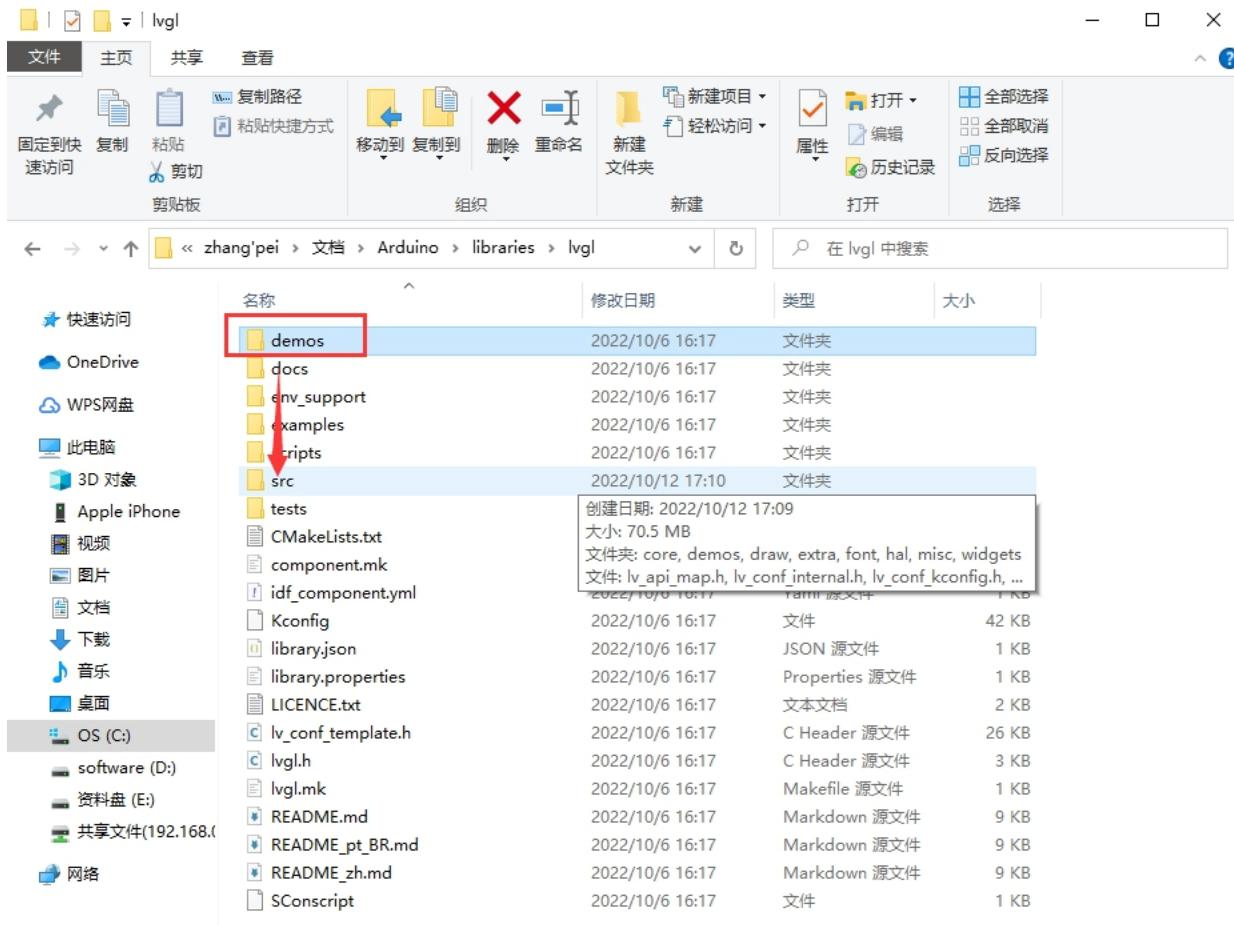
Put this file under the arduino library file, it must be in the same root directory as the library lvgl .

As shown:



Three-Lvgl demos The file is copied to the SRC folder

As shown:



After compiling, you can run LVGL and touch normally.