



# 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.



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'pei\Documents\Arduino\libraries\Touch\_test: no headers files (.h) found in C:\U  
Invalid library found in C:\Users\zhang'pei\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.



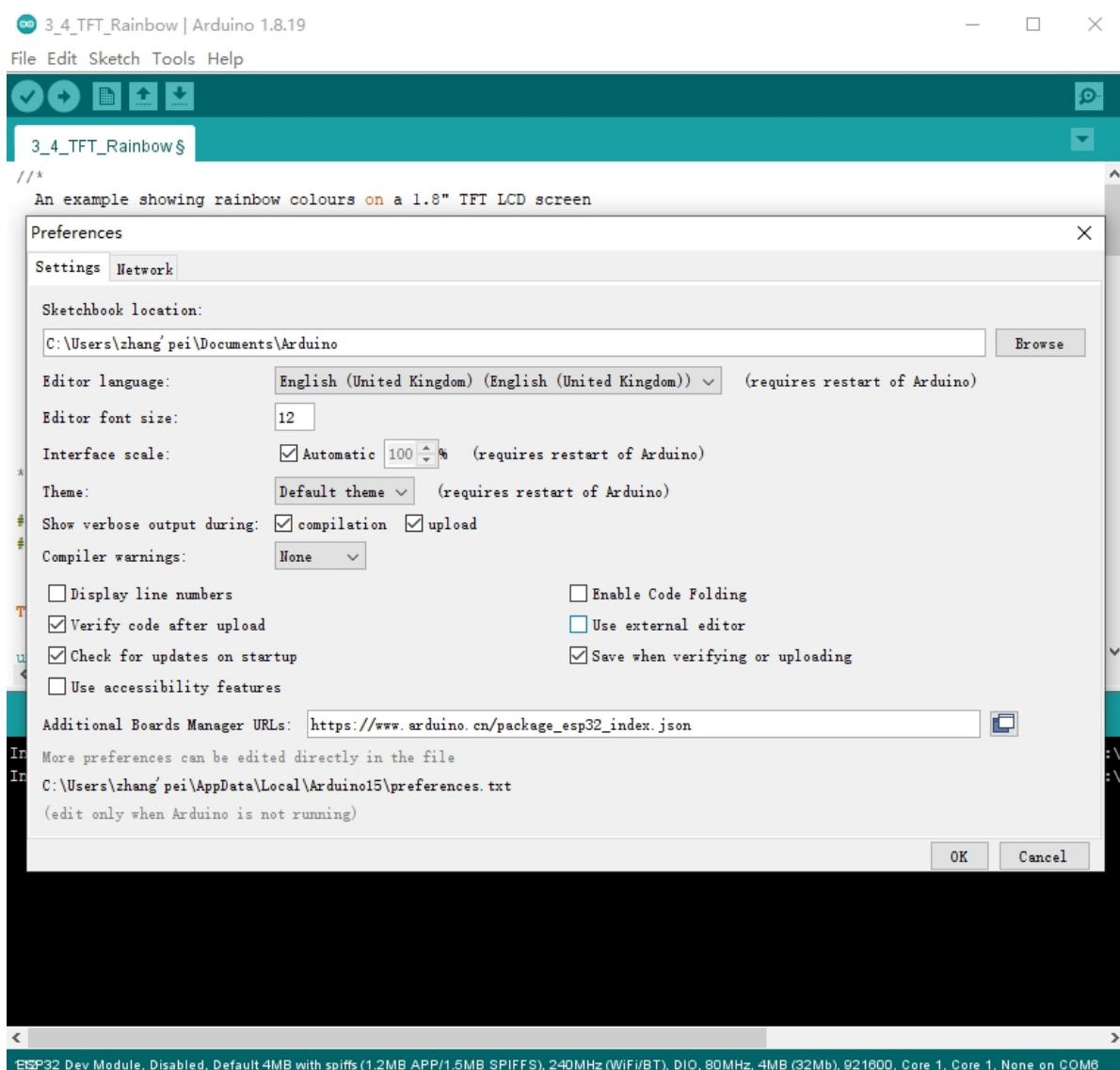
The screenshot shows the Arduino IDE interface with the title bar "3\_4\_TFT\_Rainbow | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. A context menu is open over some code, with options like New, Open..., Open Recent, Sketchbook, Examples, Close, Save, Save As..., Page Setup, 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](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)  
Development release link:

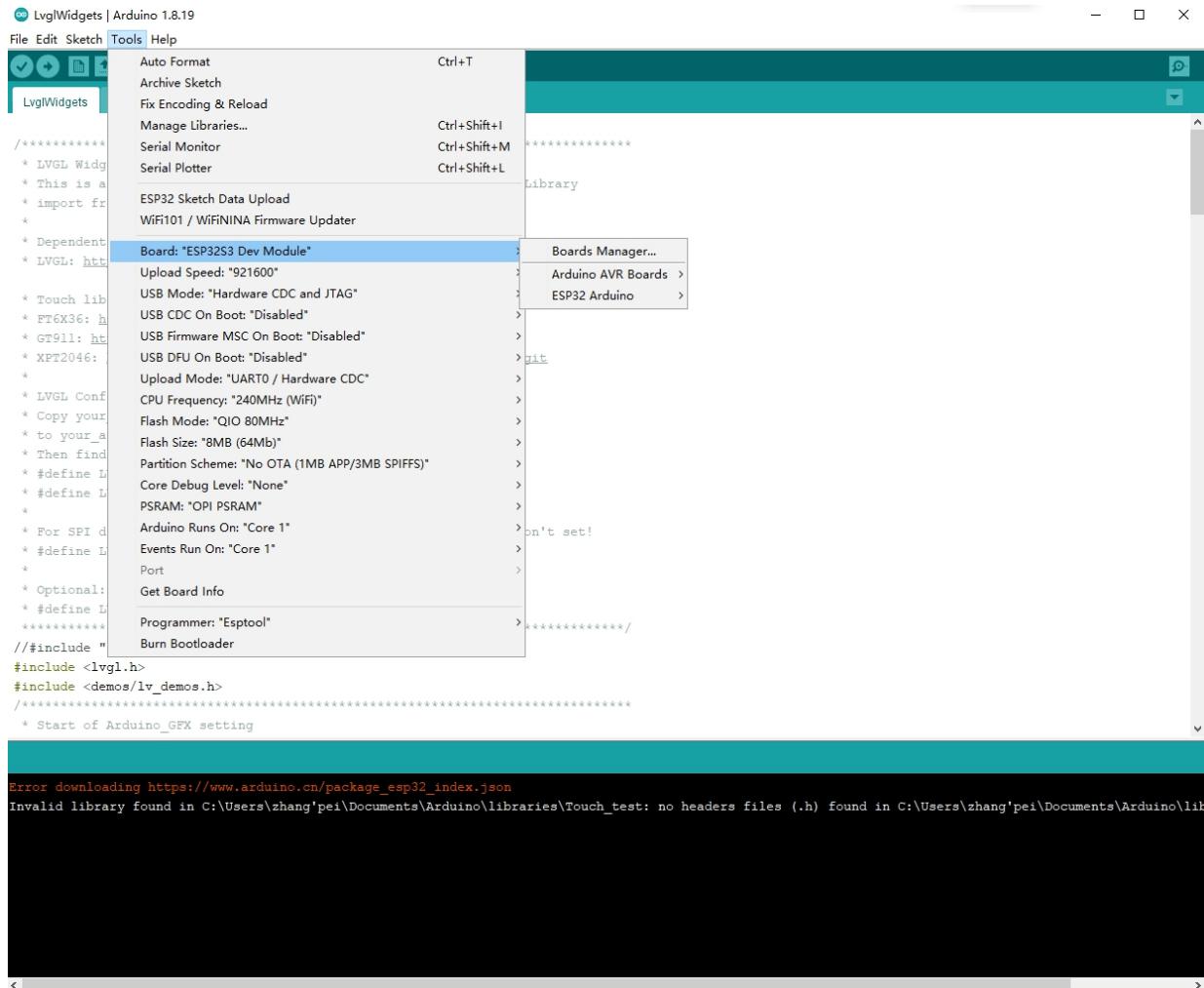
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)

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

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

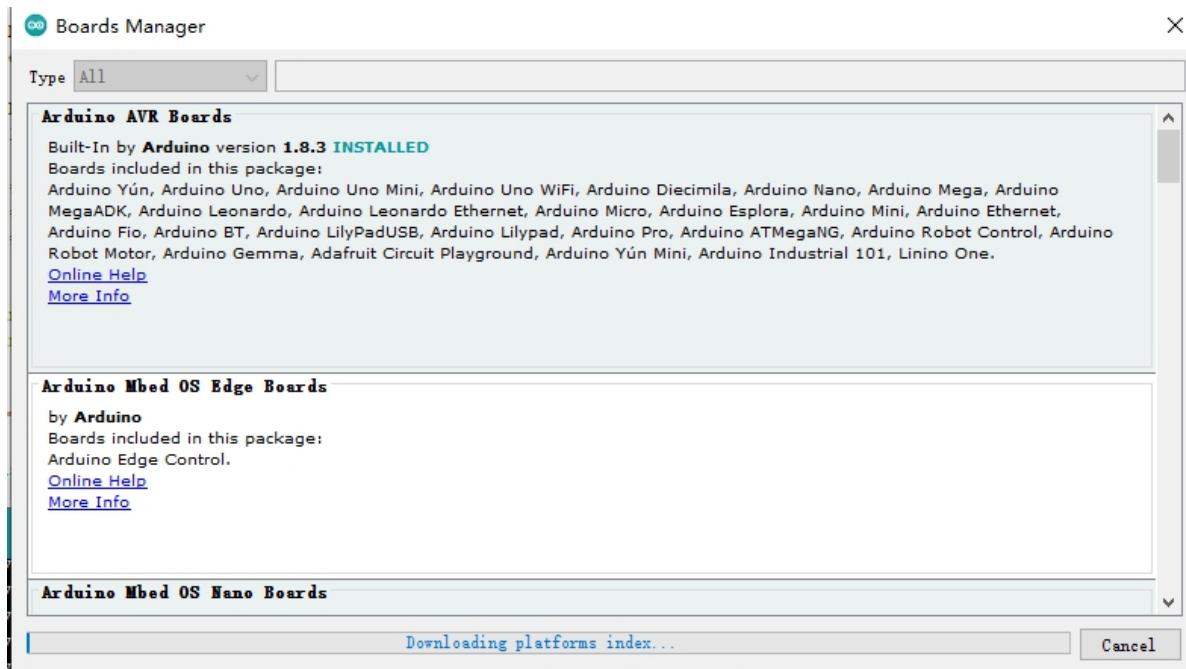


- (9) In the Arduino IDE click on the Tools menu on the top menu bar.
- (10) Scroll down to the Board: entry
- (11) A submenu will open when you highlight the Board: entry.
- (12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.
- (13) In the search box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

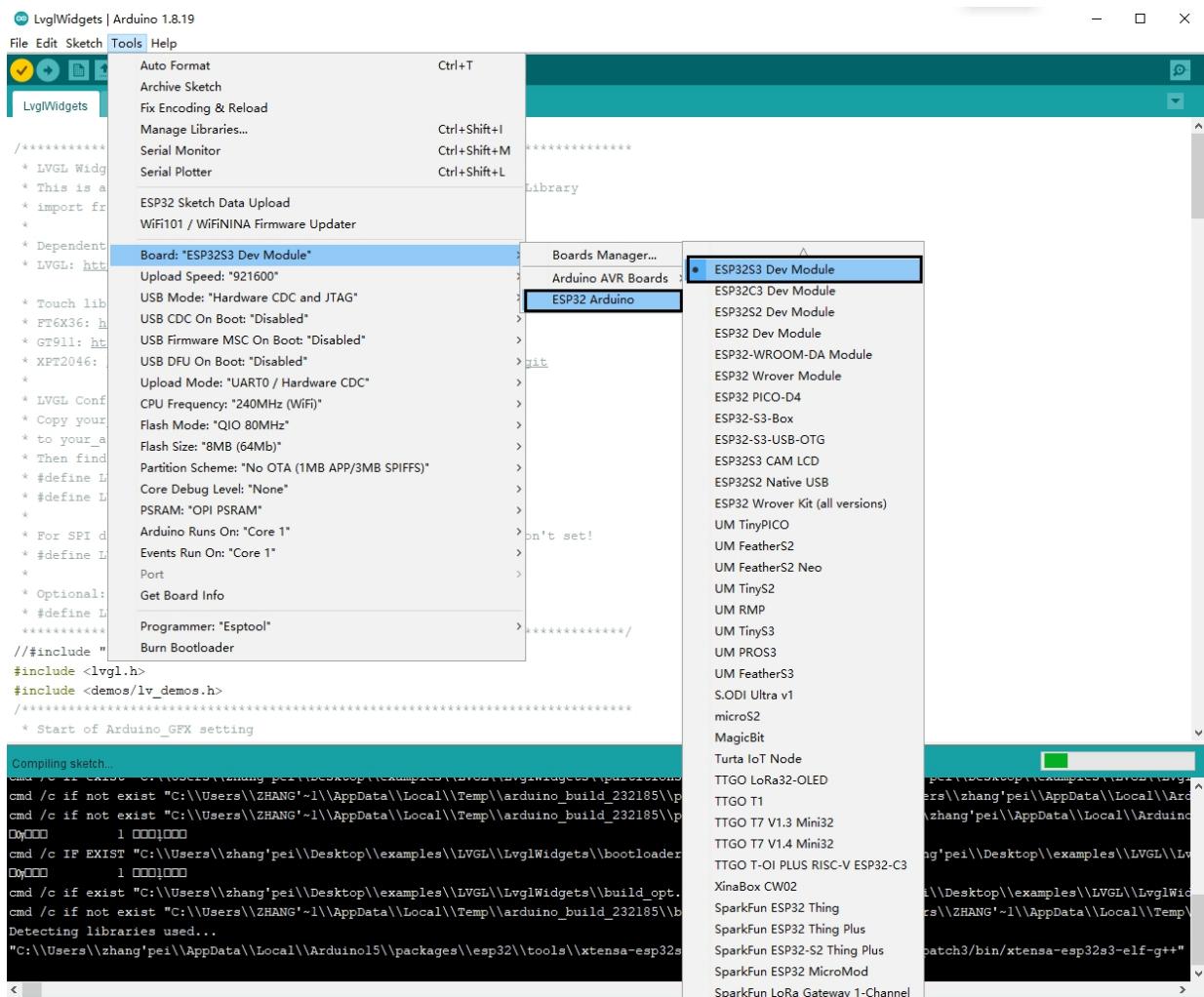


(14) You should see an entry for “esp32 by Espressif Systems”. Highlight this entry and click on the Install button.

This will install the ESP32 boards into your Arduino IDE

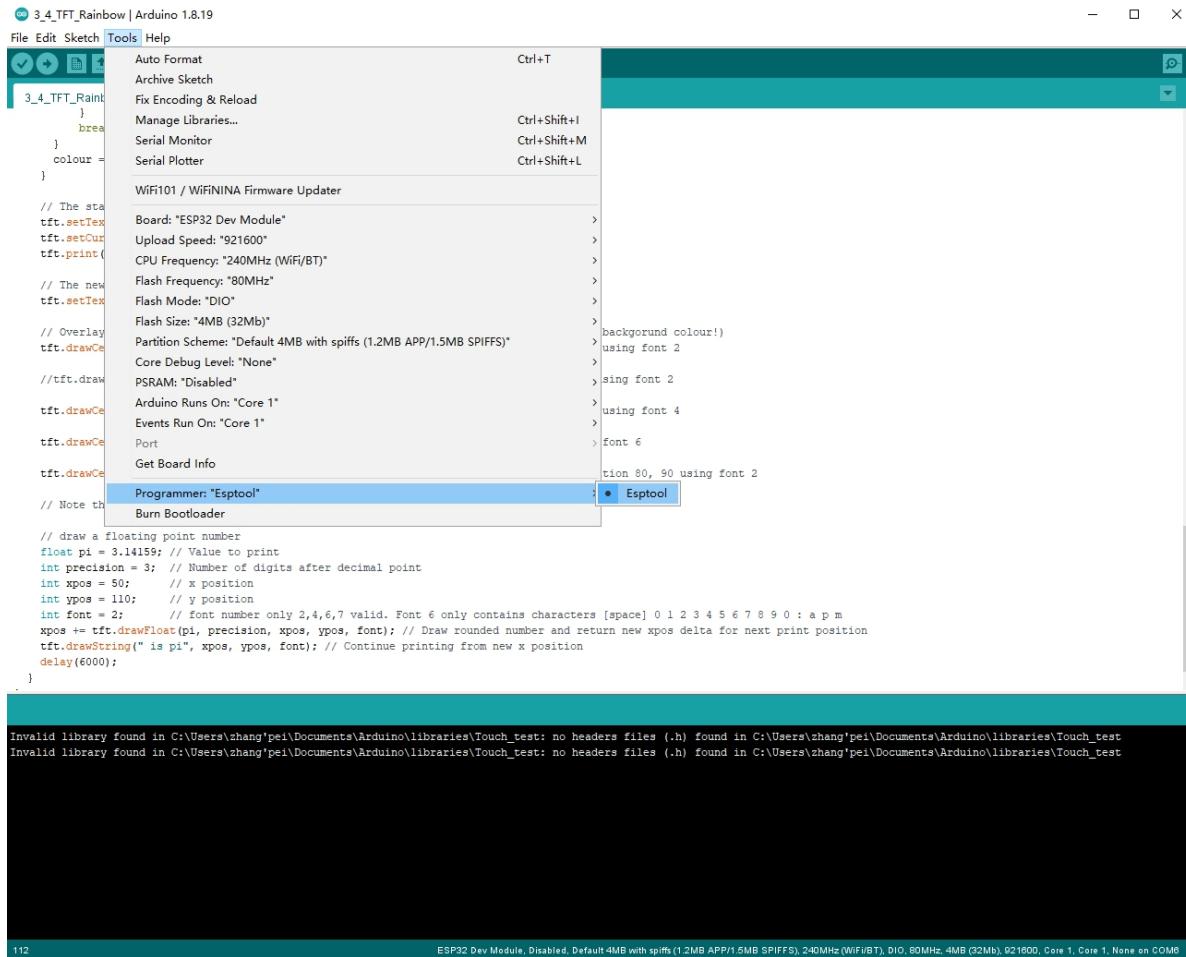


Once the installation completes, we need to select the correct board options for the "ESP32 Arduino" board. In the board type, in the tools tab, we choose "ESP32S3 Dev Module".





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.



LvglWidgets | Arduino 1.8.19

File Edit Sketch Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

**Serial Monitor** Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

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"

Burn Bootloader

```
/* LVGL Widgets
 * This is a
 * import fr
 *
 * Dependent
 * LVGL: http://lvgl体化
 * Touch lib
 * FT6X36: h
 * GT911: ht
 * XPT2046:
 *
 * LVGL Conf
 * Copy your
 * to your_a
 * Then find
 * #define L
 * #define L
 *
 * For SPI d
 * #define L
 *
 * Optional:
 * #define L
 */
#ifndef include
#include <lvgl.h>
#include <demos/lv_demos.h>
#endif
/*
 * Start of Arduino_GFX setting
 */
Compiling sketch...
C:\Users\zhang\peil\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.2.0\cores\esp32\gcc\gcc8_4_0-esp-2021r2-patch3\bin\xtensa-esp32s3-elf-g+++
"C:\Users\zhang\peil\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32s3-elf-gcc\gcc8_4_0-esp-2021r2-patch3\bin\xtensa-esp32s3-elf-g++"
"C:\Users\zhang\peil\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32s3-elf-gcc\gcc8_4_0-esp-2021r2-patch3\bin\xtensa-esp32s3-elf-g++
```

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 screenshot shows a Windows File Explorer window with the following details:

Path: Demo\_Arduino > Demo\_Arduino

File Explorer ribbon tabs: 文件 (File), 主页 (Home), 共享 (Share), 查看 (View)

Toolbar icons (from left to right): 固定到快 (Pin to Start), 复制 (Copy), 粘贴 (Paste), 移动到 (Move to), 复制到 (Copy to), 删除 (Delete), 重命名 (Rename), 新建文件夹 (New folder), 新建 (New), 属性 (Properties), 打开 (Open), 编辑 (Edit), 全部选择 (Select All), 全部取消 (Cancel All), 历史记录 (History), 反向选择 (Invert Selection), 选择 (Select)

Search bar: 在 Demo\_Arduino 中搜索 (Search in Demo\_Arduino)

名称 (Name)	修改日期 (Modified Date)	类型 (Type)
3_1_Helloworld	2022/9/27 14:27	文件夹 (Folder)
3_2_uart	2022/9/27 14:27	文件夹 (Folder)
3_3-1_TFT_HelloWorld	2022/9/30 18:20	文件夹 (Folder)
3_3-2_TFT-CLOCK	2022/9/30 18:29	文件夹 (Folder)
3_3-3_TFT_PDQgraphicstest	2022/9/30 16:44	文件夹 (Folder)
3_3-3-TFT-LVGL-Benchmark	2022/9/30 18:07	文件夹 (Folder)
3_3-4_TFT-LVGL-Widgets	2022/9/30 18:07	文件夹 (Folder)
4_1_wifi_AP	2022/9/27 14:27	文件夹 (Folder)
4_2_wifi_STA	2022/9/27 14:27	文件夹 (Folder)
4_3_wifi_SmartConfig	2022/9/27 14:27	文件夹 (Folder)
4_4_wifi_STA_TCP_Server	2022/9/27 14:27	文件夹 (Folder)
4_5_WIFI_STA_TCP_Client	2022/9/27 14:27	文件夹 (Folder)
4_6_WIFI_STA_UDP	2022/9/27 14:27	文件夹 (Folder)
4_7_WIFI Web Servers LED	2022/9/27 14:27	文件夹 (Folder)
4_8_WIFI Web Servers Relay	2022/9/27 14:27	文件夹 (Folder)
4_9_WIFI Web Servers DHT11	2022/9/27 14:27	文件夹 (Folder)
4_10_SmallDesktopDisplay	2022/9/27 14:27	文件夹 (Folder)
5_1_bleService	2022/9/27 14:27	文件夹 (Folder)
libraries	2022/9/30 16:58	文件夹 (Folder)

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 .



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

- Title Bar:** LVGL\_Arduino | Arduino 1.8.19
- File Menu:** File Edit Sketch Tools Help
- Tools Submenu:** Manage Libraries... (highlighted), Auto Format, Archive Sketch, Fix Encoding & Reload.
- Sketch Content:** LVGL\_Arduino sketch code for an ESP32 Dev Module. It includes definitions for static const u8, static lv\_disp\_t lv\_disp, static lv\_color\_t lv\_colc, TFT\_eSPI tft, and void my\_print(). It also defines lv\_examp1() and lv\_examp2() functions.
- Upload Status:** Done uploading. The output window shows the upload progress and completion message: "Writing at 0x000721c7... (71 %)" through "Writing at 0x00099999... (100 %)". It also shows the total bytes written: "Wrote 565088 bytes (331572 compressed) at 0x00010000 in 5.5 seconds (effective 816.4 kbit/s)...".
- Message:** Hash of data verified.
- Message:** Leaving... Hard resetting via RTS pin...
- Library Error:** Invalid library found in C:\Users\zhang\pei\Documents\Arduino\libraries\Touch\_test: no headers files (.h) found in C:\Users\zhang\pei\Documents\Arduino\libraries\Touch\_test



The screenshot shows the Arduino IDE interface. At the top, there's a menu bar with File, Edit, Sketch, Tools, Help, and a toolbar with icons for file operations. Below the menu is a tab bar with 'LVGLWidgets' and 'touch.h'. The main area contains the code for the LVGLWidgets sketch, which includes comments about LVGL Widgets, touch libraries, and dependent libraries. It also includes code for the Arduino\_GFX library and the gen4-IoD library. The bottom part of the IDE shows the terminal output of the compilation process, indicating successful creation of the esp32s3 image.

```
/* LVGL Widgets
 * This is a widgets demo for LVGL - Light and Versatile Graphics Library
 * Import from: https://github.com/lvgl/lv_demos.git
 *
 * Dependent libraries:
 * LVGL: https://github.com/lvgl/lvgl.git
 *
 * Touch libraries:
 * FT6X36: https://github.com/stm
 * GT911: https://github.com/TAMC
 * XPT2046: https://github.com/Pai
 *
 * LVGL Configuration file:
 * Copy your_arduino_path/libraries
 * to your_arduino_path/libraries
 * Then find and set:
 * #define LV_COLOR_DEPTH 16
 * #define LV_TICK_CUSTOM 1
 *
 * For SPI display set color swap
 * #define LV_COLOR_16_SWAP 1
 *
 * Optional: Show CPU usage and F
 * #define LV_USE_PERF_MONITOR 1
 */
#ifndef __lv_demo_widgets_h
#define __lv_demo_widgets_h
#include <lvgl.h>
#include <demos/lv_demos.h>
/* Start of Arduino_GFX setting
 */
Done compiling
Successfully created esp32s3 image.
"C:\Users\zhang'pei\AppData\Local\Arduino15\packages\esp32\hardware\esp32\2.0.3/tools/gen_esp32part.exe" -q "C:\\\\Users\\\\ZHANG'~1\\\\AppData\\\\Local\\Arduino15\\packages\\esp32\\hardware\\esp32\\2.0.3\\libraries\\Arduino_GFX\\master\\Arduino_GFX.h"
Using library lvgl at version 8.3.0-dev in folder: C:\\\\Users\\\\zhang'pei\\\\Documents\\\\Arduino\\\\libraries\\\\lvgl
Using library Arduino_GFX-master at version 1.2.8 in folder: C:\\\\Users\\\\zhang'pei\\\\Documents\\\\Arduino\\\\libraries\\\\Arduino_GFX-master
Using library SPI at version 2.0.0 in folder: C:\\\\Users\\\\zhang'pei\\\\AppData\\\\Local\\\\Arduino15\\\\packages\\\\esp32\\\\hardware\\\\esp32\\\\2.0.3\\\\libraries\\\\SPI
Using library Wire at version 2.0.0 in folder: C:\\\\Users\\\\zhang'pei\\\\AppData\\\\Local\\\\Arduino15\\\\packages\\\\esp32\\\\hardware\\\\esp32\\\\2.0.3\\\\libraries\\\\Wire
Using library gt911-arduino-main at version 1.0.2 in folder: C:\\\\Users\\\\zhang'pei\\\\Documents\\\\Arduino\\\\libraries\\\\gt911-arduino-main
"C:\\\\Users\\\\zhang'pei\\\\AppData\\\\Local\\\\Arduino15\\\\packages\\\\esp32\\\\tools\\\\xtensa-esp32s3-elf-gcc\\\\gcc8_4_0-esp-2021r2-patch3\\\\bin\\\\xtensa-esp32s3-elf-size"
Sketch uses 551605 bytes (52%) of program storage space. Maximum is 1048576 bytes.
Global variables use 65524 bytes (19%) of dynamic memory, leaving 262156 bytes for local variables. Maximum is 327680 bytes.
```

Although the Arduino\_GFX library has many advantages, it may also have a troublesome place for ordinary users, that is, after the installation

#### About the use of PDQgraphicstest:

Find the data center 3\_3-3\_TFT\_PDQgraphicstest

As shown:



文件 主页 共享 查看

固定到快速访问 复制 粘贴 复制路径 粘贴快捷方式 移动到 复制到 删除 重命名 新建文件夹 属性 打开 历史记录 全部 全部 反向 剪切 剪贴板 组织 新建 打开 选

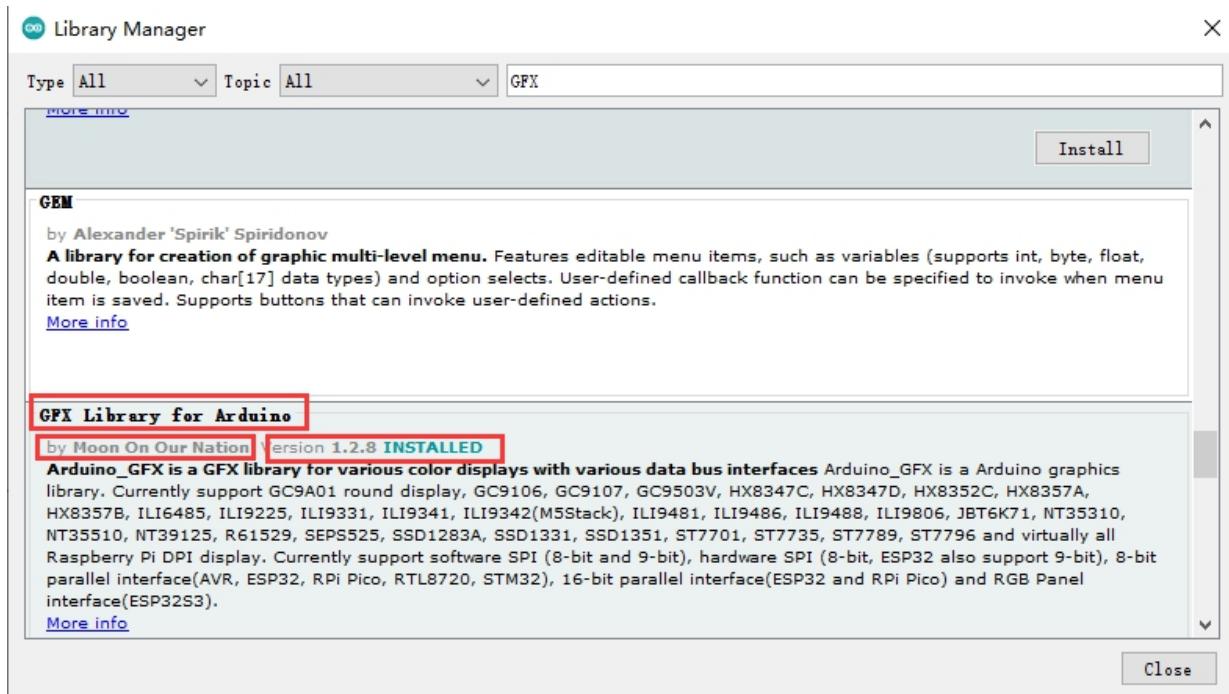
← → ⌂ ⌃ ⌁ ⌂ << 1.9inch\_ESP32-1732S019 >> 1-Demo > Demo\_Arduino ⌂ ⌁ ⌂ 在 De

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

快速访问 OneDrive WPS网盘 此电脑 3D 对象 视频 图片 文档 下载 音乐 桌面 OS (C:) software (D:) 资料盘 (E:) 共享文件(192.168.0.1) 网络

Download library files .

Arduino\_GFX library



After compiling, you can run PDQgraphicstest normally.