



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 ESP32-8048S050 module, from JCZN .

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

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 C++ code related to TFT displays, including calls to tft.setTextColor() and tft.setTextSize(). Below the code, the status bar displays "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".

```
reen << 5 | blue;
font still works as before
JK);
suit font!");

do not use the .setCursor call, coords are embedded
CK, TFT_BLACK); // Do not plot the background colour

// Overlay the black text on top of the rainbow plot (the advantage of not drawing the backgorund colour!)
tft.drawString("Font size 2", 80, 14, 2); // Draw text centre at position 80, 12 using font 2

//tft.drawString("Font size 2",81,12,2); // Draw text centre at position 80, 12 using font 2

tft.drawString("Font size 4", 80, 30, 4); // Draw text centre at position 80, 24 using font 4

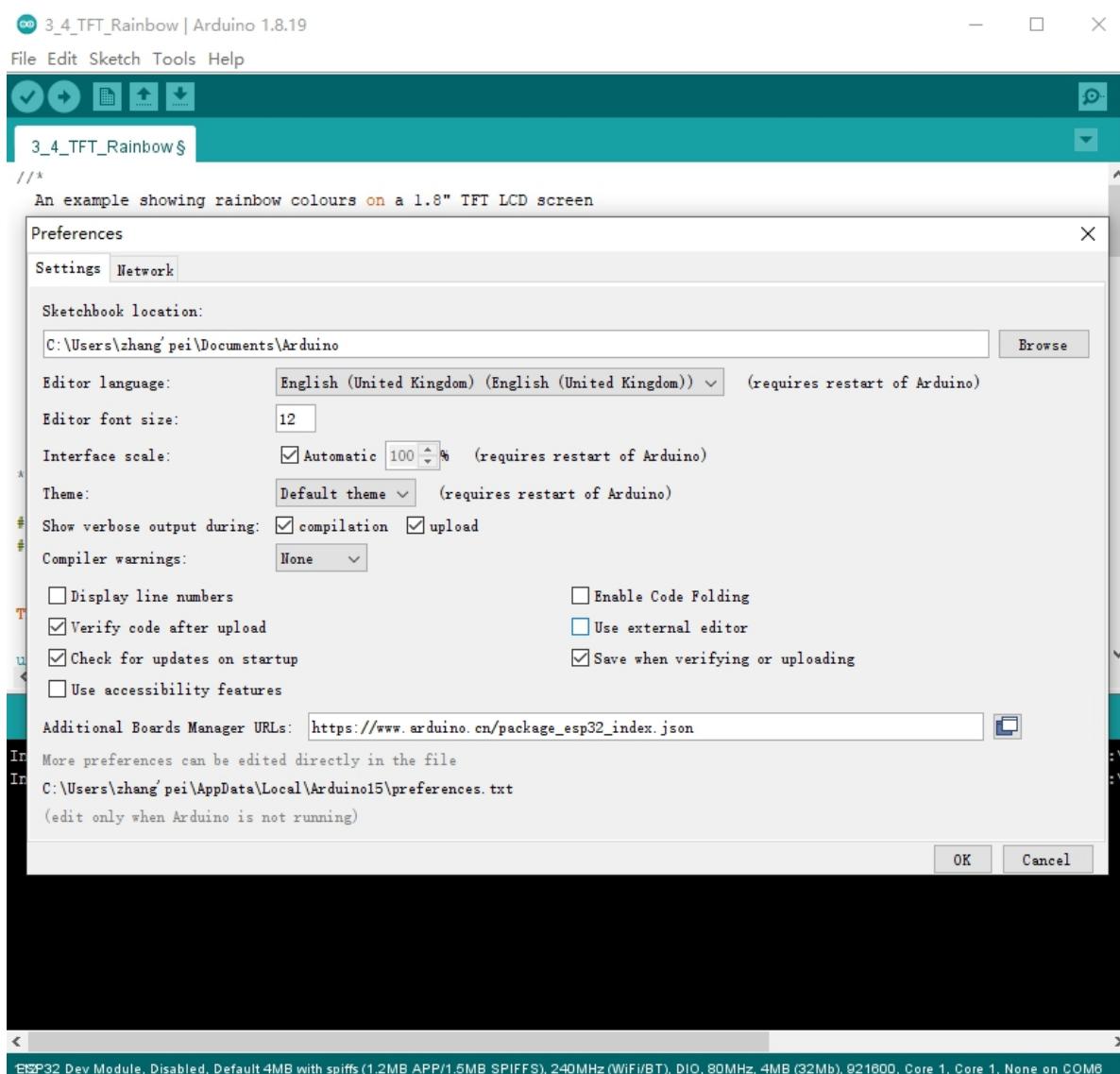
tft.drawString("12.34", 80, 54, 6); // Draw text centre at position 80, 24 using font 6
```

- (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:

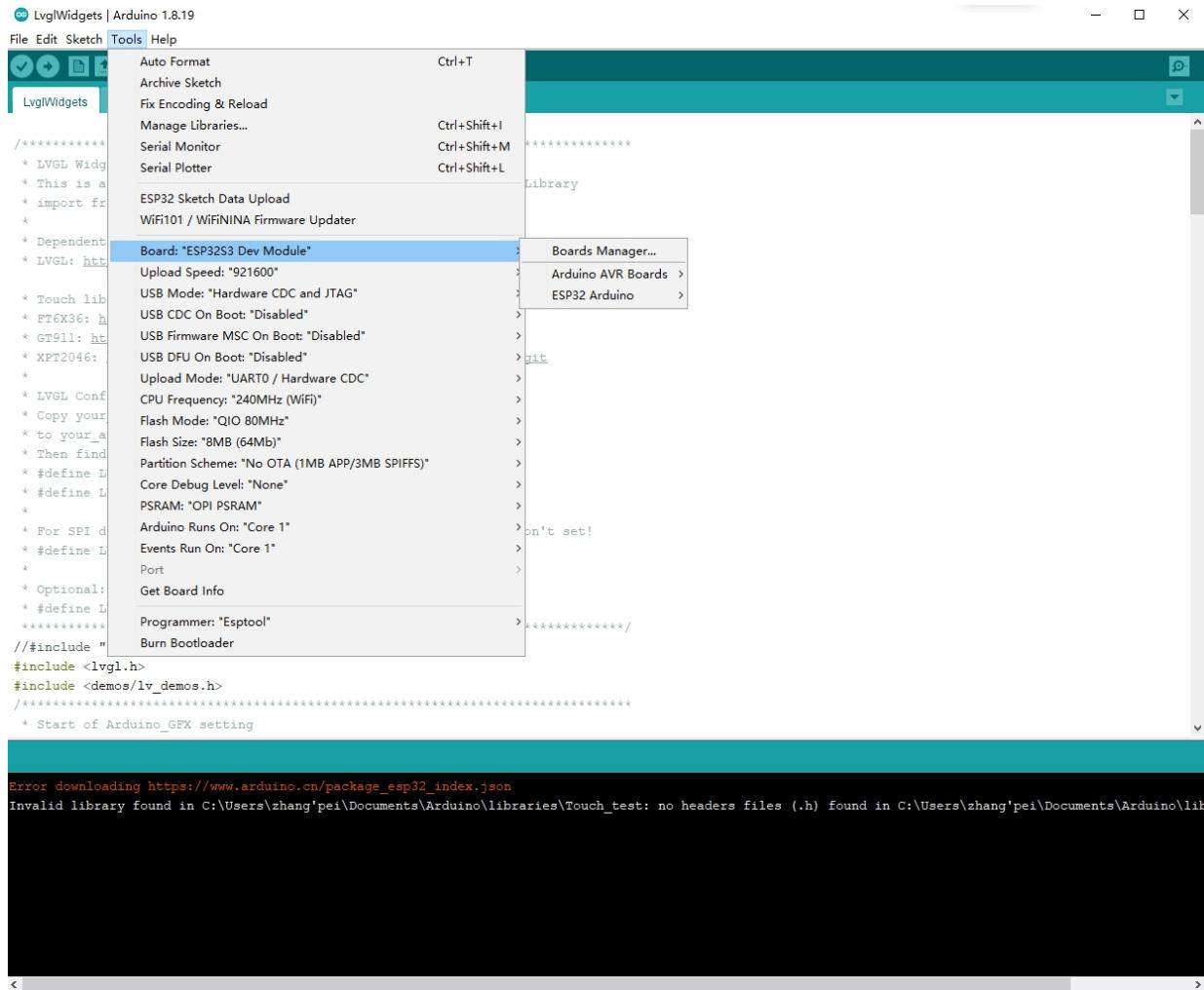
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json

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

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

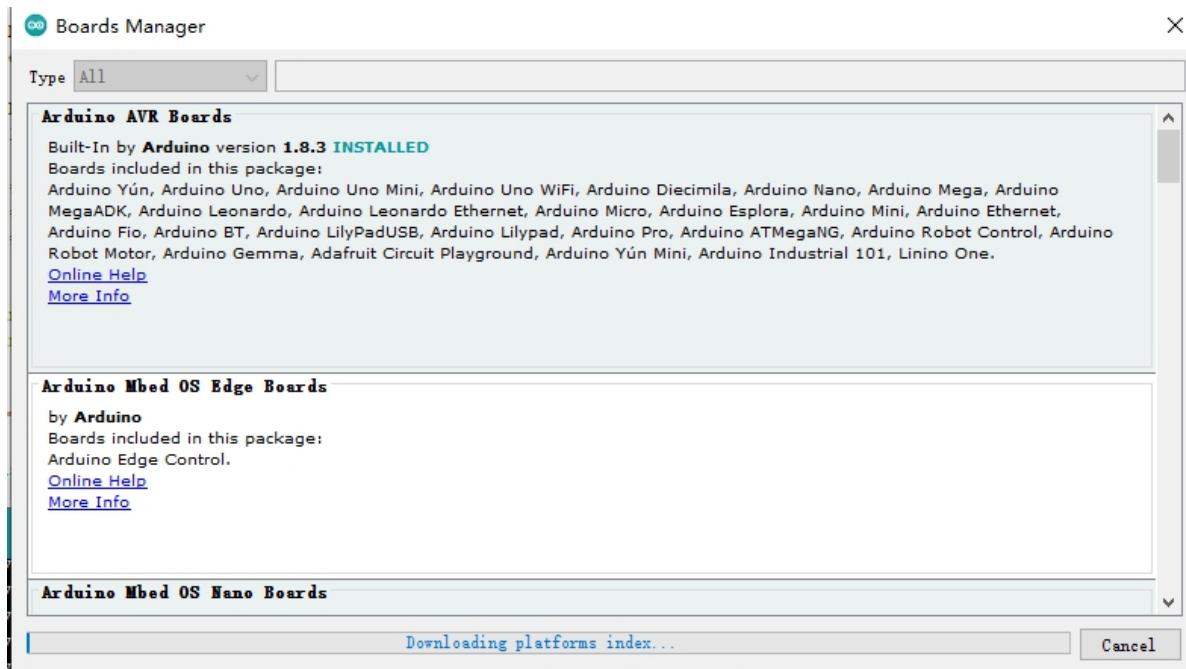


- (9) In the Arduino IDE click on the Tools menu on the top menu bar.
- (10) Scroll down to the Board: entry
- (11) A submenu will open when you highlight the Board: entry.
- (12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.
- (13) In the search box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

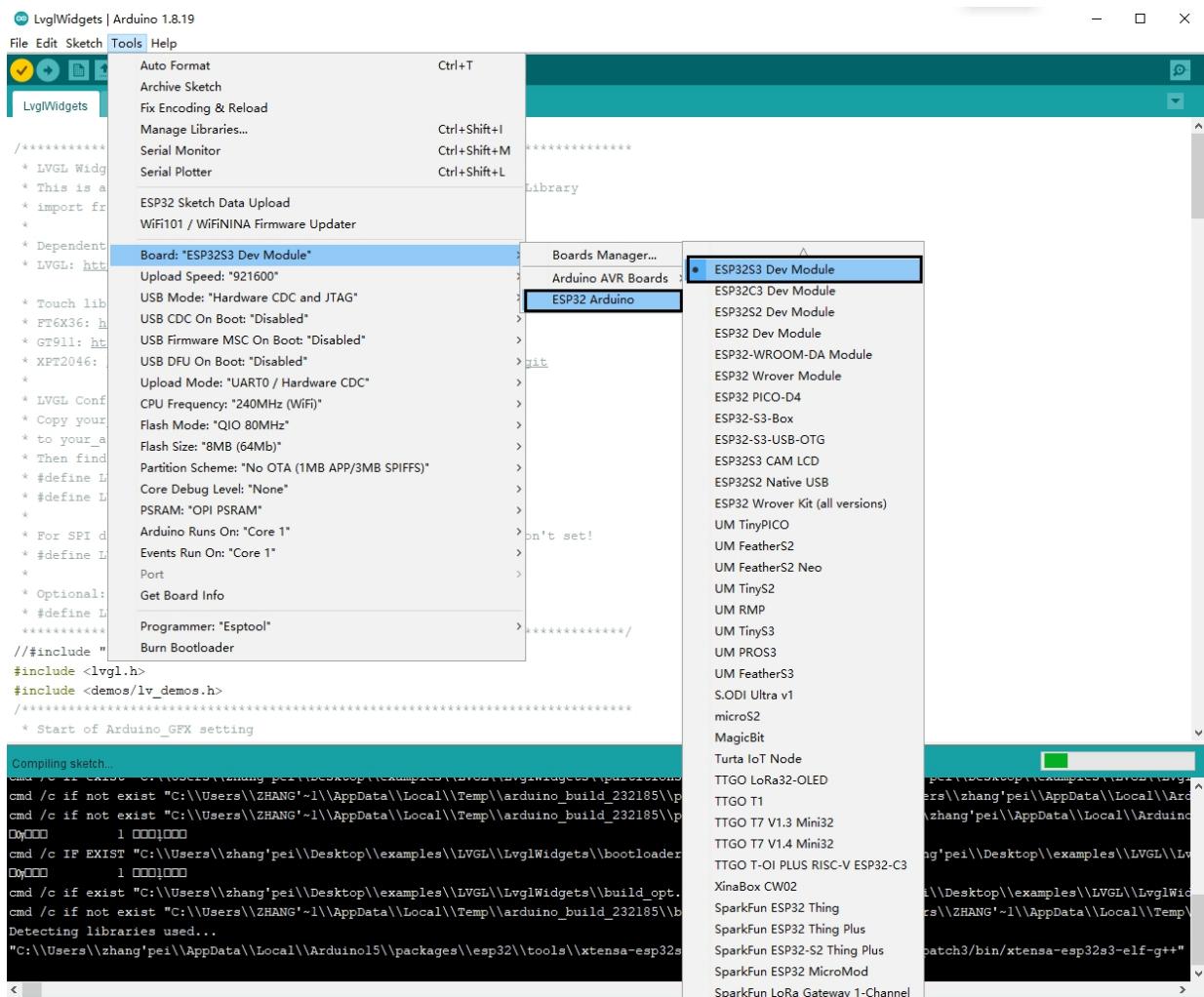


(14) You should see an entry for “esp32 by Espressif Systems”. Highlight this entry and click on the Install button.

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".



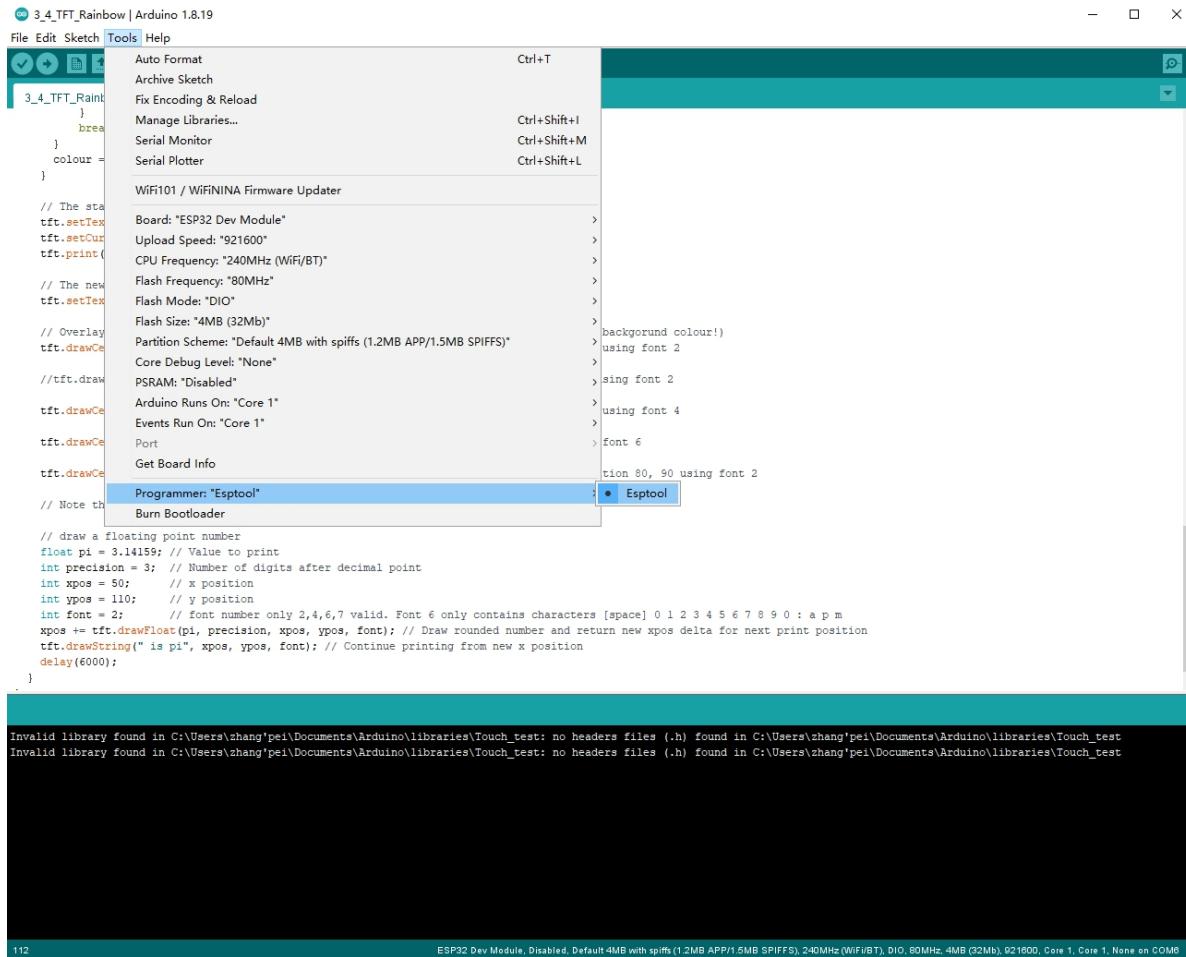


The screenshot shows the Arduino IDE interface with the title bar "LvglWidgets | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. The Tools menu is open, showing various options for sketch upload and board configuration. A red box highlights the "Programmer" option under the "Sketch" section, which is set to "esptool". Other options like "Serial Monitor" and "Serial Plotter" are also visible.

```
/* LVGL Widgets Sketch for ESP32 Dev Module
 * This is a simple example demonstrating how to use the LVGL library with the ESP32.
 * It includes basic touch handling and a scrollable list view.
 */
#include <lvgl.h>
#include <demos/lv_demos.h>
/* Start of Arduino GFX setting */

Compiling sketch
C:\Users\zhang\peil\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.0\cores\esp32\cli.g++ -c "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++
```

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

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

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 .



```
LVGL_Arduino | Arduino 1.8.19
File Edit Sketch Tools Help
LVGL_Arduino
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 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module" ...
Upload Speed: "921600" ...
CPU Frequency: "240MHz (WiFi/BT)" ...
Flash Frequency: "80MHz" ...
Flash Mode: "DIO" ...
Flash Size: "4MB (32Mb)" ...
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ...
Core Debug Level: "None" ...
PSRAM: "Disabled" ...
Arduino Runs On: "Core 1" ...
Events Run On: "Core 1" ...
Port: "COM6" ...
Get Board Info ...
Programmer: "Esptool" ...
Burn Bootloader ...
LV_STYLE_TRANSFORM_WIDTH, LV_STYLE_TRANSFORM_HEIGHT, LV_STYLE_TEXT_LETTER_SPACE);

/*Transition descriptor when going back to the default state.
 *Add some delay to be sure the press transition is visible even if the press was very short*/
static lv_style_transition_dsc_t transition_dsc_def;
lv_style_transition_dsc_init(&transition_dsc_def, props, lv_anim_path_overshoot, 250, 100, NULL);

/*Transition descriptor when going to pressed state.
 *No delay, go to presses state immediately*/
Done uploading.
Writing at 0x000721c7... (71 %)
Writing at 0x00077b55... (76 %)
Writing at 0x0007a03b... (80 %)
Writing at 0x00085715... (85 %)
Writing at 0x0008d8a9... (90 %)
Writing at 0x0009323e... (95 %)
Writing at 0x00099999... (100 %)
Wrote 565088 bytes (331572 compressed) at 0x00010000 in 5.5 seconds (effective 816.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
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 LVGL Widgets sketch. The code includes comments about LVGL Widgets, its purpose, imports, dependent libraries, touch libraries, and configuration. It also includes code for SPI display color swap and optional CPU usage monitoring. The code ends with '#include "lv_demo_widgets.h"' and '#include <lvgl.h>'.

Below the code, the terminal window shows the compilation process:

```
Done compiling
Successfully created esp32s3s image.
"C:\Users\zhang\pe1\AppData\Local\Arduino15\packages\esp32\hardware\esp32\2.0.3\tools\gen_esp32part.exe" -q "C:\\\\Users\\\\ZHANG'~1\\\\AppData\\\\Local\\\\Arduino\\\\packages\\\\esp32\\\\hardware\\\\esp32\\\\2.0.3\\\\libraries\\\\Wire\\\\Wire.h"
Using library lvgl at version 8.3.0-dev in folder: C:\\\\Users\\\\zhang\\\\pe1\\\\Documents\\\\Arduino\\\\libraries\\\\lvgl
Using library Arduino_GFX-master at version 1.2.8 in folder: C:\\\\Users\\\\zhang\\\\pe1\\\\Documents\\\\Arduino\\\\libraries\\\\Arduino_GFX-master
Using library SPI at version 2.0.0 in folder: C:\\\\Users\\\\zhang\\\\pe1\\\\AppData\\\\Local\\\\Arduino\\\\15\\\\packages\\\\esp32\\\\hardware\\\\esp32\\\\2.0.3\\\\libraries\\\\SPI
Using library Wire at version 2.0.0 in folder: C:\\\\Users\\\\zhang\\\\pe1\\\\AppData\\\\Local\\\\Arduino\\\\15\\\\packages\\\\esp32\\\\hardware\\\\esp32\\\\2.0.3\\\\libraries\\\\Wire
Using library gt911-arduino-main at version 1.0.2 in folder: C:\\\\Users\\\\zhang\\\\pe1\\\\Documents\\\\Arduino\\\\libraries\\\\gt911-arduino-main
"C:\\\\Users\\\\zhang\\\\pe1\\\\AppData\\\\Local\\\\Arduino\\\\15\\\\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 touch and LVGL:

Find the data center 3_3-4_TFT-LVGL-Widgets

As shown:



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

- Address Bar:** < 1-Demo > Demo_Arduino >
- Search Bar:** 在 Demo_Arduino 中搜索
- Left Sidebar:** 快速访问, OneDrive, WPS网盘, 此电脑, 3D 对象, Apple iPhone, 视频, 图片, 文档, 下载, 音乐, 桌面, OS (C:), software (D:), 资料盘 (E:), 共享文件(192.168.0.100)
- Content Area:** 显示了多个文件夹，其中 '3_3-4_TFT-LVGL-Widgets' 文件夹被红色框选。

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

Download two library files .

One -Arduino_GFX library

The screenshot shows the Arduino Library Manager interface with the following details:

- Search Bar:** Type: All, Topic: All, Search term: GFX
- Library List:**
 - GEM** by Alexander 'Spirik' Spiridonov: A library for creation of graphic multi-level menu. Features editable menu items, such as variables (supports int, byte, float, double, boolean, char[17] data types) and option selects. User-defined callback function can be specified to invoke when menu item is saved. Supports buttons that can invoke user-defined actions.
[More info](#)
 - GFX Library for Arduino** by Moon On Our Nation, Version 1.2.8 INSTALLED: Arduino_GFX is a GFX library for various color displays with various data bus interfaces. Arduino_GFX is an Arduino graphics library. Currently support GC9A01 round display, GC9106, GC9107, GC9503V, HX8347C, HX8347D, HX8352C, HX8357A, HX8357B, ILI6485, ILI9225, ILI9331, ILI9341, ILI9342(M5Stack), ILI9481, ILI9486, ILI9488, ILI9806, JBT6K71, NT35310, NT35510, NT39125, R61529, SEPS525, SSD1283A, SSD1331, SSD1351, ST7701, ST7735, ST7789, ST7796 and virtually all Raspberry Pi SPI display. 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 interface(ESP32S3).
[More info](#)
- Buttons:** Install, Close

Two -Lvgl



Library Manager

Type: All Topic: All LVGL

lv_arduino
by Pavel Brychta
Full-featured Graphics Library for embedded systems Littlev Graphics Library provides everything you need to create a Graphical User Interface (GUI) on embedded systems with easy-to-use graphical elements, beautiful visual effects and low memory footprint.
[More info](#)

lv_examples
by kisvegabor,embeddeddt
Examples for LVGL graphics library Demos and examples to see and try the features of LVGL embedded GUI library.
[More info](#)

lvgl
by kisvegabor,embeddeddt,pete-pjb Version 8.3.3 INSTALLED
Full-featured Graphics Library for Embedded Systems Powerful and easy-to-use embedded GUI with many widgets, advanced visual effects (opacity, antialiasing, animations) and low memory requirements (16K RAM, 64K Flash).
[More info](#)

Close

Copy the lv_conf.h of the data center .

As shown:

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

- Path: 3.3-4_TFT-LVGL-Widgets > TFT_eSPI bottom layer replacement file
- Selected file: lv_conf.h (highlighted with a red box)
- File properties:
 - Name: lv_conf.h
 - Modified Date: 2022/9/28 13:15
 - Type: C Header 源文件
 - Size: 24 KB
- Toolbar buttons include: 固定到快 (Pin to Taskbar), 复制 (Copy), 粘贴 (Paste), 移动到 (Move to), 复制到 (Copy to), 删 (Delete), 重命名 (Rename), 新建文件夹 (New Folder), 新建 (New), 属性 (Properties), 打开 (Open), 历史记录 (History), 全部选择 (Select All), 全部取消 (Deselect All), 反向选择 (Invert Selection), and 选择 (Select).
- Left sidebar navigation includes: 快速访问 (Quick Access), OneDrive, WPS网盘 (WPS Cloud), 此电脑 (This PC), 3D 对象 (3D Objects), Apple iPhone, 视频 (Videos), 图片 (Images), and 文档 (Documents).

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

As shown:



此电脑 > OS (C:) > 用户 > zhang'pei > 文档 > Arduino > libraries			
名称	修改日期	类型	大小
Adafruit_CCS3501_CDS	2022/6/27 12:06	文件夹	
Adafruit_Unified_Sensor	2022/7/6 9:23	文件夹	
ArduinoJson	2022/6/27 12:06	文件夹	
AsyncTCP	2022/6/27 12:06	文件夹	
Audio	2022/6/28 17:44	文件夹	
DallasTemperature	2022/6/27 12:06	文件夹	
DHT_sensor_library	2022/6/27 12:06	文件夹	
DHT_sensor_library_for_ESPx	2022/6/25 10:23	文件夹	
ESP32Servo	2022/6/27 12:06	文件夹	
ESPAsyncWebServer	2022/6/27 12:06	文件夹	
FastLED	2022/7/6 9:23	文件夹	
GFX_Library_for_Arduino	2022/8/9 18:08	文件夹	
gt911-arduino-main	2022/8/17 10:21	文件夹	
GT911-master	2022/8/15 15:10	文件夹	
IRremote	2022/6/27 12:06	文件夹	
JPEGDecoder	2022/6/28 13:49	文件夹	
LiquidCrystal_I2C	2022/6/27 12:06	文件夹	
LovyanGFX	2022/7/31 14:05	文件夹	
lvgl	2022/3/4 10:31	文件夹	
MFRC522	2022/6/27 12:06	文件夹	
OneWire	2022/6/27 12:06	文件夹	
PNGdec	2022/6/28 10:48	文件夹	
Rtc_by_Makuna	2022/6/27 12:06	文件夹	
TFT_eSPI	2022/8/16 12:46	文件夹	
TFT_Touch-master	2022/8/1 12:16	文件夹	
Time	2022/7/6 9:23	文件夹	
TJpg_Decoder	2022/8/3 14:25	文件夹	
Touch_test	2022/8/1 12:12	文件夹	
TP_Arduino_DigitalRain_Anim-main	2022/7/31 13:13	文件夹	
XPT2046_Touchscreen	2022/7/17 18:09	文件夹	
XT_DAC_Audio	2022/7/2 17:12	文件夹	
lv_arduino.rar	2022/7/21 14:20	360压缩 RAR 文件	6,740 KB
lv_conf.h	2022/8/19 17:01	C Header 源文件	24 KB
readme.txt	2022/6/15 15:12	文本文档	1 KB

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

As shown:



名称

名称	修改日期	类型	大小
demos	2022/10/6 16:17	文件夹	
docs	2022/10/6 16:17	文件夹	
env_support	2022/10/6 16:17	文件夹	
examples	2022/10/6 16:17	文件夹	
scripts	2022/10/6 16:17	文件夹	
src	2022/10/12 17:10	文件夹	
tests			
CMakeLists.txt			
component.mk			
idf_component.yml			
Kconfig	2022/10/6 16:17	文件	42 KB
library.json	2022/10/6 16:17	JSON 源文件	1 KB
library.properties	2022/10/6 16:17	Properties 源文件	1 KB
LICENCE.txt	2022/10/6 16:17	文本文档	2 KB
lv_conf_template.h	2022/10/6 16:17	C Header 源文件	26 KB
lvgl.h	2022/10/6 16:17	C Header 源文件	3 KB
lvgl.mk	2022/10/6 16:17	Makefile 源文件	1 KB
README.md	2022/10/6 16:17	Markdown 源文件	9 KB
README_pt_BR.md	2022/10/6 16:17	Markdown 源文件	9 KB
README_zh.md	2022/10/6 16:17	Markdown 源文件	9 KB
SConscript	2022/10/6 16:17	文件	1 KB

After compiling, you can run LVGL and touch normally.