



# Welcome to the JCZN Workshop!

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

一、 Introduction	2
二、 Installing using Arduino IDE	2
三、 sample program usage	10



# Getting Started

## Introduction

The objective of this post is to explain how to upload an Arduino program to the ESP32-2432S022 module, from JCZN .

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.

ESP wroom 32 has a significant performance improvement. It is equipped with a high-performance dual-core Tensilica LX6 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.



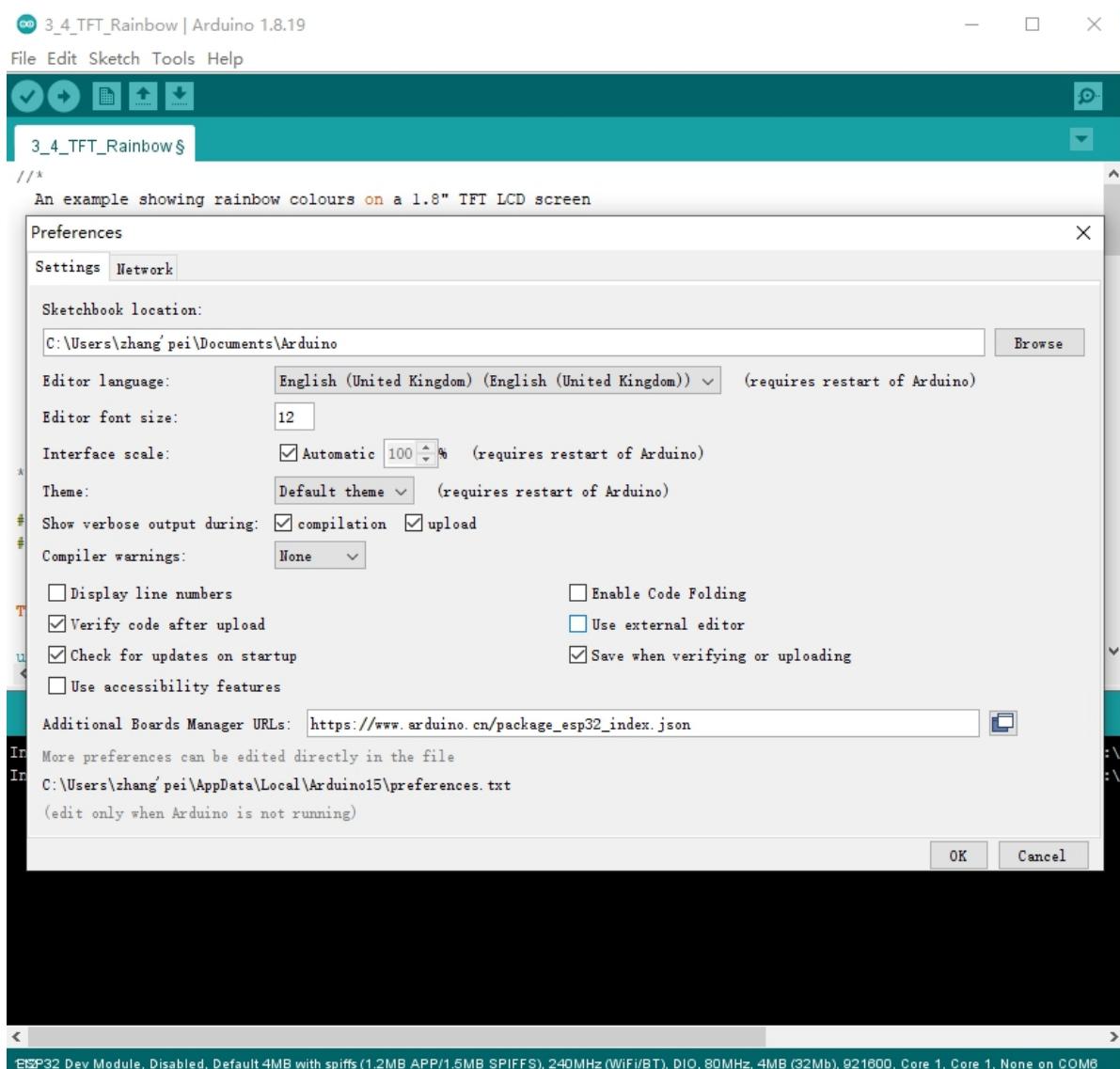
The screenshot shows the Arduino IDE interface with the title bar "3\_4\_TFT\_Rainbow | Arduino 1.8.19". The menu bar includes File, Edit, Sketch, Tools, and Help. A context menu is open over some code, with options like New, Open..., Open Recent, Sketchbook, Examples, Close, Save, Save As..., Page Setup, 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 shows "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:

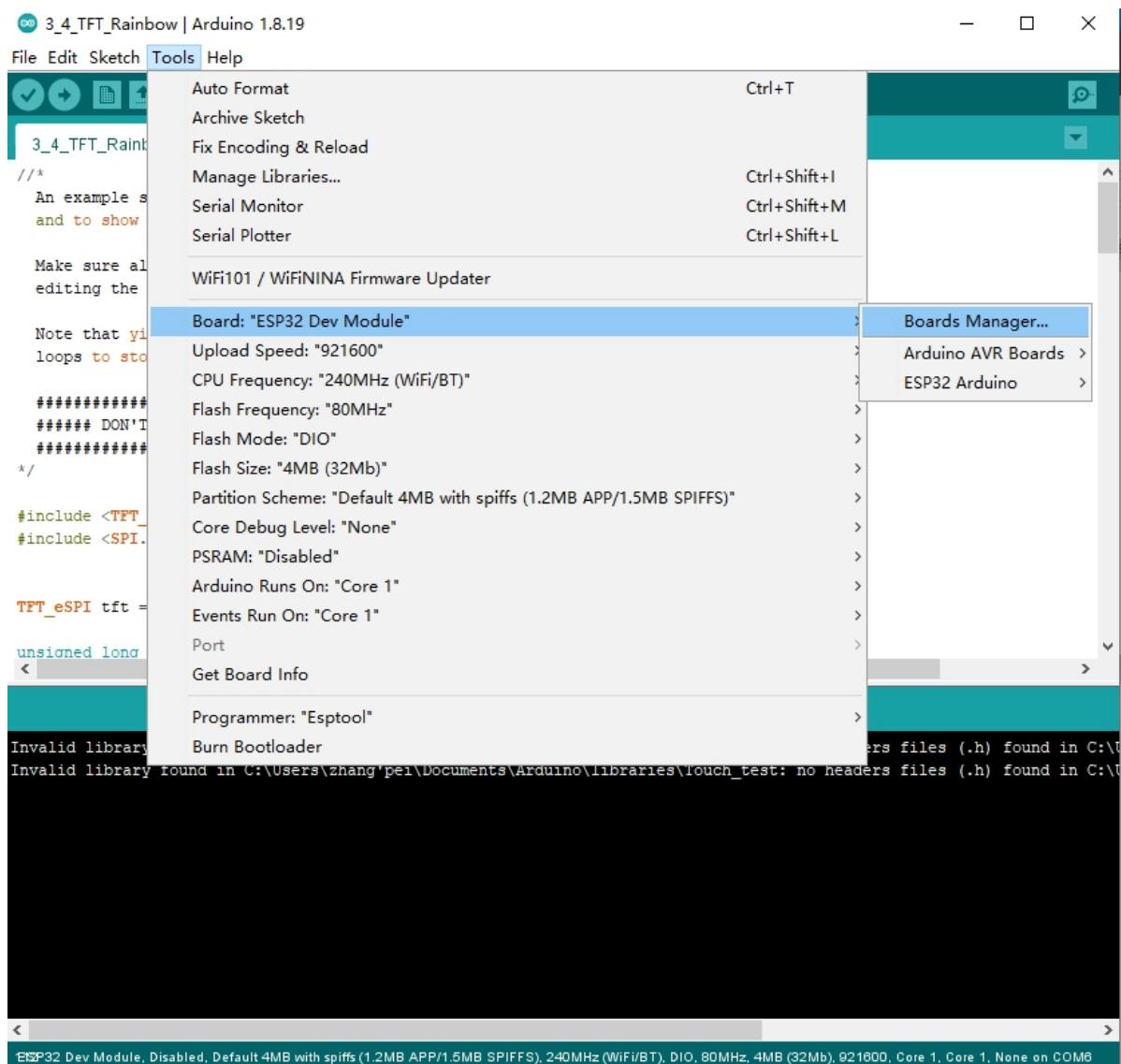
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_dev\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json)

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

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

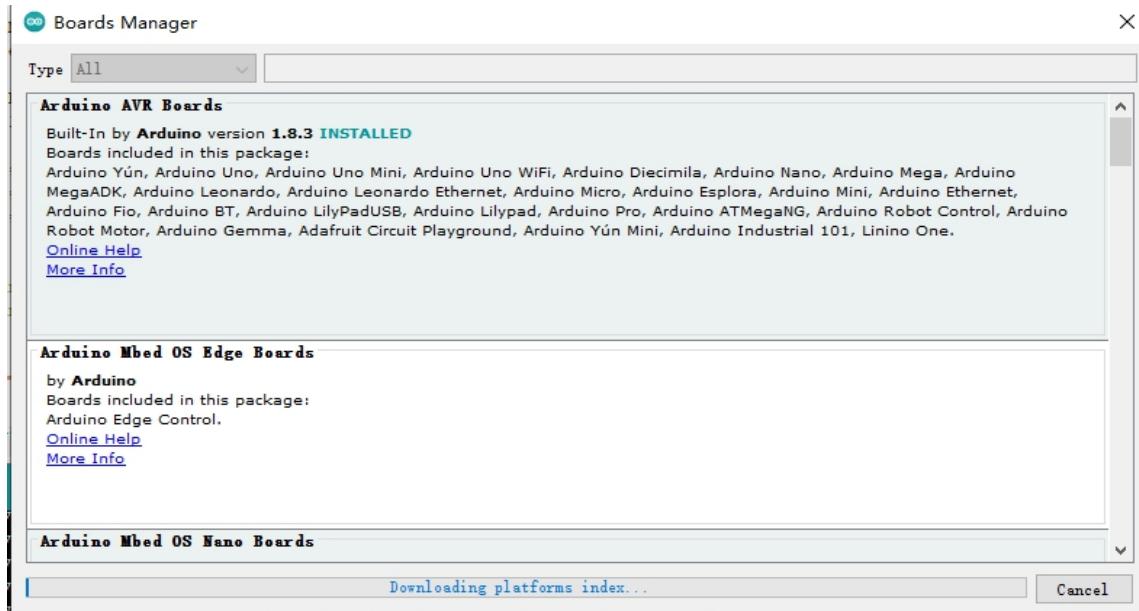


- (9) In the Arduino IDE click on the Tools menu on the top menu bar.
- (10) Scroll down to the Board: entry
- (11) A submenu will open when you highlight the Board: entry.
- (12) At the top of the submenu is Boards Manager. Click on it to open the Boards Manager dialog box.
- (13) In the search box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

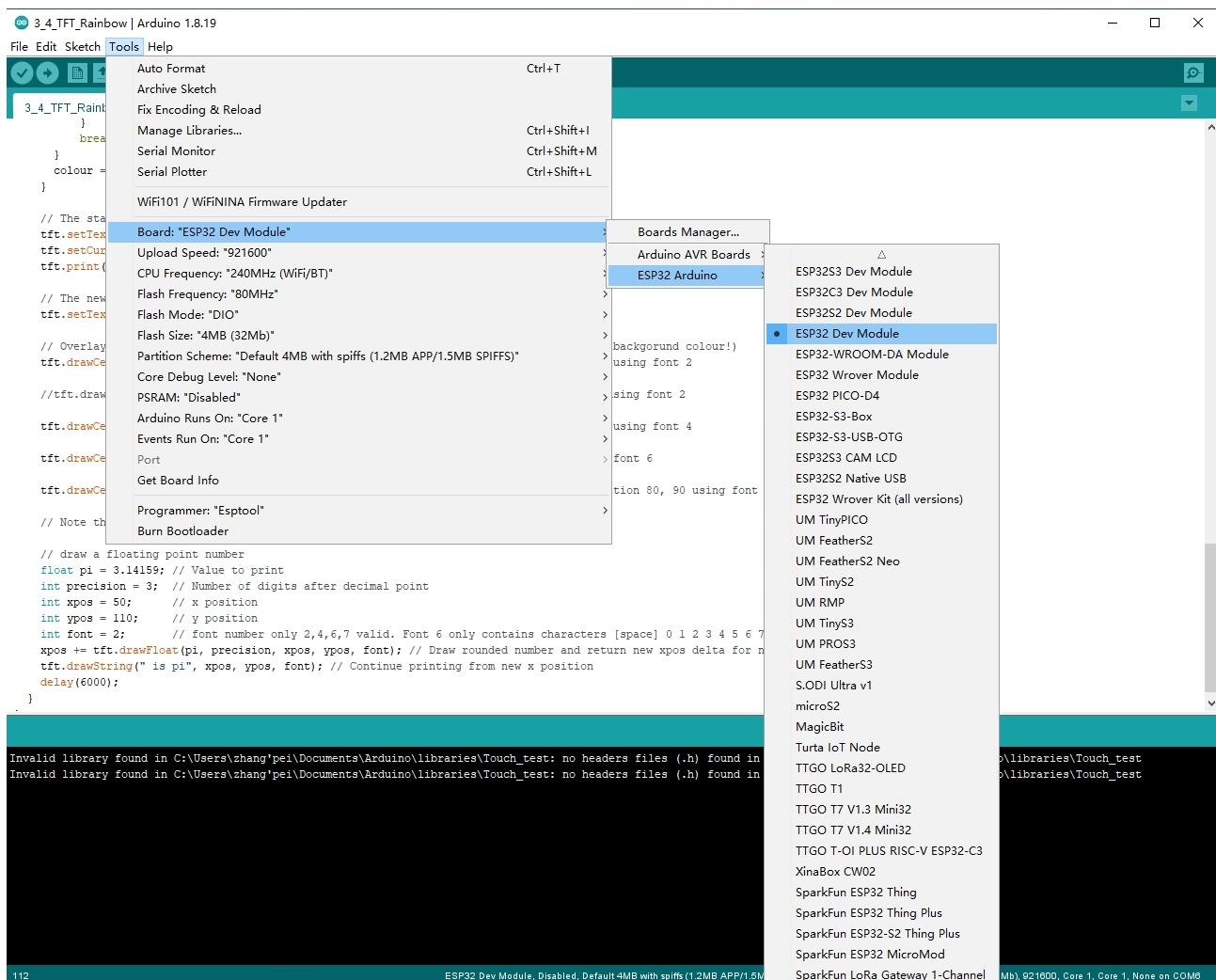


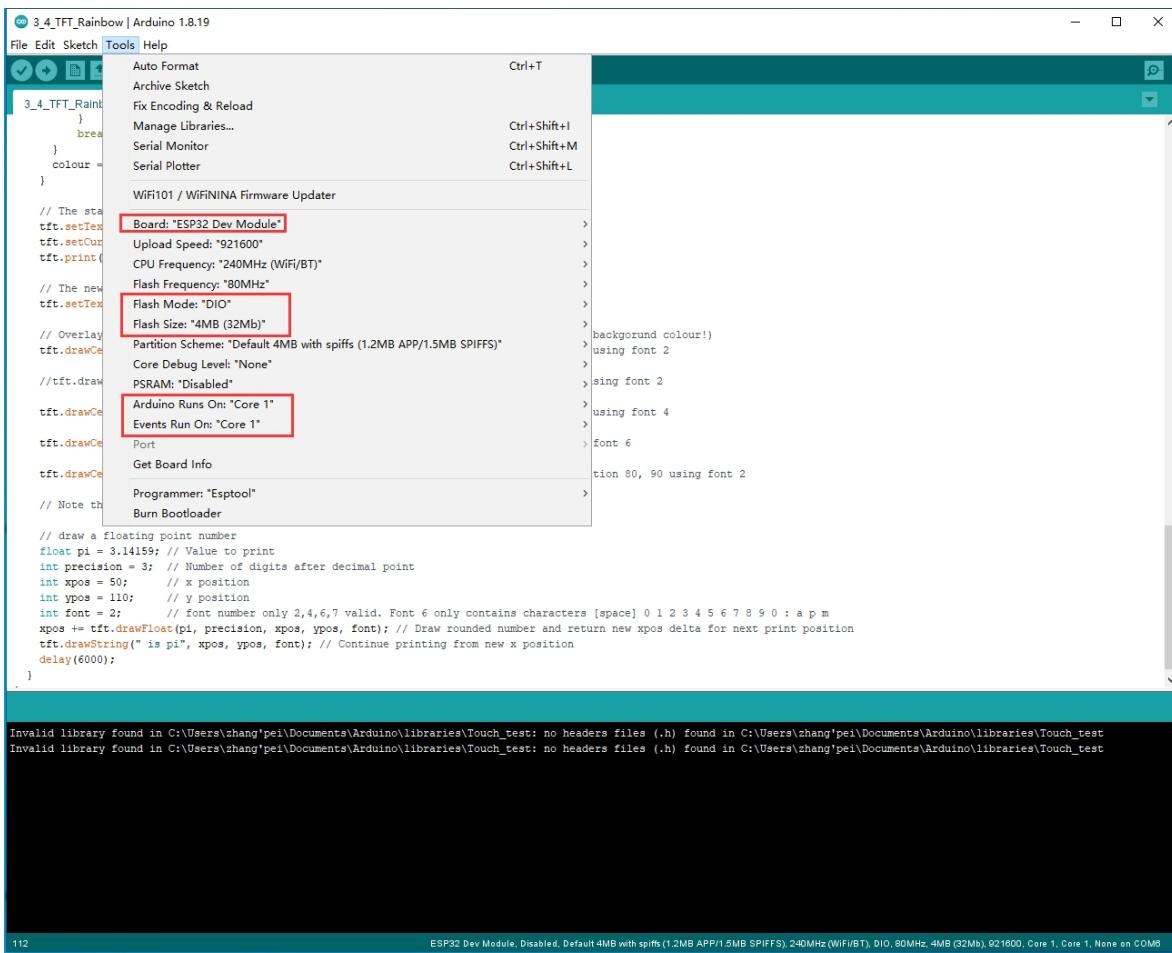
(14) You should see an entry for “esp32 by Espressif Systems”. Highlight this entry and click on the Install button.

This will install the ESP32 boards into your Arduino IDE

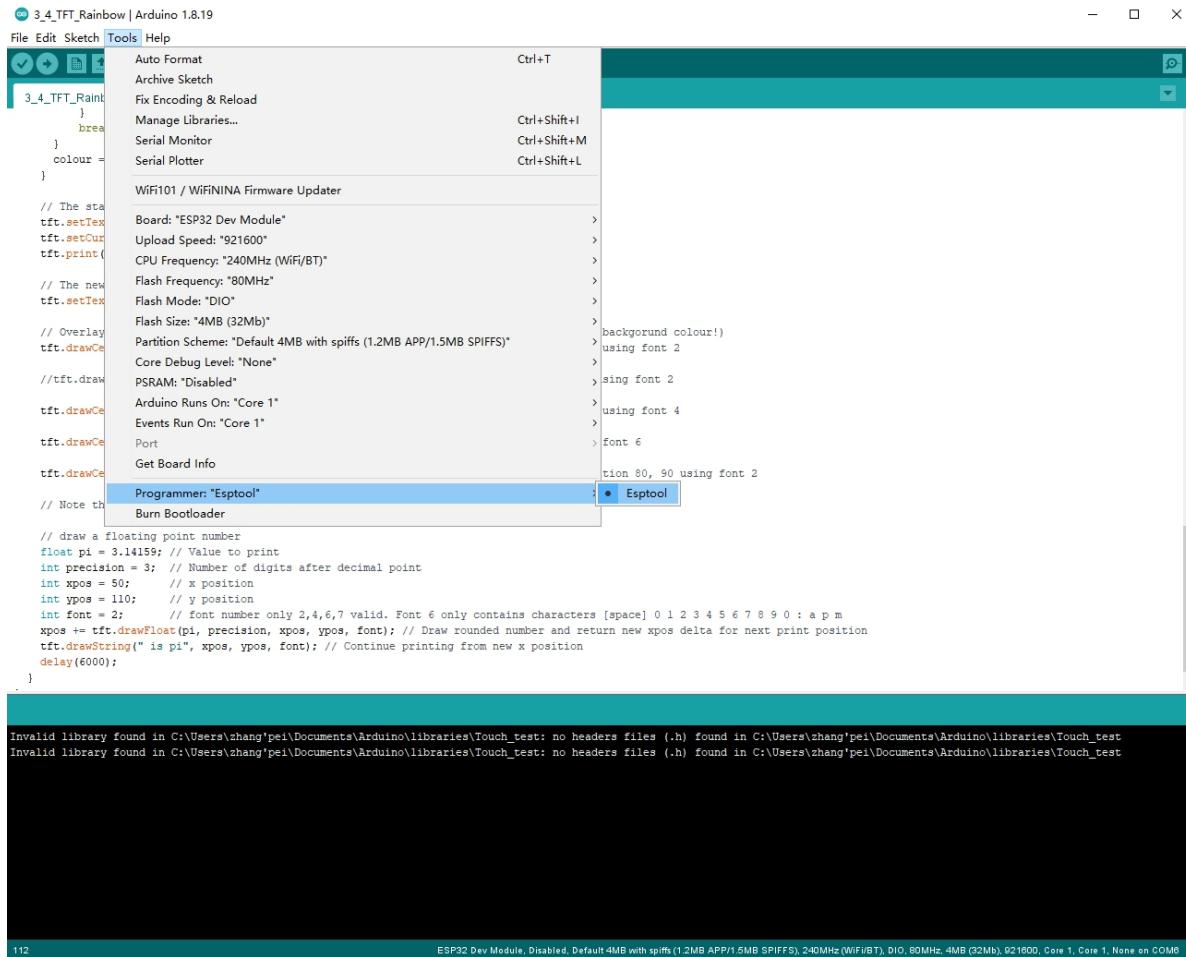


Once the installation completes, we need to select the correct board options for the "ESP32 Arduino" board. In the board type, in the tools tab, we choose "ESP32 Dev Module".





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.

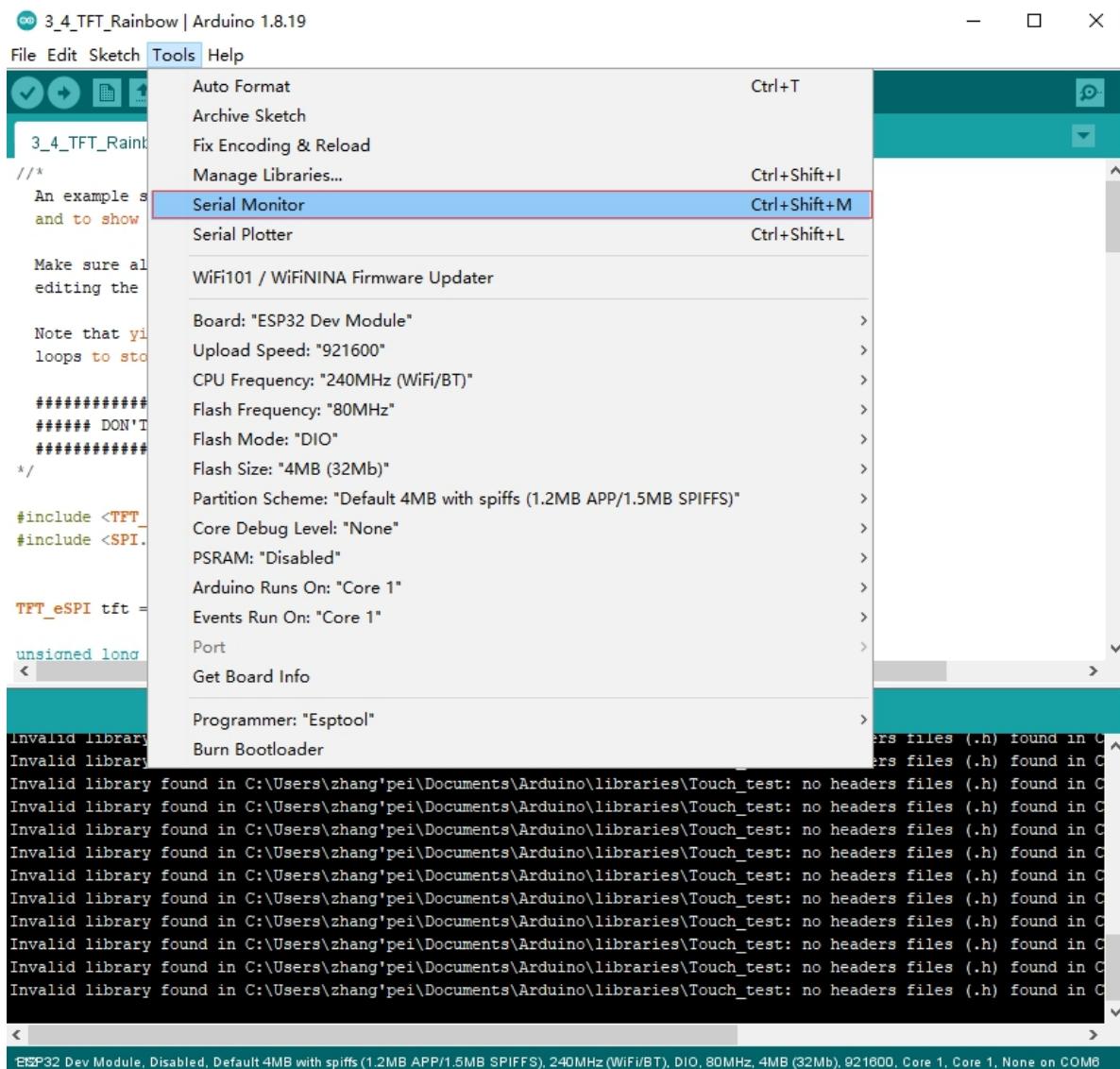
### First program

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);
}
```

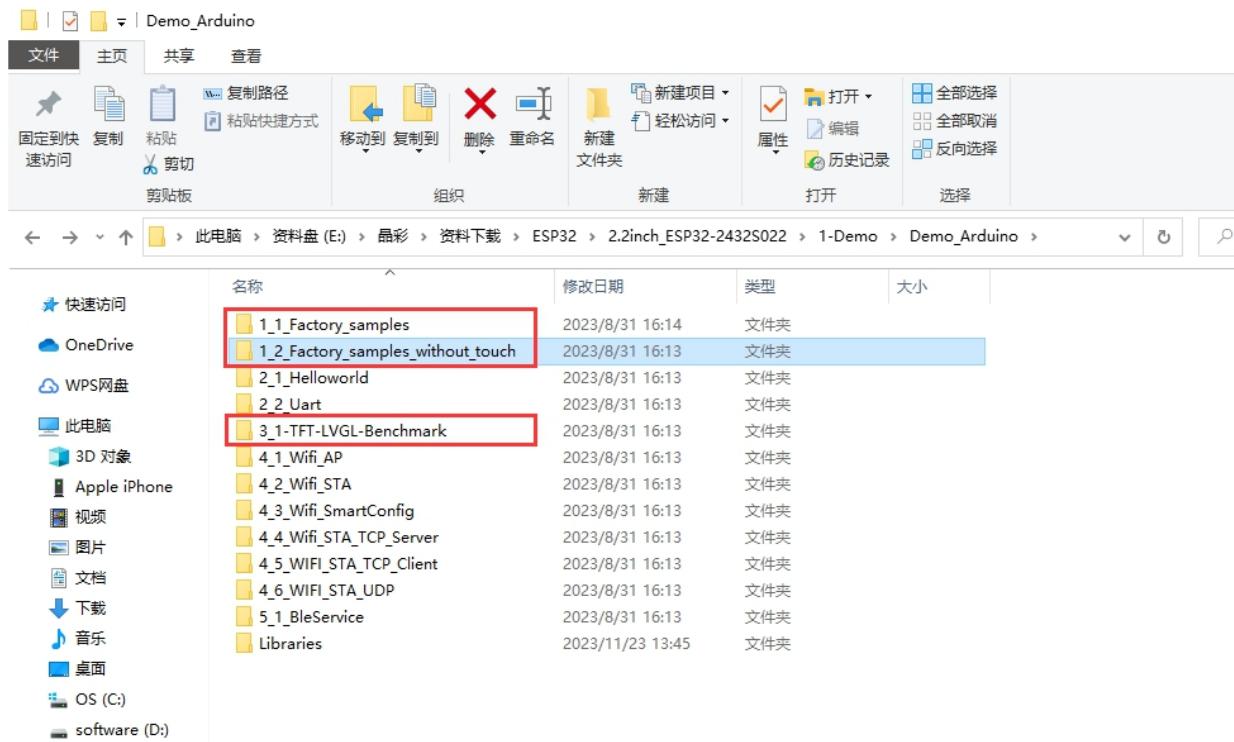
If everything is working fine, we will see the output in the serial console shown.



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 LovyanGFX library as the basic application. This library supports various commonly used driver chips, such as ST7735, ST7789, ILI9341, etc., and has good compatibility.

LovyanGFX library file installation:

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



Factory\_samples | 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

WiFi101 / WiFiNINA Firmware Updater

Board: "ESP32 Dev Module" >

Upload Speed: "921600" >

CPU Frequency: "240MHz (WiFi/BT)" >

Flash Frequency: "80MHz" >

Flash Mode: "QIO" >

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" >

Erase All Flash Before Sketch Upload: "Disabled" >

JTAG Adapter: "Disabled" >

Port: "COM5" >

Get Board Info

Programmer

Burn Bootloader >

```
cfg.pin_rst = -1;
cfg.pin_busy = -1;

cfg.panel_width = 240;
cfg.panel_height = 320;
cfg.offset_x = 0;
cfg.offset_y = 0;
cfg.offset_rotation = 0;
// cfg.dummy_read_pixel = 8;
// cfg.dummy_read_bits = 1;
```

Done uploading.

Writing at 0x0008fc1f... (86 %)  
Writing at 0x00096cf6... (90 %)  
Writing at 0x0009c303... (95 %)  
Writing at 0x000a1c86... (100 %)  
Wrote 599840 bytes (345816 compressed) at 0x00010000 in 4.7 seconds (effective 1012.0 kbit/s)...  
Hash of data verified.

Leaving...  
Hard resetting via RTS pin...



Library Manager

Type All Topic All LovyanGFX

by @chrmlinux03  
Use the DAC function of ESP32 to output C\_ESP\_8\_BIT\_composite with LovyanGFX. a library to Use the DAC function of ESP32 to output C\_ESP\_8\_BIT\_composite with LovyanGFX.

**LGFXMeter**  
by tobozo  
**LGFXMeter** based Gauge Decoration and Animation library Create and animate a custom gauge using LovyanGFX for high speed, antialiased rendering.

**LovyanGFX**  
by lovyan03 Version 1.1.5 INSTALLED  
**TFT LCD Graphics driver with touch for ESP32, ESP8266, SAMD21, SAMD51, RP2040** Supports TFT LCD displays using drivers that operate with hardware SPI. ESP32, ESP8266, SAMD21, SAMD51, RP2040, M5Stack, M5StackCore2, M5StickC, M5StickCPlus, ODROID-GO, TTGO T-Watch, TTGO T-Wristband, ESP-WROVER-KIT, WioTerminal, WiFiBoy, MakePython, HX8357, ILI9163, ILI9342, ILI9341, ILI9486, ILI9488, ST7735, ST7789, ST7796, SSD1351

More info Select version Install Update Close

After configuring these, compile the arduino function in Factory\_samples to light up the screen .

### About the use of touch and LVGL:

Find the data center Factory\_samples

As shown :

文件 主页 共享 查看

复制 貼 粘贴 剪切 复制路径 粘贴快捷方式 移动到 复制到 删除 重命名 新建项目 新建文件夹 新建

新建项目 轻松访问 属性 打开 编辑 历史记录 全部选择 全部取消 反向选择

此电脑 > 资料盘 (E:) > 晶彩 > 资料下载 > ESP32 > 2.2inch\_ESP32-2432S022 > 1-Demo > Demo\_Arduino

名称	修改日期	类型	大小
1_1_Factory_samples	2023/8/31 16:14	文件夹	
1_2_Factory_samples_without_touch	2023/8/31 16:13	文件夹	
2_1_Helloworld	2023/8/31 16:13	文件夹	
2_2_Uart	2023/8/31 16:13	文件夹	
3_1-TFT-LVGL-Benchmark	2023/8/31 16:13	文件夹	
4_1_Wifi_AP	2023/8/31 16:13	文件夹	
4_2_Wifi_STA	2023/8/31 16:13	文件夹	
4_3_Wifi_SmartConfig	2023/8/31 16:13	文件夹	
4_4_Wifi_STA_TCP_Server	2023/8/31 16:13	文件夹	
4_5_WIFI_STA_TCP_Client	2023/8/31 16:13	文件夹	
4_6_WIFI_STA_UDP	2023/8/31 16:13	文件夹	
5_1_BleService	2023/8/31 16:13	文件夹	
Libraries	2023/11/23 13:45	文件夹	



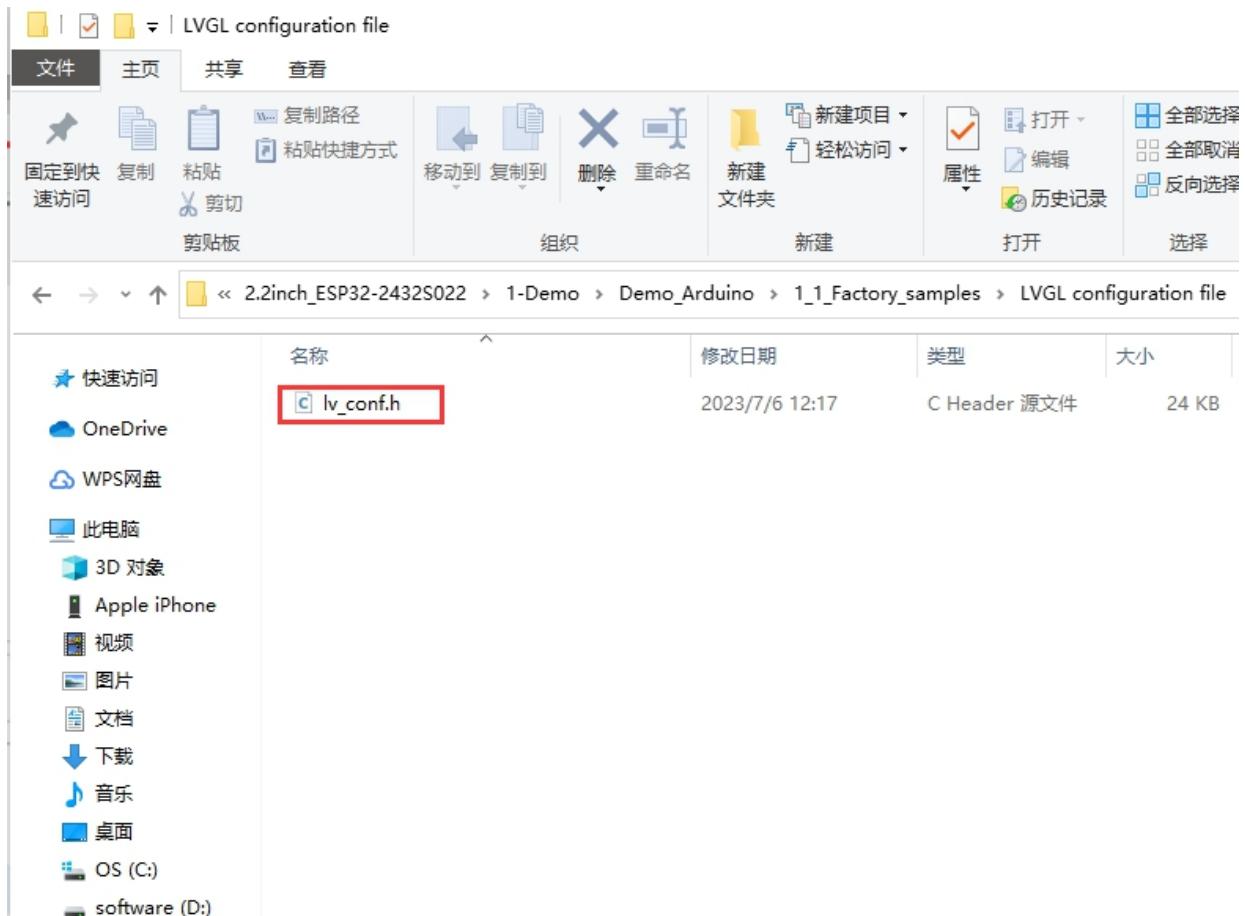
Download library files .

Lvgl

The screenshot shows a software window titled "Library Manager". At the top, there are dropdown menus for "Type" (set to "All") and "Topic" (set to "All"), and a search bar containing "LVGL". Below the search bar, there is a list of libraries. The first item in the list is "lv\_examples" by "kisvegabor,embeddedt". The second item, "lvgl", is highlighted with a red border. It is described as a "Full-featured Graphics Library for Embedded Systems" developed by "kisvegabor,embeddedt,pete-pjb". The description states it provides a powerful and easy-to-use embedded GUI with many widgets, advanced visual effects, and low memory requirements (16K RAM, 64K Flash). There are "More info" links for both entries. At the bottom right of the window is a "Close" button.

Copy the data center's lv\_conf.h

As shown in the figure:



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_CCS811	2022/6/27 12:06	文件夹	
Adafruit_Unified_Sensor	2022/6/27 12:06	文件夹	
ArduinoJson	2022/7/6 9:23	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

After compiling, you can run LVGL and touch normally.