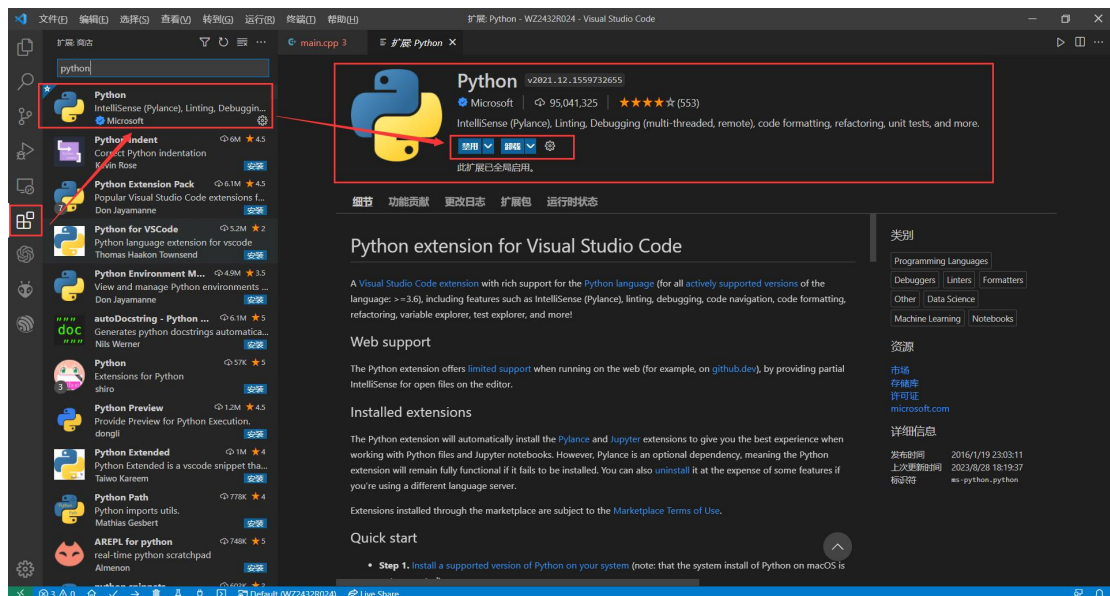


# PlatformIO

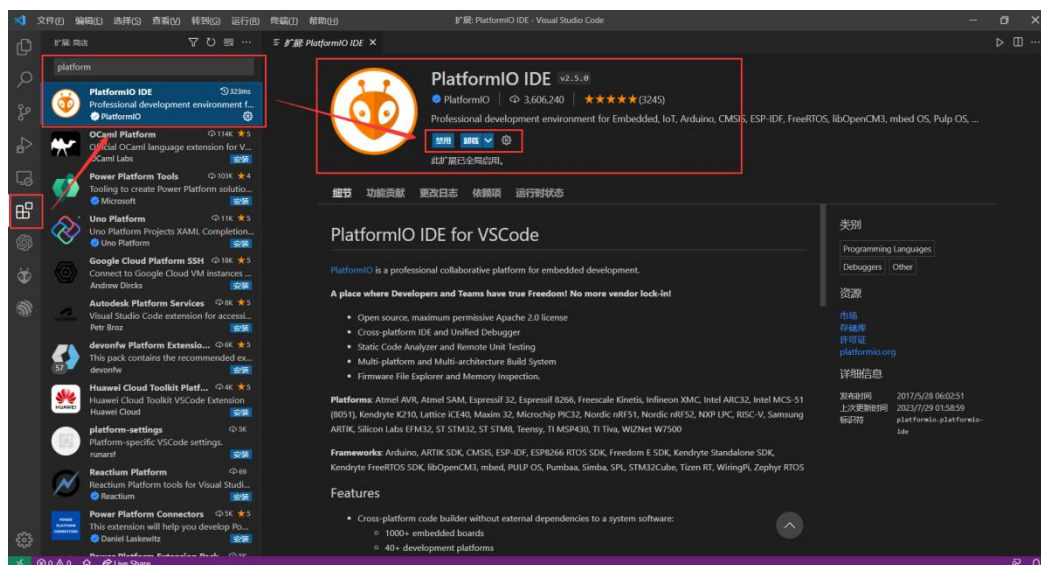
## DIS07050H

Take the DIS07050H as an example

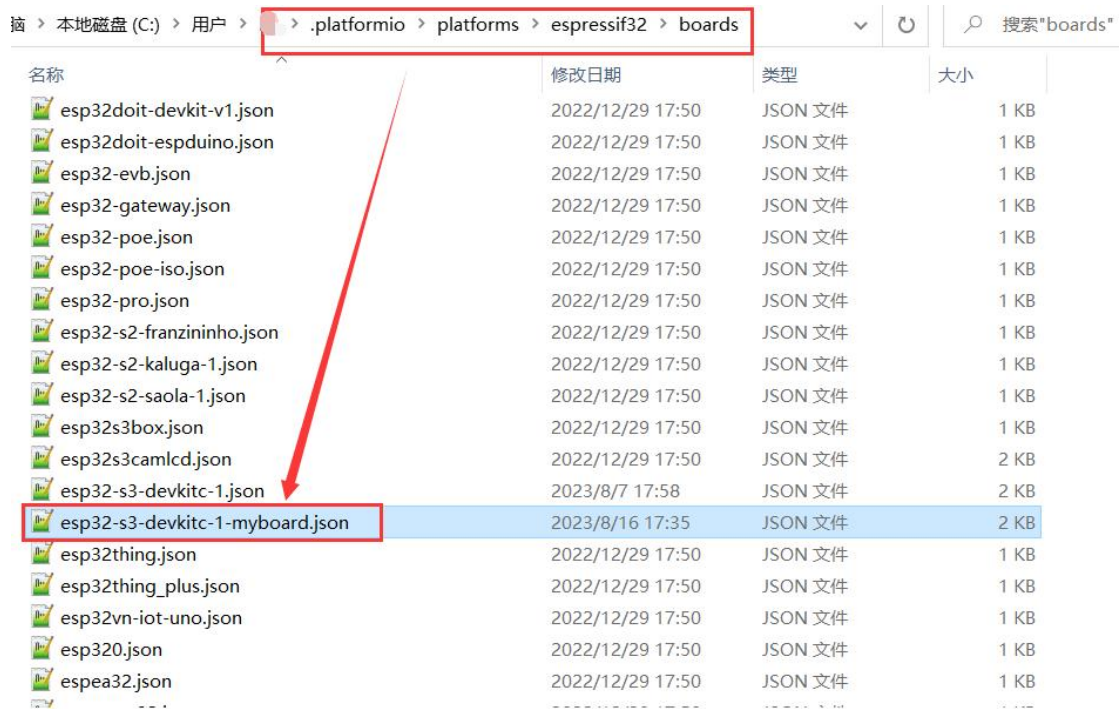
First open the VScode to check if the python is installed



Open the VScode to download the PlatformIO

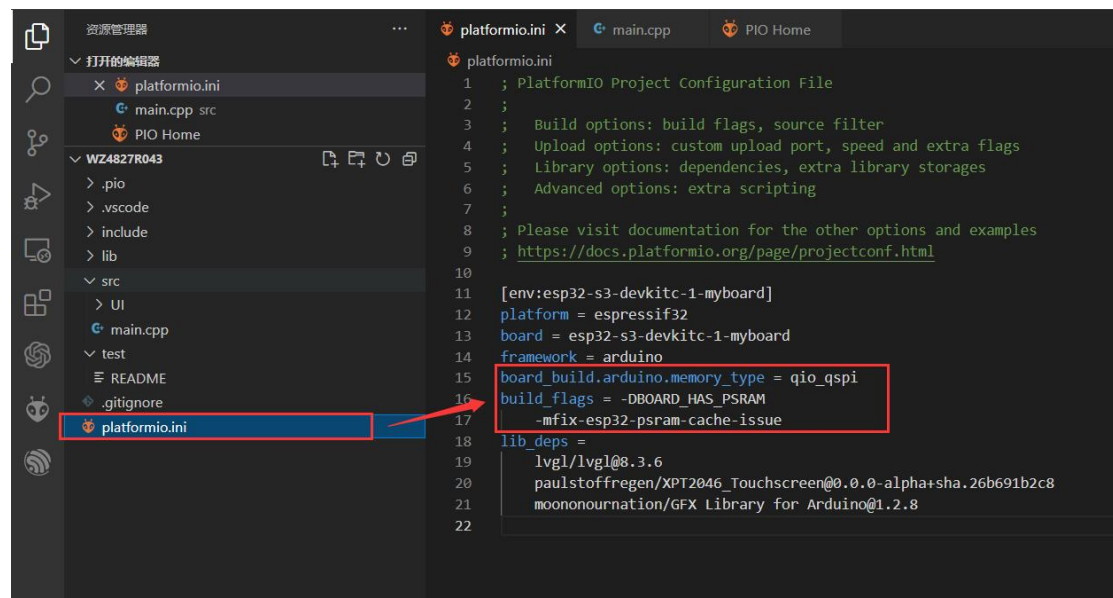


First, add the custom board to the directory in the figure below

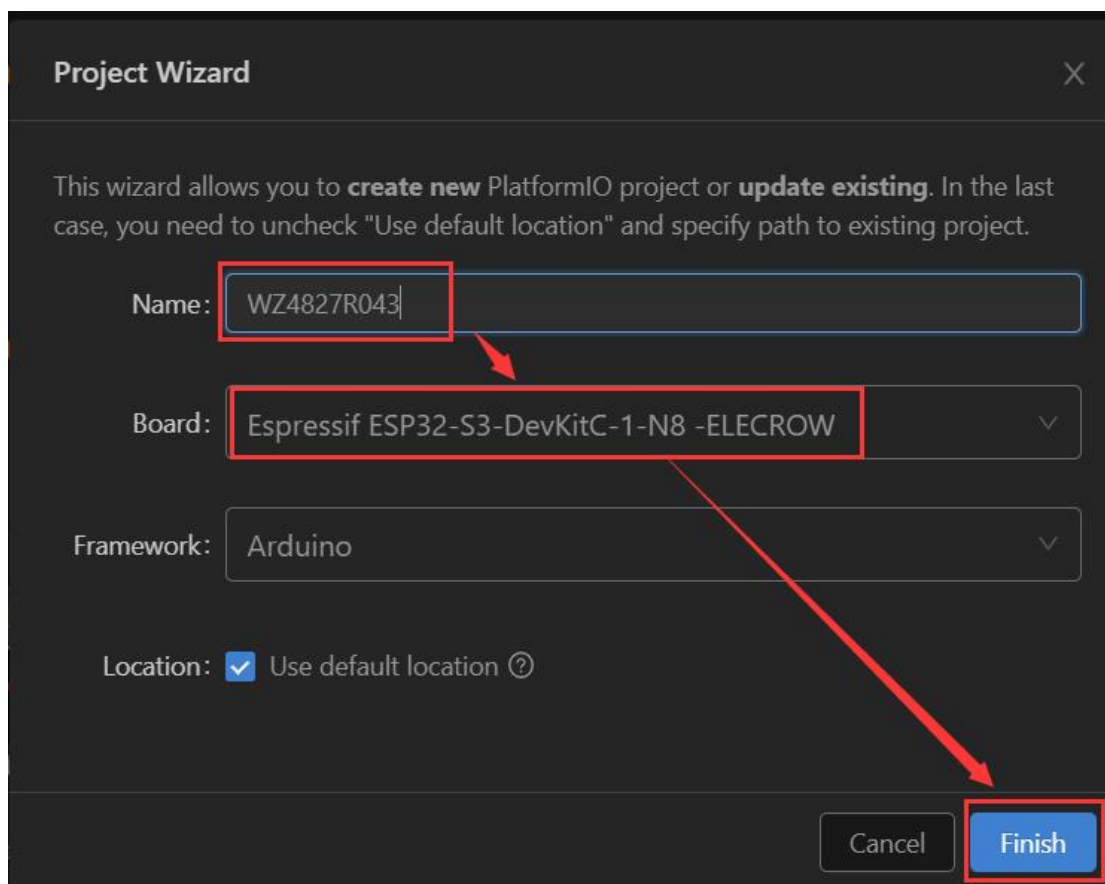
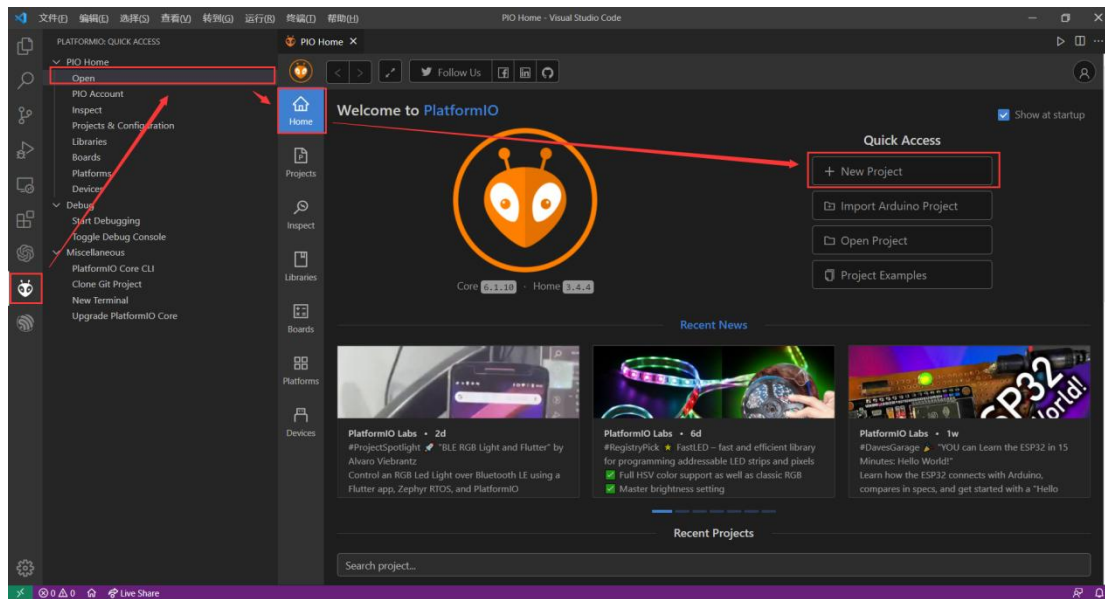


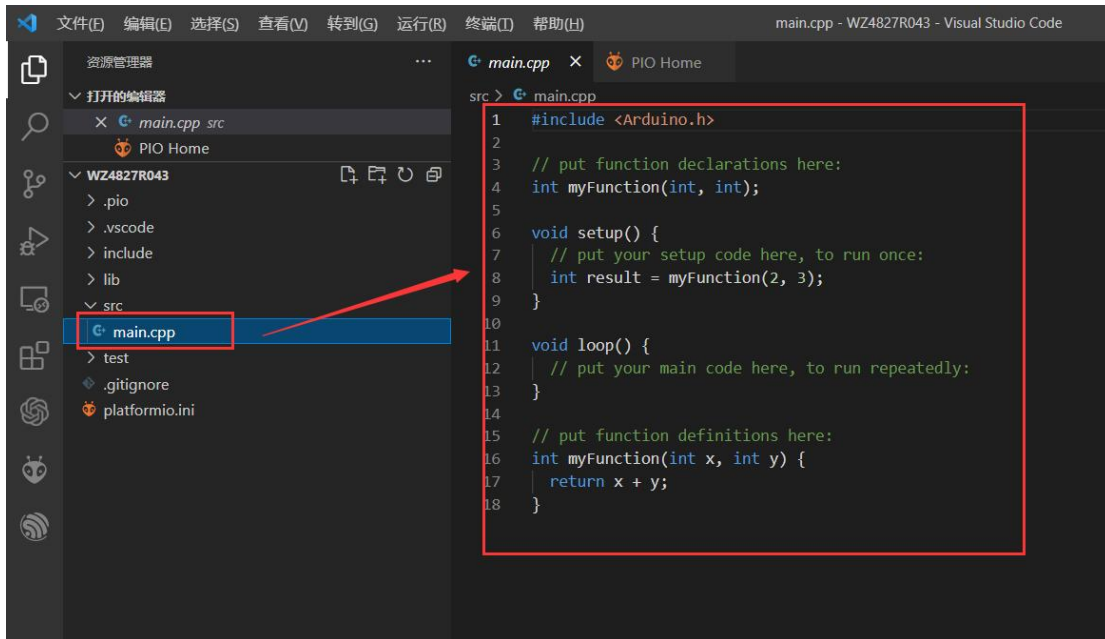
Add the following code to the platformio.ini file

```
board_build.arduino.memory_type = qio_qspi
build_flags = -DBOARD_HAS_PSRAM
-mfix-esp32-psram-cache-issue
```

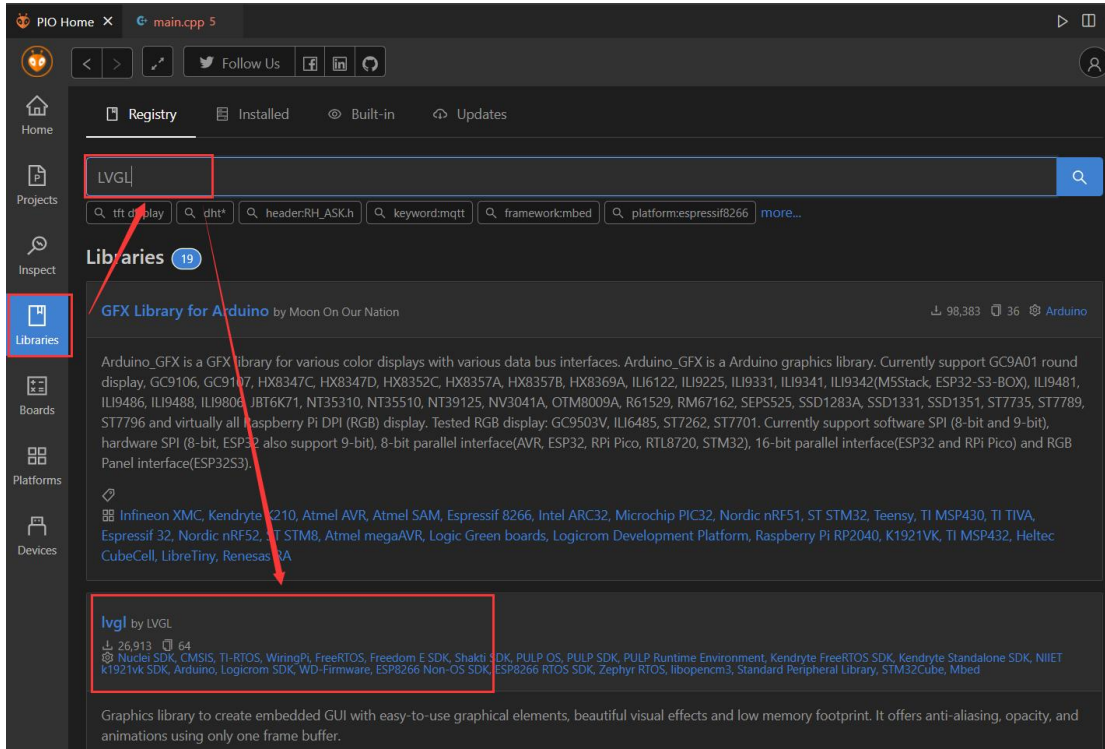


## Create new projects





### Download the required library (lvgl, XPT2046\_Touchscreen, GFX Library for Arduino)



Registry Installed Built-in Updates

lvgl by LVGL

Graphics library to create embedded GUI with easy-to-use graphical elements, beautiful visual effects and low memory usage, and animations using only one frame buffer.

Installation

8.3.6 released 3 months ago **Add to Project** | More info

Examples Installation Headers Changelog

lv\_example\_chart\_1

lv\_example\_chart\_1

```
#include "../lv_examples.h"
#if LV_USE_CHART && LV_BUILD_EXAMPLES

void lv_example_chart_1(void)
{
    /*Create a chart*/
    lv_obj_t * chart;
    chart = lv_chart_create(lv_scr_act());
```

Add project dependency

lvgl/lvgl@8.3.6

Projects\WZ4827R043

You can manage your projects in the "Projects" section: create a new or add existing.

Information

- > Registry and Specification
- > External resources

Cancel **Add**

Registry Installed Built-in Updates

# lvgl

by LVGL

Graphics library to create embedded GUI with easy-to-use graphical elements, beautiful visual effects, opacity, and animations using only one frame buffer.

8.3.6 released 3 months ago [Add to Project](#) | [More info](#)

Examples Installation Headers Changelog

lv\_example\_chart\_1

```
#include "../lv_examples.h"
#if LV_USE_CHART && LV_BUILD_EXAMPLES

void lv_example_chart_1(void)
{
    /*Create a chart*/
    lv_obj_t * chart;
    chart = lv_chart_create(lv_scr_act());
    lv_obj_set_size(chart, 200, 150);
    lv_obj_center(chart);
    lv_chart_set_type(chart, LV_CHART_TYPE_LINE); /*Show lines and points too*/

    /*Add two data series*/

```

Tags: graphics, gui, embedded, tft, lvgl

Platforms: Aceinna IMU, Infineon XMC, Kendryte K210, Nuclei, Atmel AVR, Atmel SAM, Espressif 8266, Freescale Kinetis, Intel ARC32, Linux ARM, Linux i686, Linux x86\_64, Native, Windows x86, Microchip PIC32, Nordic nRF51, NXP LPC, Silicon Labs EFM32

Registry Installed Built-in Updates

XPT2046

tft display, dh1\*, header-RH\_ASK.h, keyword.mqtt, framework:mbed, platform:espressif8266, more...

## Libraries

27

**TFT\_eSPI** by Bodmer ↓ 80,577 🔒 152 🔗 Arduino

A TFT and ePaper (SPI or parallel interface) graphics library with optimisation for Raspberry Pi Pico, RP2040, ESP8266, ESP32 and STM32 processors

🔗 arduino, tft, display, ttgo, lilypi, wt32-sc01, epaper, pico, rp2040 nano connect, rp2040, stm32, esp8266, nodemcu, esp32, m5stack, ili9341, st7735, ili9163, s6d02a1, ili9481

🔗 Espressif 8266, ST STM32, Espressif 32, Raspberry Pi RP2040

**LovyanGFX** by loyva03 ↓ 11,756 🔒 29 🔗 Arduino, ESP8266 Non-OS SDK, ESP8266 RTOS SDK, Zephyr RTOS, Espidf

TFT LCD Graphics driver with touch for ESP32, ESP8266, SAMD21, SAMD51, RP2040

🔗 lcd, tft, gfx, lgfx, esp32, esp8266, samd21, samd51, m5stack, m5stackcore2, m5stickc, m5stickcplus, odroid-go, ttgo t-watch, ttgo t-wristband, esp-wrover-kit, wioterminal, wifiboy, nakepython, hx8357

🔗 Atmel SAM, Espressif 8266, Native, Espressif 32

**XPT2046 Touchscreen** by Paul Stoffregen ↓ 4,475 🔒 3 🔗 Arduino

Registry Installed Built-in Updates

### XPT2046\_Touchscreen

by Paul Stoffregen

Touchscreens using the XPT2046 controller chip. Many very low cost color TFT displays with touch screens have this controller.

#### Installation

0.0.0-alpha+sha.26b691b2c8 released 4 years ago **Add to Project** More info

Examples Installation Headers Changelog

Boards: ILI9341Test

#### ILI9341Test

```
#include <ILI9341_t3.h>
#include <font_Arial.h> // from ILI9341_t3
#include <XPT2046_Touchscreen.h>
#include <SPI.h>

#define CS_PIN 8
#define TFT_DC 9
#define TFT_CS 10
// MOSI=11, MISO=12, SCK=13

XPT2046_Touchscreen ts(CS_PIN);
#define TIRQ_PIN 2
//XPT2046_Touchscreen ts(CS_PIN); // Param 2 - NULL - No interrupts
//XPT2046_Touchscreen ts(CS_PIN, TFF); // Param 3 - TFF - No interrupts
```

### Add project dependency

paulstoffregen/XPT2046\_Touchscreen

Projects\WZ4827R043

You can manage your projects in the "Projects" section: create a new or add existing.

#### Information

- > Registry and Specification
- > External resources

Cancel **Add**

Registry Installed Built-in Updates

## XPT2046\_Touchscreen

by Paul Stoffregen

Touchscreens using the XPT2046 controller chip. Many very low cost color TFT displays with touch

**Installation**

0.0.0-alpha+sha.26b691b2c8 released 4 years ago [Add to Project](#) | [More info](#)

Examples Installation Headers Changelog

ILI9341Test

```

#include <ILI9341_t3.h>
#include <font_Arial.h> // from ILI9341_t3
#include <XPT2046_Touchscreen.h>
#include <SPI.h>

#define CS_PIN 8
#define TFT_DC 9
#define TFT_CS 10
// MOSI=11, MISO=12, SCK=13

XPT2046_Touchscreen ts(CS_PIN);
#define T_IRQ_PIN 2

```

**Tags**

- display
- tft
- lcd
- graphics
- spi
- touchscreen

**Platforms**

- Infineon XMC
- Kendryte K210
- Atmel AVR
- Atmel SAM
- Espressif 8266
- Intel ARC32
- Microchip PIC32
- Nordic nRF51
- ST STM32
- Teensy
- TI MSP430
- TI TIVA
- Espressif 32
- Nordic nRF52
- ST STM8
- Atmel megaAVR

**Congrats!**

Resolving esp32-s3-devkitc-1-myboard dependencies... Installing paulstoffregen/XPT2046\_Touchscreen

Unpacking 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

XPT2046\_Touchscreen@0.0.0-alpha+sha.26b691b2c8 has been installed!

GFX Library

tft display dht\* header:RH\_ASK.h keyword:mqtt framework:mbed platform:espressif8266 more...

**Libraries** 133

**Adafruit GFX Library** by Adafruit ↓ 251,629 2 Arduino

Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.

- display
- Infineon XMC, Kendryte K210, Atmel AVR, Atmel SAM, Espressif 8266, Intel ARC32, Microchip PIC32, Nordic nRF51, ST STM32, Teensy, TI MSP430, TI TIVA, Espressif 32, Nordic nRF52, ST STM8, Atmel megaAVR, Logic Green boards, Logicroom Development Platform, Raspberry Pi RP2040, K1921VK, TI MSP432, Heltec CubeCell, LibreTiny, Renesas RA

**Adafruit SSD1306** by Adafruit ↓ 152,824 5 Arduino

SSD1306 oled driver library for monochrome 128x64 and 128x32 displays

- display
- Infineon XMC, Kendryte K210, Atmel AVR, Atmel SAM, Espressif 8266, Intel ARC32, Microchip PIC32, Nordic nRF51, ST STM32, Teensy, TI MSP430, TI TIVA, Espressif 32, Nordic nRF52, ST STM8, Atmel megaAVR, Logic Green boards, Logicroom Development Platform, Raspberry Pi RP2040, K1921VK, TI MSP432, Heltec CubeCell, LibreTiny, Renesas RA

**GFX Library for Arduino** by Moon On Our Nation ↓ 98,399 36 Arduino

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, HX8347C, HX8347D, HX8352C, HX8357A, HX8357B, HX8369A, ILI6122, ILI9225, ILI9331, ILI9341, ILI9342(M5Stack, ESP32-S3-BOX), ILI9481,



Registry Installed Built-in Updates

### GFX Library for Arduino

by Moon On Our Nation

Arduino\_GFX is a GFX library for various color displays with various data bus interfaces. Arduino\_GFX is a Arduino graph round display, GC9106, GC9107, HX8347C, HX8347D, HX8352C, HX8357A, HX8357B, HX8369A, ILI6122, ILI9225, ILI93 S3-BOX), ILI9481, ILI9486, ILI9488, ILI9806, JBT6K71, NT35310, NT35510, NT39125, NV3041A, OTM8009A, R61529, RM SSD1331, SSD1351, ST7735, ST7789, ST7796 and virtually all Raspberry Pi DPI (RGB) display. Tested RGB display: GC95 Currently support software SPI (8-bit and 9-bit), hardware SPI (8-bit, ESP32 also support 9-bit), 8-bit parallel interface( STM32), 16-bit parallel interface(ESP32 and RPI Pico) and RGB Panel interface(ESP32S3).

**Installation**

1.2.8 released 11 months ago **Add to Project** | More info

Examples Installation Headers Changelog

U8g2FontUTF8Chinese

#### U8g2FontUTF8Chinese

```
/*  
 * U8g2 Chinese font example  
 * Please note this font is 1,024,137 in size and cannot fit in many platform.  
 * This font is generated by U8g2 tools:  
 * u8g2/tools/font/bdfconv/bdfconv -v -f 1 -b 1 -m "32-127,11904-12351,19968-40959,63744-64255,65280-65376" u  
 */
```

### Add project dependency

moononournation/GFX Library for Arduino@1.2.8

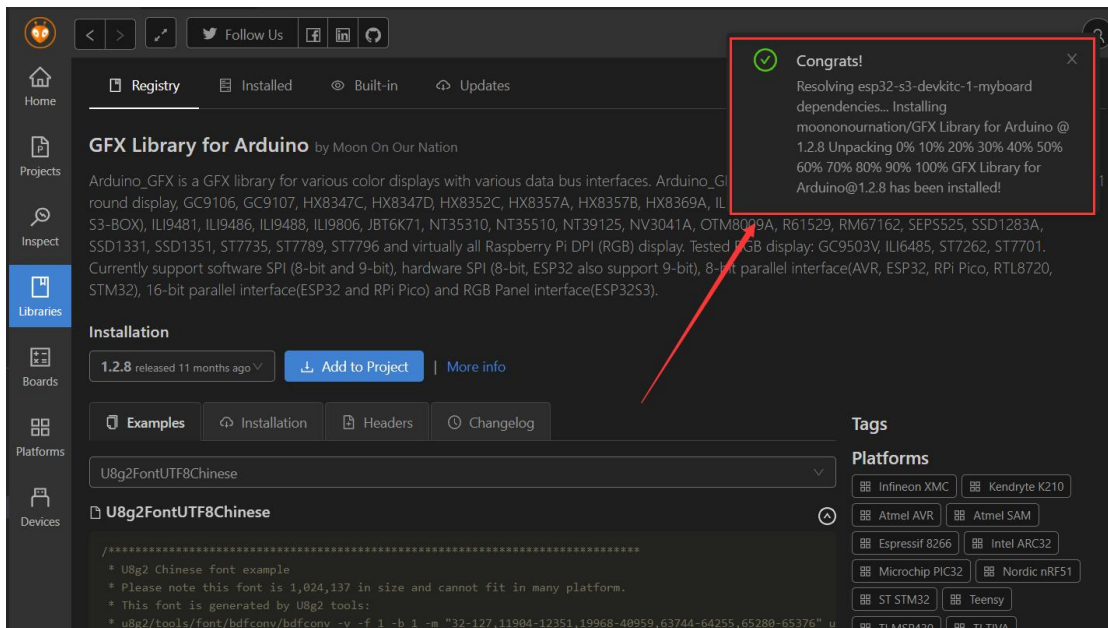
Projects\WZ4827R043

You can manage your projects in the "Projects" section: create a new or add existing.

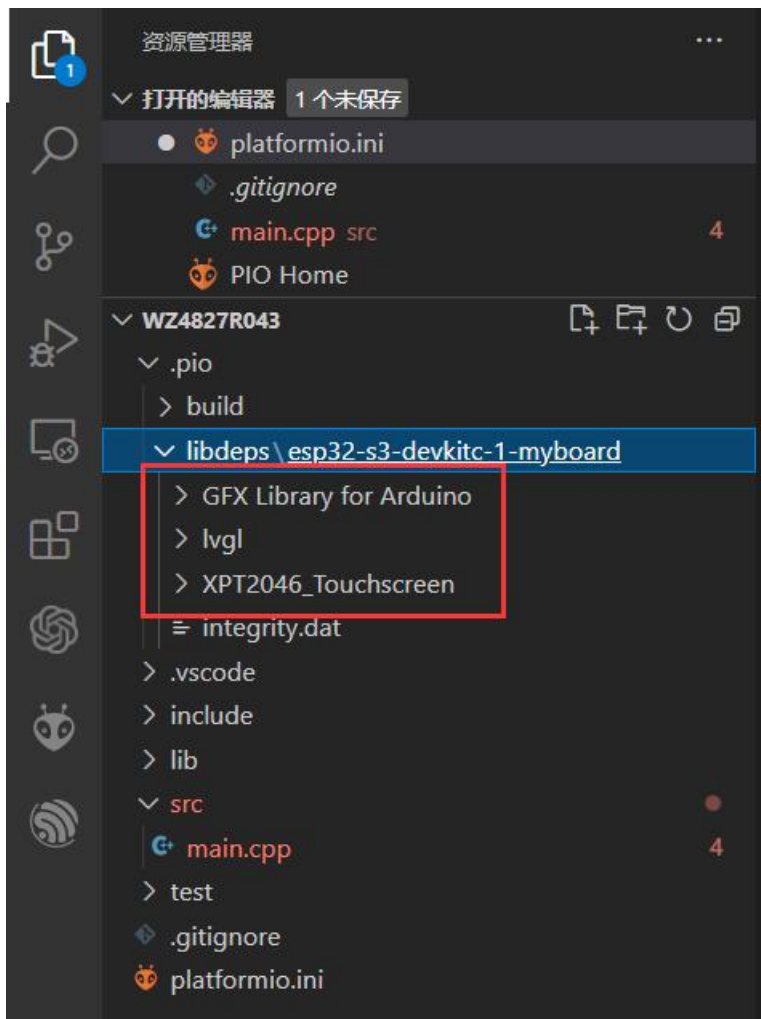
#### Information

- > Registry and Specification
- > External resources

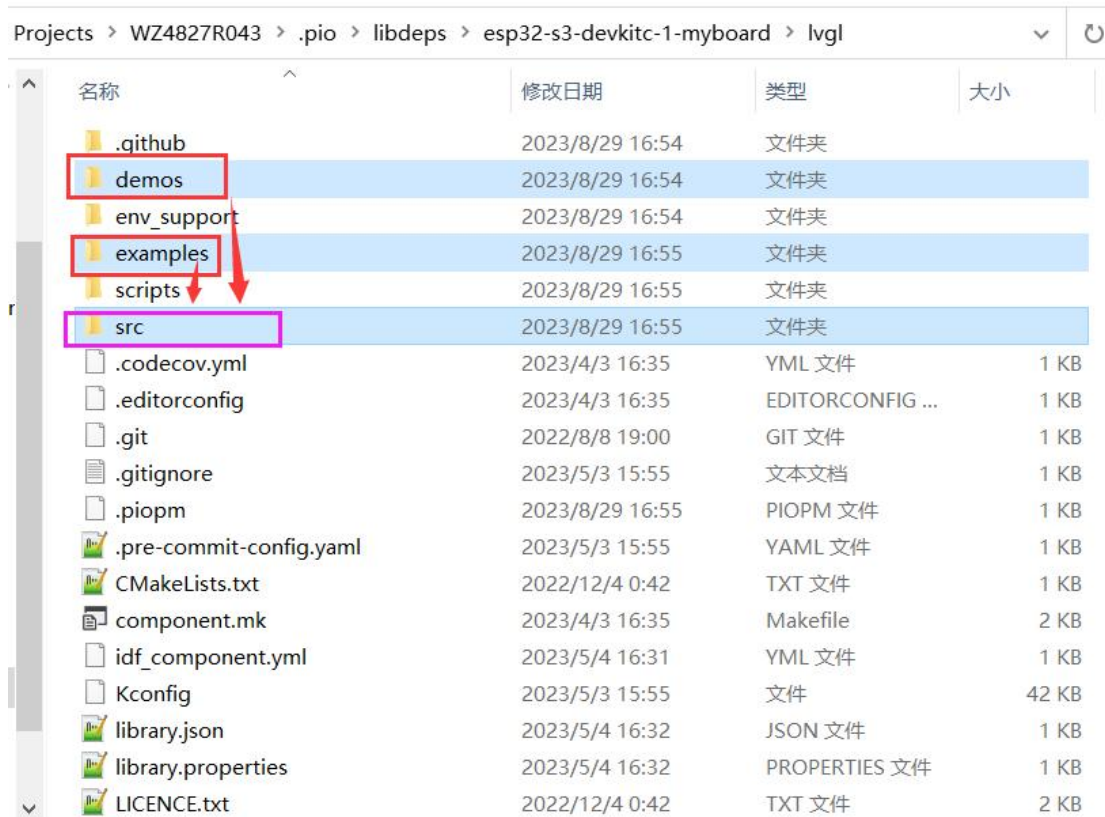
Cancel **Add**



We can see that the library has been added successfully!



Next, we want to configure the lvgl library, right-click to open the folder directory, and put the demo and examples folders into the src folder!



Projects > WZ4827R043 > .pio > libdeps > esp32-s3-devkitc-1-myboard > lvgl > src

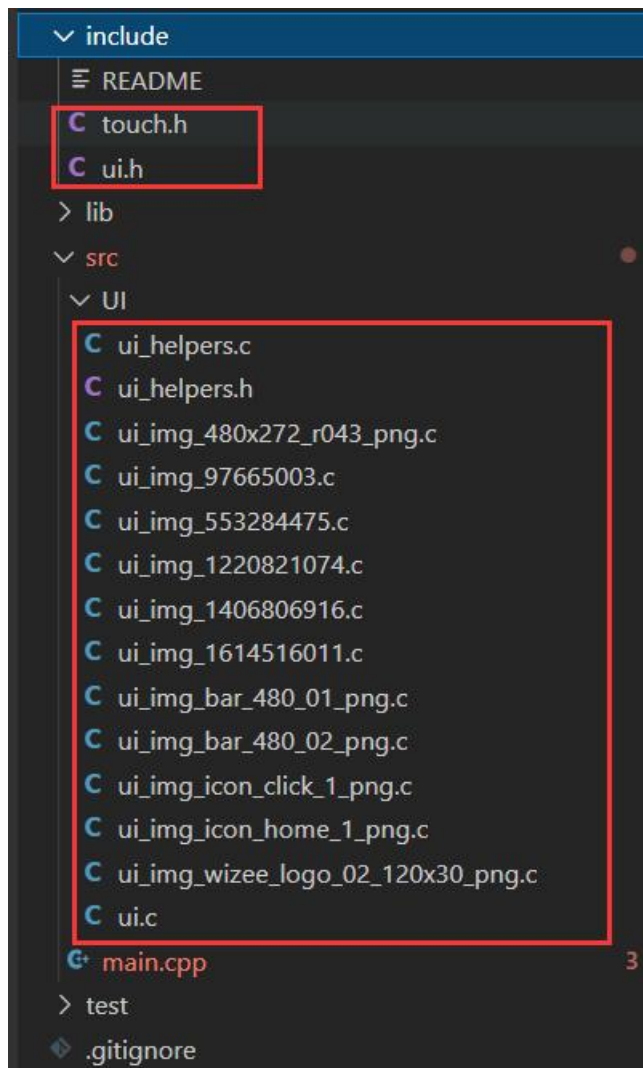
名称	修改日期	类型	大小
core	2023/8/29 16:55	文件夹	
demos	2023/8/29 17:03	文件夹	
draw	2023/8/29 16:55	文件夹	
examples	2023/8/29 17:03	文件夹	
extra	2023/8/29 16:55	文件夹	
font	2023/8/29 16:55	文件夹	
hal	2023/8/29 16:55	文件夹	
misc	2023/8/29 16:55	文件夹	
widgets	2023/8/29 16:55	文件夹	
lv_api_map.h	2023/4/3 16:35	H 文件	2 KB
lv_conf_internal.h	2023/5/4 16:32	H 文件	74 KB
lv_conf_kconfig.h	2023/5/3 15:55	H 文件	7 KB
lvgl.h	2023/4/3 16:35	H 文件	1 KB

Place the lv\_conf.h file under this directory again

PlatformIO > Projects > WZ4827R043 > .pio > libdeps > esp32-s3-devkitc-1-myboard >

名称	修改日期	类型	大小
GFX Library for Arduino	2023/8/29 16:59	文件夹	
lvgl	2023/8/29 16:55	文件夹	
XPT2046_Touchscreen	2023/8/29 16:57	文件夹	
integrity.dat	2023/8/29 16:59	DAT 文件	1 KB
lv_conf.h	2023/8/25 17:25	H 文件	26 KB

Next, let's configure our own UI files (the UI files are generated from the SquareLine Studio)



In the UI folder that will be generated. The c file is placed in the /src folder, and in the generated UI folder. Place the h file in the /include folder  
At this time, we will complete all the configuration, write the code and start compiling the program

```
1 //lvgl
2 #include <lvgl.h>
3 #include <demoes/lv_demos.h>
4 //UI
5 #include "ui.h"
6 static int first_flag = 0;
7 extern int zero_clean;
8 extern int goto_widget_flag;
9 extern int bar_flag;
10 extern lv_obj_t * ui_MENU;
11 extern lv_obj_t * ui_TOUCH;
12 extern lv_obj_t * ui_JIAOZHUN;
13 extern lv_obj_t * ui_Label2;
14 extern lv_obj_t * ui_touch_calibrate;//校准界面
15 static lv_obj_t * ui_Label1;//TOUCH界面Label
16 static lv_obj_t * ui_Label3;//TOUCH界面Label3
17 static lv_obj_t * ui_Label2;//Menu界面进度条Label
18 static lv_obj_t * bar;//Menu界面进度条
19
20 uint16_t touchCalibration_x0 = 300, touchCalibration_x1 = 3600, touchCalibration_y0 = 300, touchCalibration_y1 = 3600;
21 uint8_t touchCalibration_rotate = 1, touchCalibration_invert_x = 2, touchCalibration_invert_y = 0;
22 static int val = 100;
23
24 static uint32_t screenWidth;
25 static uint32_t screenHeight;
26 static lv_disp_draw_buf_t draw_buf;
27 static lv_color_t disp_draw_buf[480 * 272 / 8];
28 static lv_disp_drv_t disp_drv;
29 #include <Arduino_GFX_Library.h>
30 #define TFT_B1 2
31 #define GFX_B1_DE_GFX_B1 // default backlight pin, you may replace OF_GFX_B1 to actual backlight pin
32
33
34 #if defined(DISPLAY_DEV_KIT)
35 Arduino_GFX *lcd = create default Arduino_GFX();
36 #else /* !defined(DISPLAY_DEV_KIT) */
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
1 //lvgl
2 #include <lvgl.h>
3 #include <demoes/lv_demos.h>
4 //UI
5 #include "ui.h"
6 static int first_flag = 0;
7 extern int zero_clean;
8 extern int goto_widget_flag;
9 extern int bar_flag;
10 extern lv_obj_t * ui_MENU;
11 extern lv_obj_t * ui_TOUCH;
12 extern lv_obj_t * ui_JIAOZHUN;
13 extern lv_obj_t * ui_Label2;
14 extern lv_obj_t * ui_touch_calibrate;//校准界面
15 static lv_obj_t * ui_Label1;//TOUCH界面Label
16 static lv_obj_t * ui_Label3;//TOUCH界面Label3
17 static lv_obj_t * ui_Label2;//Menu界面进度条Label
18 static lv_obj_t * bar;//Menu界面进度条
19
20 uint16_t touchCalibration_x0 = 300, touchCalibration_x1 = 3600, touchCalibration_y0 = 300, touchCalibration_y1 = 3600;
21 uint8_t touchCalibration_rotate = 1, touchCalibration_invert_x = 2, touchCalibration_invert_y = 0;
22 static int val = 100;
23
24 static uint32_t screenWidth;
25 static uint32_t screenHeight;
26 static lv_disp_draw_buf_t draw_buf;
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
Build: [==] 16.0% (used 68464 bytes from 327680 bytes)
Flash: [==] 21.8% (used 68469 bytes from 3145728 bytes)
Building .pio/build/esp32-s3-devkitc-1-usbboard/firmware.bin
esptool.py v4.4
Creating esp32s3 image...
Merged 2 ELF sections
Successfully created esp32s3 image.
===== [SUCCESS] Took 165.85 seconds =====
```

Next we began to burn the program, finished!

```
1 //lvgl
2 #include <lvgl.h>
3 #include <demoes/lv_demos.h>
4 //UI
5 #include "ui.h"
6 static int first_flag = 0;
7 extern int zero_clean;
8 extern int goto_widget_flag;
9 extern int bar_flag;
10 extern lv_obj_t * ui_MENU;
11 extern lv_obj_t * ui_TOUCH;
12 extern lv_obj_t * ui_JIAOZHUN;
13 extern lv_obj_t * ui_Label2;
14 extern lv_obj_t * ui_touch_calibrate;//校准界面
15 static lv_obj_t * ui_Label1;//TOUCH界面Label
16 static lv_obj_t * ui_Label3;//TOUCH界面Label3
17 static lv_obj_t * ui_Label2;//Menu界面进度条Label
18 static lv_obj_t * bar;//Menu界面进度条
19
20 uint16_t touchCalibration_x0 = 300, touchCalibration_x1 = 3600, touchCalibration_y0 = 300, touchCalibration_y1 = 3600;
21 uint8_t touchCalibration_rotate = 1, touchCalibration_invert_x = 2, touchCalibration_invert_y = 0;
22 static int val = 100;
23
24 static uint32_t screenWidth;
25 static uint32_t screenHeight;
26 static lv_disp_draw_buf_t draw_buf;
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
Writing at 0x00003c00... (95 %)
Writing at 0x00003400... (100 %)
Wrote 68469 bytes (68907 compressed) at 0x00010000 in 10.0 seconds (effective 549.2 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 36.10 seconds =====
```