

1. 测试平台介绍

开发板：ESP32-WROOM-32E devKit

MCU：ESP32-32E模组

主频：240MHz

2. 引脚连接说明

本模块可以直插到ESP32-32E开发板上，如下图所示：

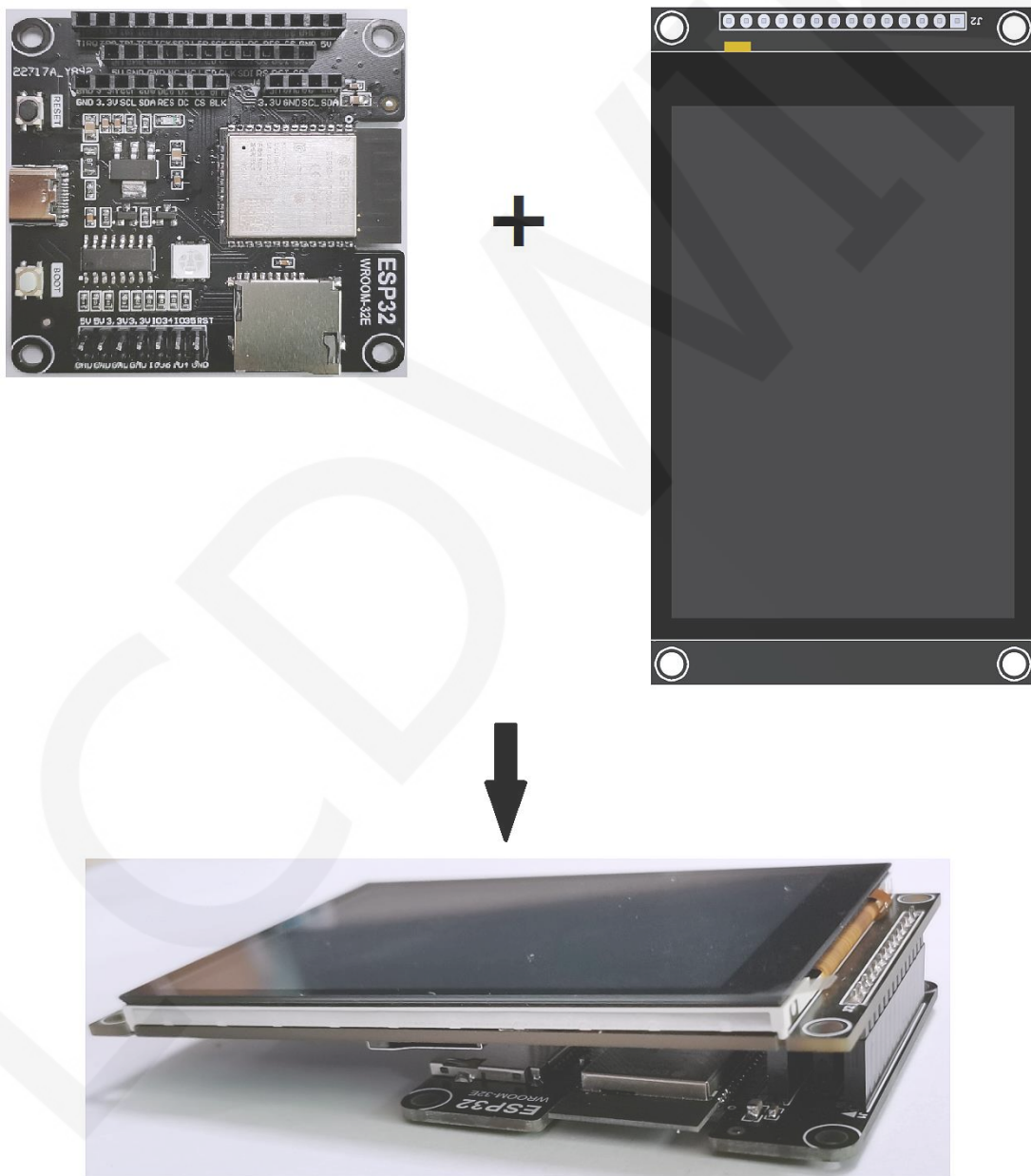


图1 模块直插ESP32-32E开发板

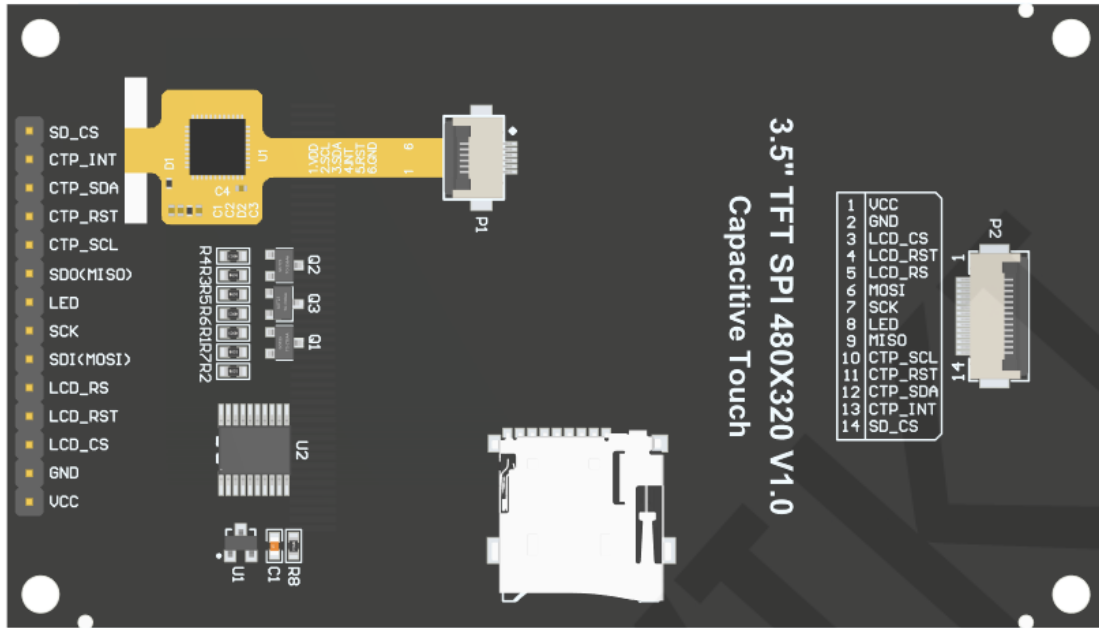


图2 模块背面引脚

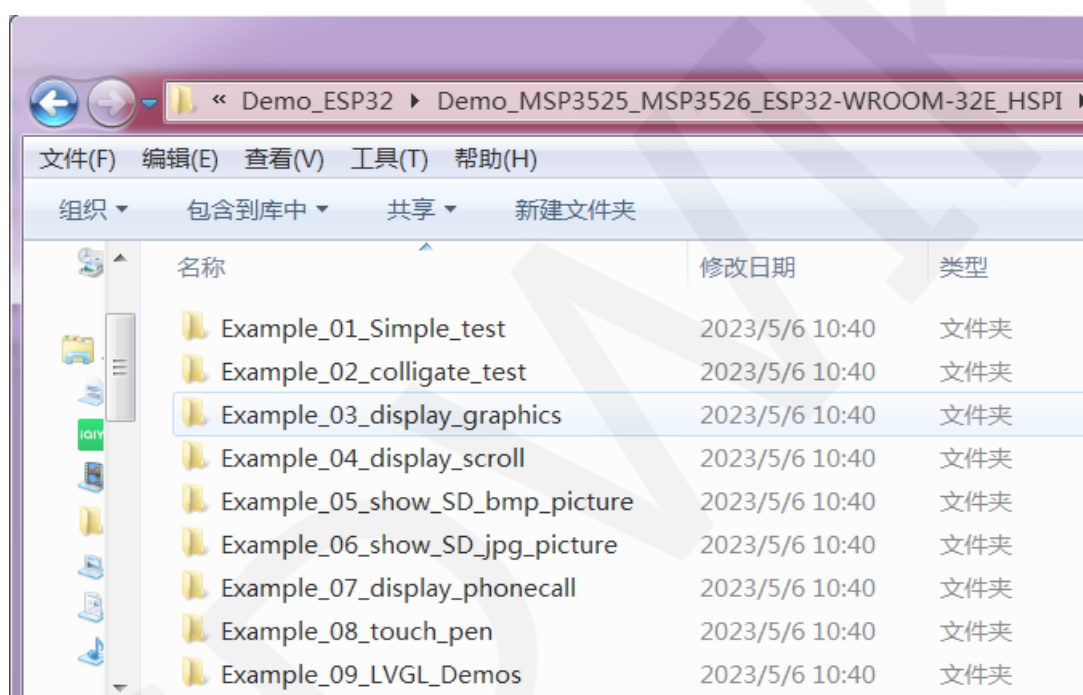
ESP32-32E测试程序引脚直插说明			
序号	模块引脚	对应ESP32-32E开发板接线引脚	备注
1	VCC	5V	液晶屏电源正
2	GND	GND	液晶屏电源地
3	LCD_CS	IO15	液晶屏片选控制信号，低电平有效
4	LCD_RST	IO27	液晶屏复位控制信号，低电平复位
5	LCD_RS	IO2	液晶屏命令/数据选择控制信号 高电平：数据，低电平：命令
6	SDI(MOSI)	IO13	SPI总线写数据信号（SD卡和液晶屏共用）
7	SCK	IO14	SPI总线时钟信号（SD卡和液晶屏共用）
8	LED	IO21	液晶屏背光控制信号（如需控制，请接引脚，如不需控制，可以不接）
9	SDO(MISO)	IO12	SPI总线读数据信号（SD卡和液晶屏共用）
10	CTP_SCL	IO25	电容触摸屏IIC总线时钟信号（无触摸屏的模块不需连接）
11	CTP_RST	IO33	电容触摸屏复位控制信号，低电平复位（无触摸屏的模块不需连接）
12	CTP_SDA	IO32	电容触摸屏IIC总线数据信号（无触摸屏的模块不需连接）

13	CTP_INT	IO39	电容触摸屏IIC总线触摸中断信号，产生触摸时，输入低电平到主控（无触摸屏的模块不需连接）
14	SD_CS	IO22	SD卡片选控制信号，低电平有效（不使用SD卡功能，可以不接）

3. 例程功能说明

本套示例程序使用ESP32硬件HSPI总线，其位于

Demo_MSP3525_MSP3526_ESP32-WROOM-32E_HSPI目录下，如下图所示：



- A、Example_01_Simple_test为刷屏测试程序，此程序不依赖任何软件库；
- B、Example_02_colligate_test为综合测试程序，显示图形、线条并统计程序运行时间；
- C、Example_03_display_graphics为图形显示测试程序，显示各种图形；
- D、Example_04_display_scroll为滚动测试程序，显示文字滚动；
- E、Example_05_show_SD_bmp_picture为BMP图片显示程序，显示SD内BMP格式图片；
- F、Example_06_show_SD_jpg_picture为JPG图片显示程序，显示SD内jpg格式图片；
- G、Example_07_display_phoncall为电话拨号触摸测试程序，通过触摸模拟拨号功能；
- H、Example_08_touch_pen为触摸笔画图测试程序，通过触摸在液晶屏上画画；
- K、Example_09_LVGL_Demos为LVGL示例显示程序，可以体验LVGL强大的UI设计功能。该示例的bin文件已经提取出来，用相应的工具直接烧录就可以使用。

4. 例程使用说明

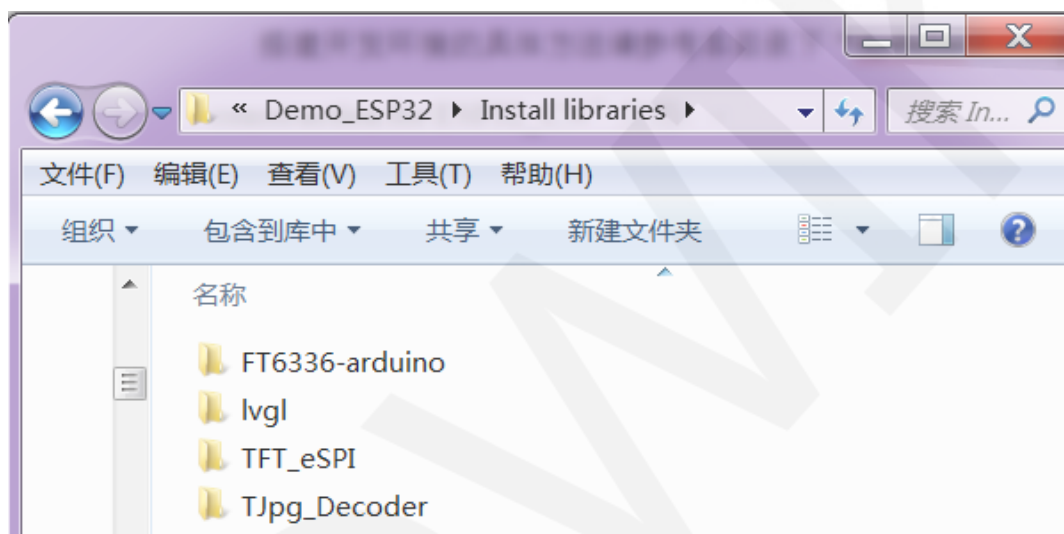
✧ 搭建开发环境

搭建开发环境的具体方法请参考本目录下

“[Arduino_development_environment_construction_for_ESP32_CN](#)”文档。

✧ 安装软件库

开发环境搭建好之后，需要将示例程序使用的软件库复制到工程库目录下，以便示例程序调用。软件库位于**Install libraries**目录下，如下图所示：



其中：

FT6336-arduino为FT6336电容触摸IC的驱动

lvgl为LVGL GUI图形软件库

TFT_eSPI为TFT-LCD液晶屏的Arduino图形库,支持多种平台和多种LCD驱动IC

TJpg_Decoder为Arduino平台JPG格式图片解码库

这些软件库都已经配置好，直接复制到工程库目录下就可以使用。工程库目录默认的路径为C:\Users\Administrator\Documents\Arduino\libraries。也可以更改工程库目录：打开Arduino IDE软件，点击文件->首选项，在弹出的界面里重新设置项目文件夹位置，如下图所示：



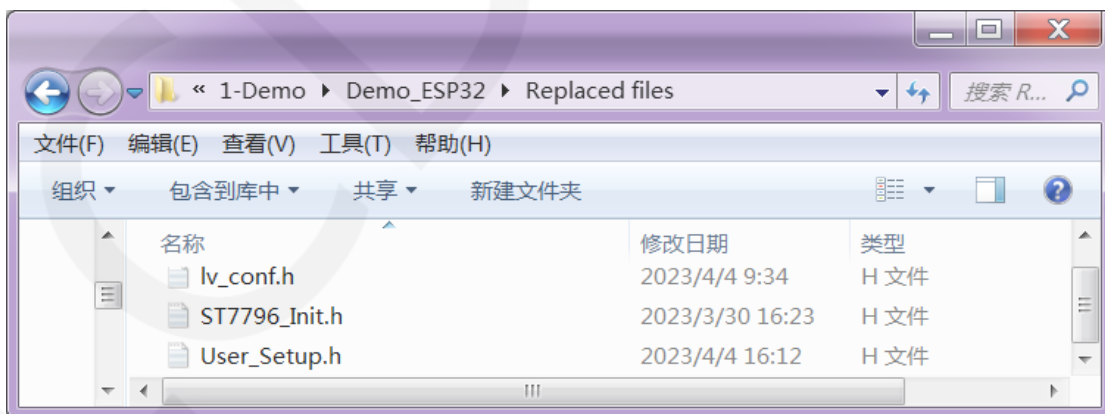
如果不想使用已经配置好的库，那么可以去github下载最新版本的库，再配置（FT6336-arduino除外），下载地址如下：

lvgl: <https://github.com/lvgl/lvgl/tree/release/v8.3>（只能使用v8.x版本，v9.x版本不能使用）

TFT_eSPI: https://github.com/Bodmer/TFT_eSPI

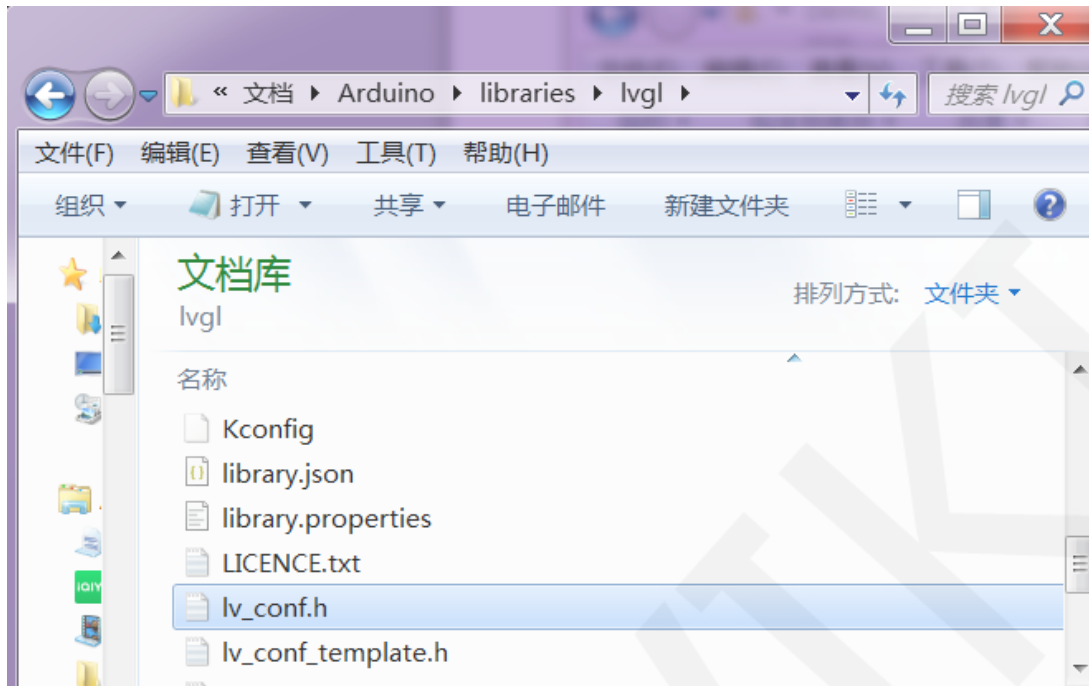
TJpg_Decoder: https://github.com/Bodmer/TJpg_Decoder

库下载完成后，将其解压（为了便于区分，可对解压后的库文件夹进行重命名，如**Install libraries**目录下所示），然后拷贝到工程库目录下。接下来进行库配置，需要替换的文件位于**Replaced files**目录，如下图所示：

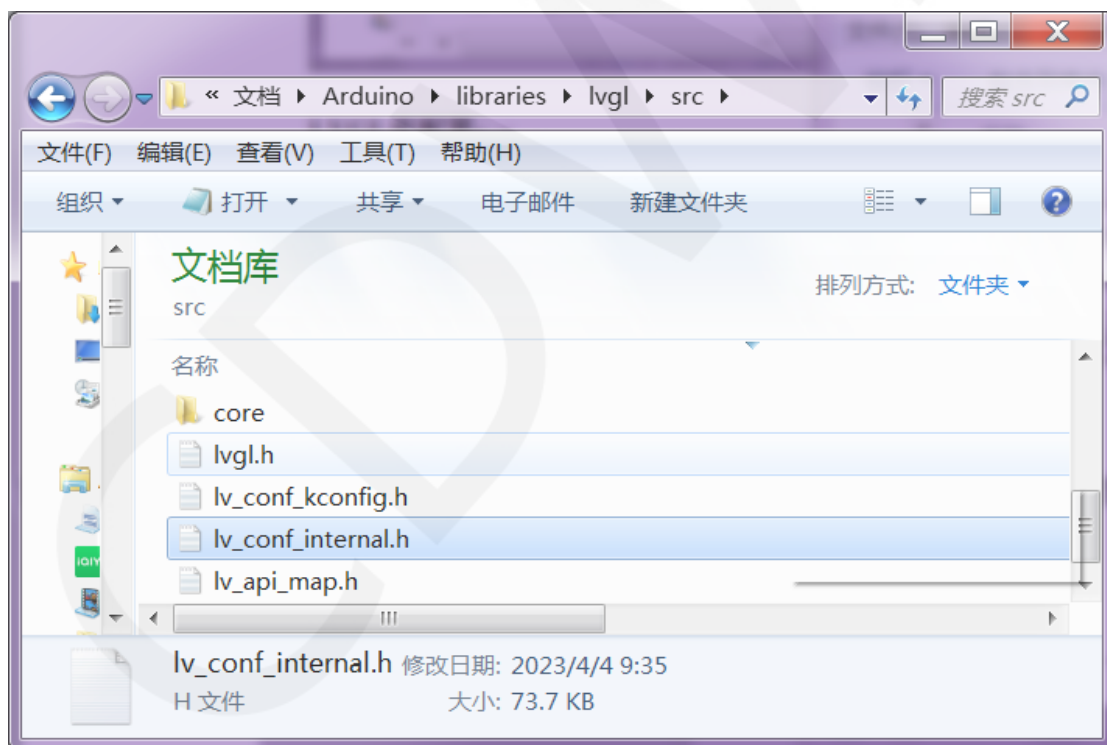


LVGL库配置：

将**Replaced files**目录下的**lv_conf.h**文件拷贝到工程库目录下lvgl库的顶层目录，如下图所示：



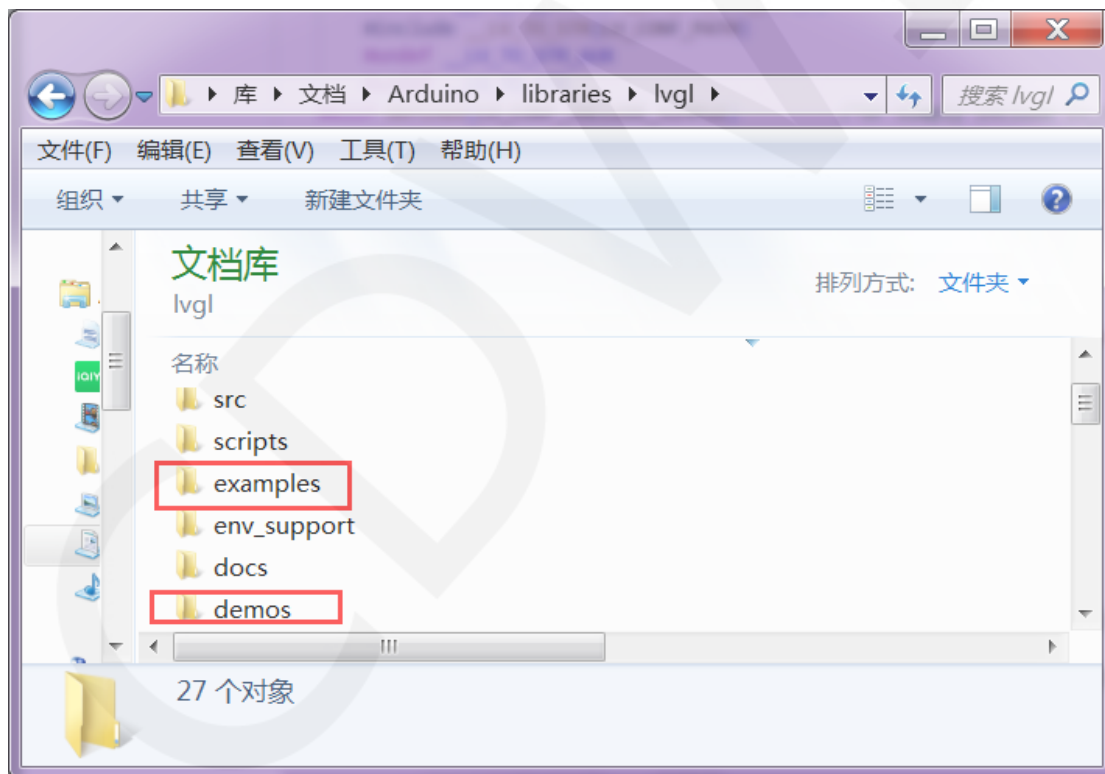
打开工程库目录下lvgl库src目录下的lv_conf_internal.h文件，如下图所示：



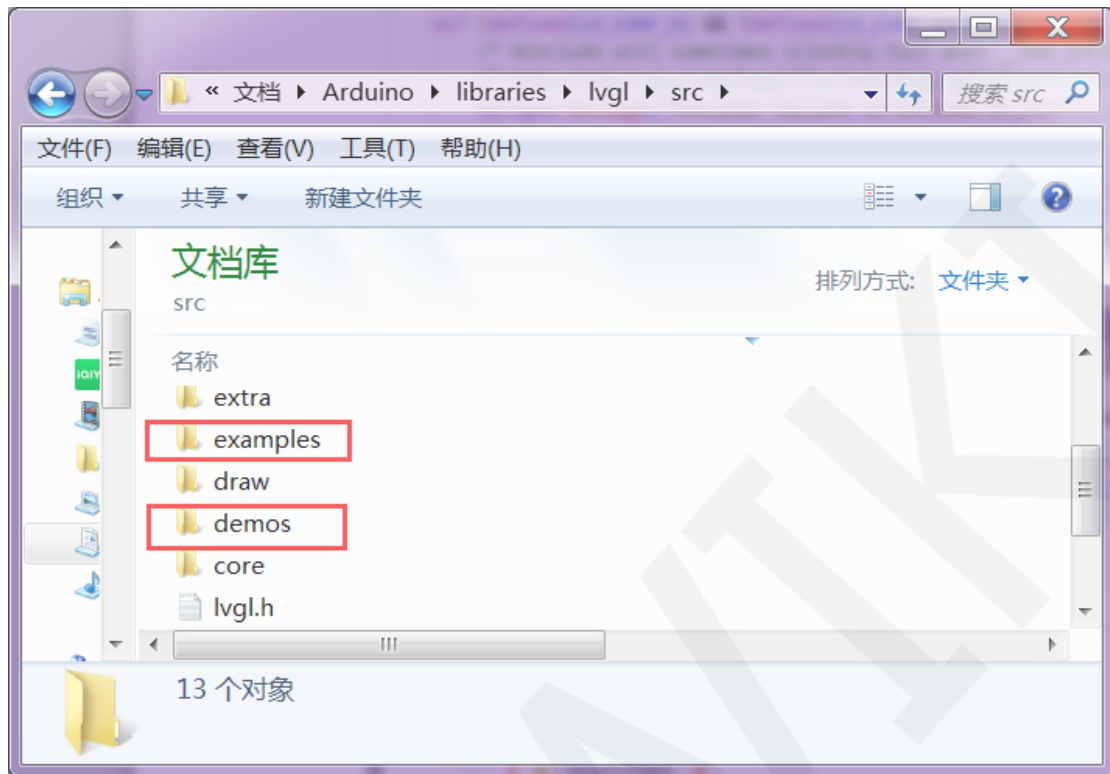
打开文件后，将第41行内容按如下图所示修改（由“`../lv_conf.h`”修改为“`./lv_conf.h`”），修改完成后保存。

```
/*If lv_conf.h is not skipped include it*/
#ifndef LV_CONF_SKIP
#ifdef LV_CONF_PATH /*If there is a path defined for lv_conf.h i
#define __LV_TO_STR_AUX(x) #x
#define __LV_TO_STR(x) __LV_TO_STR_AUX(x)
#include __LV_TO_STR(LV_CONF_PATH)
#undef __LV_TO_STR_AUX
#undef __LV_TO_STR
#elif defined(LV_CONF_INCLUDE_SIMPLE) /*Or simply include lv_conf.h is enabled*/
#include "lv_conf.h"
#else
#include "../lv_conf.h" /*Else assume lv_conf.h is next to the lvgl fo
#endif
#if !defined(LV_CONF_H) && !defined(LV_CONF_SUPPRESS_DEFINE_CHECK)
/* #include will sometimes silently fail when __has_include is used */
/* https://gcc.gnu.org/bugzilla/show_bug.cgi?id=80753 */
#pragma message("Possible failure to include lv_conf.h, please read the comment in th
#endif
#endif
```

将工程库目录下lvgl库下的**examples**和**demos**两个目录拷贝到lvgl库下的src目录里，此两个目录在lvgl库如下图所示：



拷贝后的目录状态:



TFT_eSPI库配置:

首先将工程库目录下TFT_eSPI库顶层目录的**User_Setup.h**文件重命名为**User_Setup_bak.h**, 然后将**Replaced files**目录下的**User_Setup.h**文件拷贝到工程库目录下TFT_eSPI库顶层目录, 如下图所示:



首先将工程库目录下TFT_eSPI库TFT_Drivers目录下的ST7796_Init.h重命名为ST7796_Init_bak.h，然后将Replaced files目录下的ST7796_Init.h拷贝到工程库目录下TFT_eSPI库TFT_Drivers目录，如下图所示：



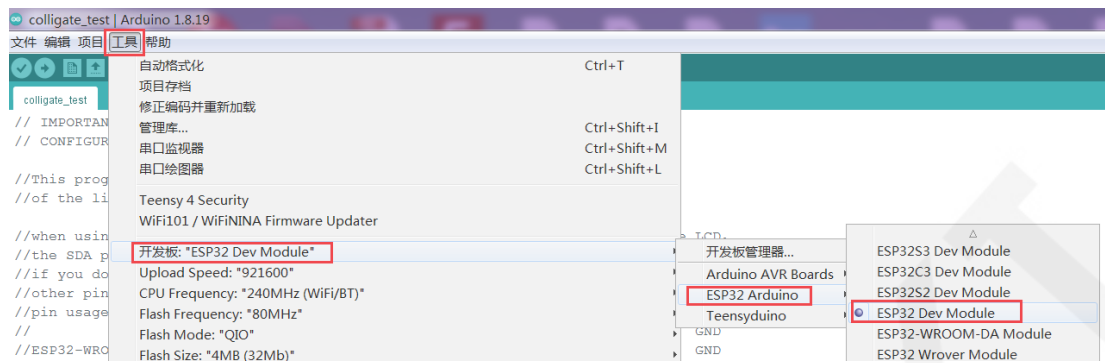
✧ 编译并运行程序

库安装完成之后，就可以进行示例程序编译及运行了，步骤如下：

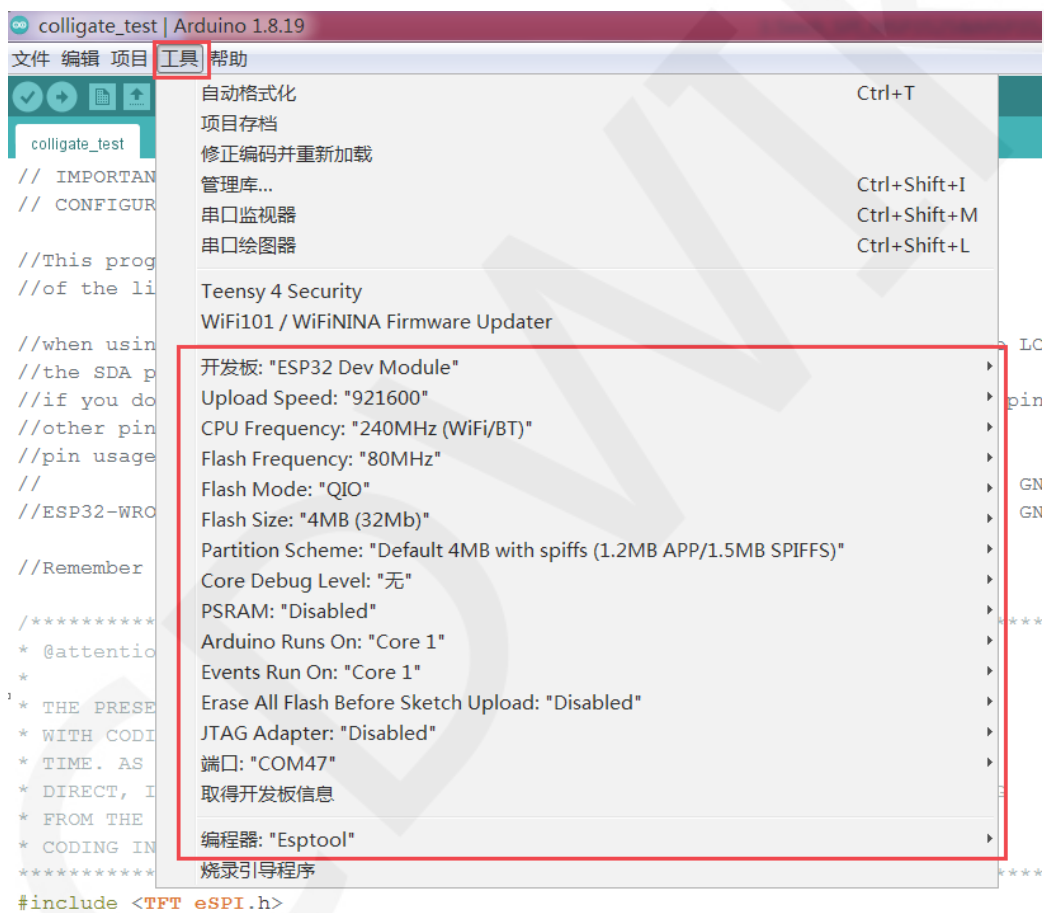
- A、将显示模块直插到 ESP32 开发板，将开发板连接 PC 机上电；
- B、打开 **Demo_ESP32-WROOM-32E_HSPI** 目录下任意一个示例程序，如下图所示（这里以 colligate_test 测试程序为例）：



C、打开示例程序后，选择 ESP32 设备，如下图所示：



D、进行 ESP32 Flash、PSRAM、端口等配置，如下图所示：



E、点击上传按钮进行程序编译和下载，如下图所示：



```
colligate_test | Arduino 1.8.19
文件 编辑 项目 工具 帮助
colligate_test
// IMPORTANT: LCDWIKI_SPI LIBRARY MUST BE SPECIFICALLY
// CONFIGURED FOR EITHER THE TFT SHIELD OR THE BREAKOUT BOARD.

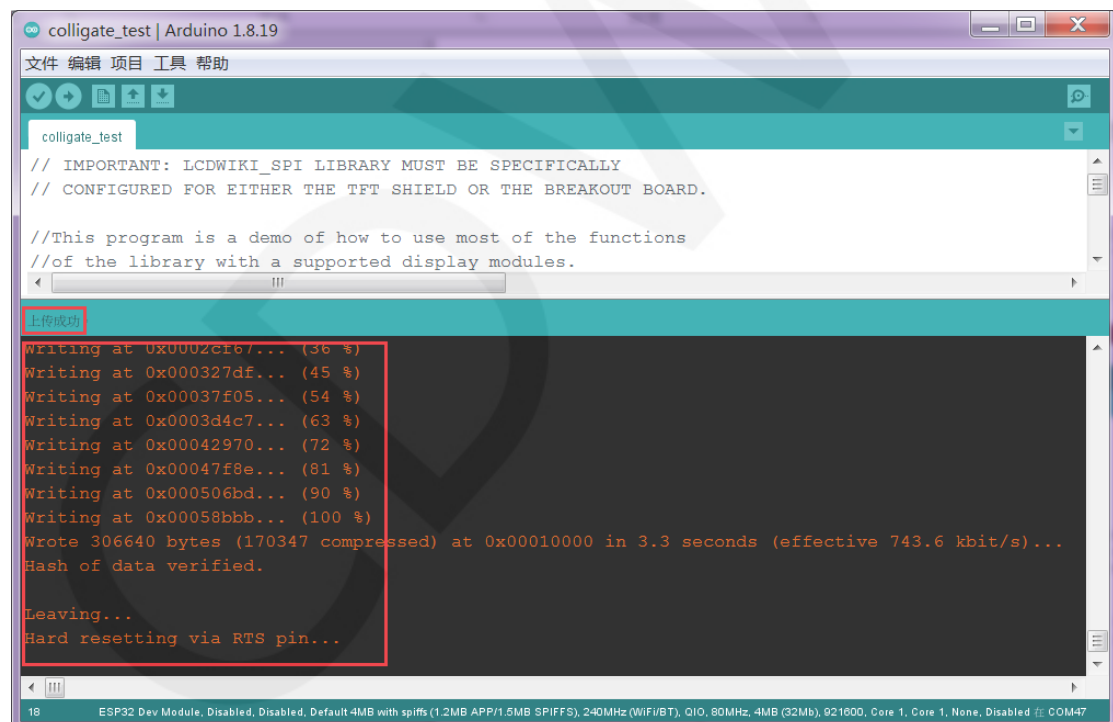
//This program is a demo of how to use most of the functions
//of the library with a supported display modules.

//when using the BREAKOUT BOARD only and using these hardware spi lines to the LCD,
//the SDA pin and SCK pin is defined by the system and can't be modified.
//if you don't need to control the LED pin,you can set it to 3.3V and set the pin definition to
//other pins can be defined by yourself,for example
//pin usage as follow:
//          CS DC/RS  RESET  SDI/MOSI  SCK  SDO/MISO  LED   VCC   GND
//ESP32-WROOM-32E:  15   2    27    13     14    12     21    5V   GND

//Remember to set the pins to suit your display module!

/*****
 * @attention|
 *
 * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY ATMS AT PROVIDING CUSTOMERS
```

F、出现如下提示则说明程序编译完成并下载成功，且已经运行：



```
colligate_test | Arduino 1.8.19
文件 编辑 项目 工具 帮助
colligate_test
// IMPORTANT: LCDWIKI_SPI LIBRARY MUST BE SPECIFICALLY
// CONFIGURED FOR EITHER THE TFT SHIELD OR THE BREAKOUT BOARD.

//This program is a demo of how to use most of the functions
//of the library with a supported display modules.

上传成功
Writing at 0x0002c167... (36 %)
Writing at 0x000327df... (45 %)
Writing at 0x00037f05... (54 %)
Writing at 0x0003d4c7... (63 %)
Writing at 0x00042970... (72 %)
Writing at 0x00047f8e... (81 %)
Writing at 0x000506bd... (90 %)
Writing at 0x00058bbb... (100 %)
Wrote 306640 bytes (170347 compressed) at 0x00010000 in 3.3 seconds (effective 743.6 kbit/s)...
Hash of data verified.

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

18 ESP32 Dev Module, Disabled, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled in COM47
```

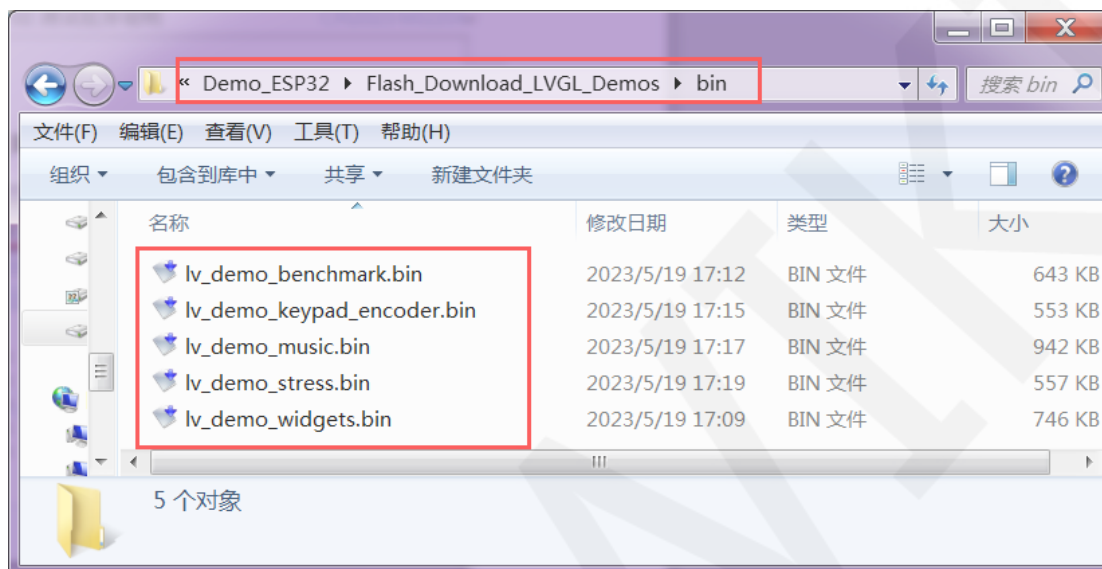
G、如果显示模块有内容显示，则说明程序运行成功。

✧ LVGL示例bin文件烧录

由于LVGL示例程序编译时间较长,现在已经将编译好的bin文件提取出来,用的flash download 工具直接烧录就可以使用。

Bin 文件位于

Demo_ESP32\Flash_Download_LVGL_Demos\bin 目录下,如下图所示:



使用 **Demo_ESP32\Flash_Download_LVGL_Demos** 目录下的 **flash_download_tool** 工具就可以烧录了,如下图所示:

