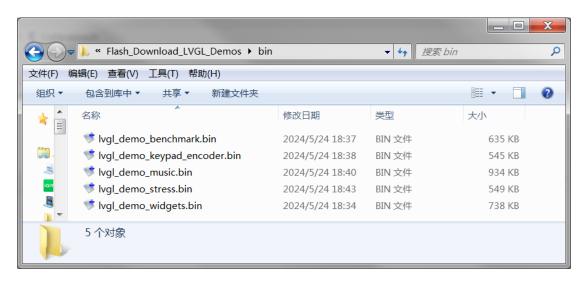
1. 例程功能说明

- 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_jpg_picture为JPG图片显示程序,显示SD内jpg格式图片;
- F、Example_06_display_phonecall为电话拨号触摸测试程序,通过触摸模拟拨号功能;
- G、Example_07_touch_pen为触摸笔画图测试程序,通过触摸在液晶屏上画画;
- H、Example_08_LVGL_Demos为LVGL示例显示程序,可体验LVGL强大的UI设计功能。
- I、Flash_Download_LVGL_Demos文件夹包含已经编译完成的LVGL示例程序的bin文件和flash烧录工具。bin文件可使用flash烧录工具直接烧录,其位于如下目录:



- J、Install libraries文件夹包含已经配置好的软件库,可以拷贝到工程库目录下直接使用;
- K、Replaced files文件夹包含该显示模块的LVGL库的配置文件和TFT_eSPI库的引脚配置文件以及LCD初始化文件。可以使用这些文件去替换库里相关的文件。

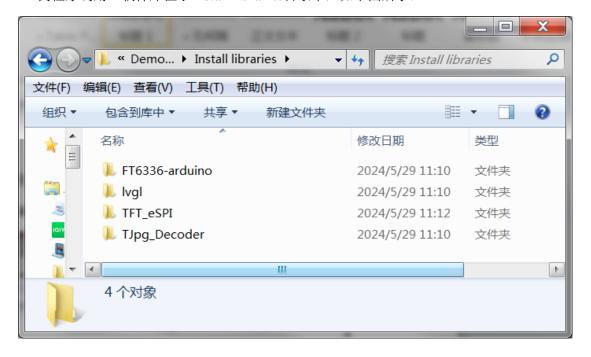
2. 例程使用说明

♦ 搭建开发环境

开发环境搭建方法可参考官方网站或进行网上查阅。

♦ 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拷贝到工程库目录下,以便示例程序调用。软件库位于Install libraries目录下,如下图所示:

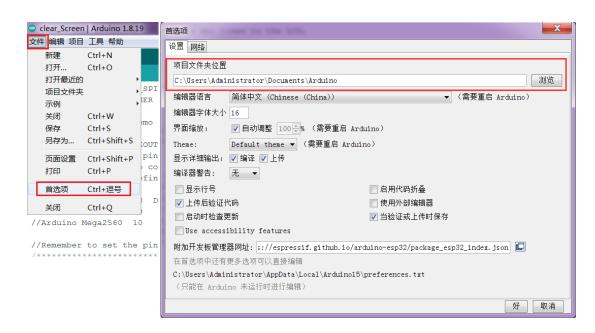


其中:

FT6336-arduino为FT6336电容触摸IC的驱动(如果产品不带触摸则不需使用)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

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:



如果不想使用已经下载好的库,那么可以去github下载最新版本的库

(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

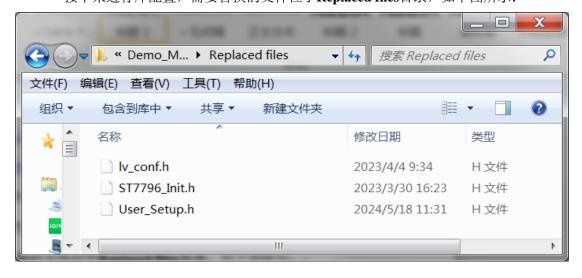
本不能使用)

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:

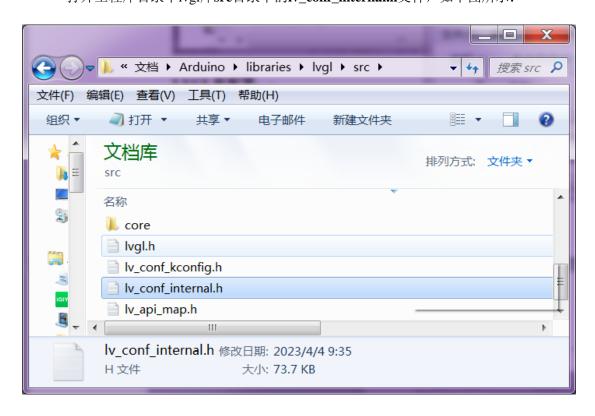


LVGL库配置:

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:

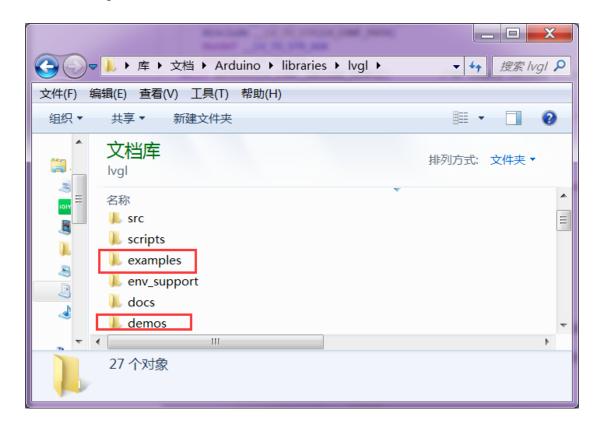


打开文件后,将第41行内容按如下图所示修改(由"../../lv_conf.h"修改为"../lv_conf.h"),

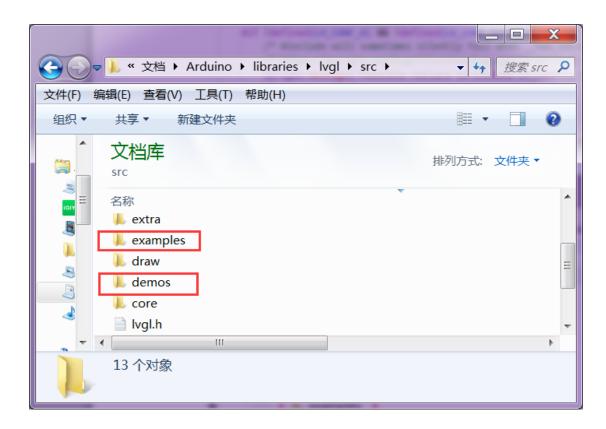
修改完成后保存。

```
/*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 u
        #define __LV_TO_STR_AUX(x) #x
        \label{eq:lv_to_str_aux} \text{\#define } \underline{\quad LV\_T0\_STR}(x) \ \underline{\quad LV\_T0\_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"
        #include "../lv conf.h"
                                                    /*Else assume lv_conf.h is next to the lvgl fo.
    #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:



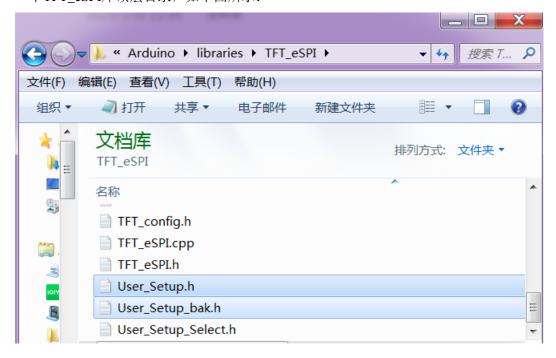
拷贝后的目录状态:



TFT_eSPI库配置:

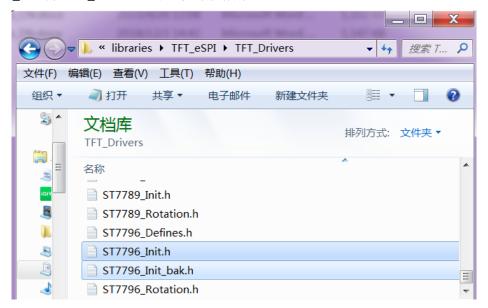
首先将工程库目录下TFT_eSPI库项层目录的User_Setup.h文件重命名为

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库项层目录,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

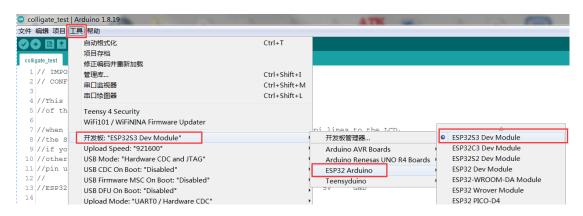
ST7796_Init_bak.h, 然后将Replaced files目录下的ST7796_Init.h拷贝到工程库目录下TFT eSPI库TFT Drivers目录,如下图所示:



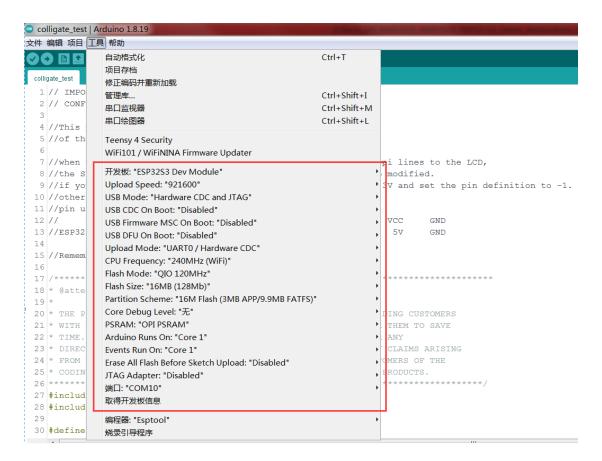
♦ 编译并运行程序

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、将显示模块接到 ESP32-S3 开发板上,将开发板连接 PC 机上电;
- B、打开目录下任意一个示例程序(这里以 colligate_test 测试程序为例):
- C、打开示例程序后,选择 ESP32-S3 设备,如下图所示:



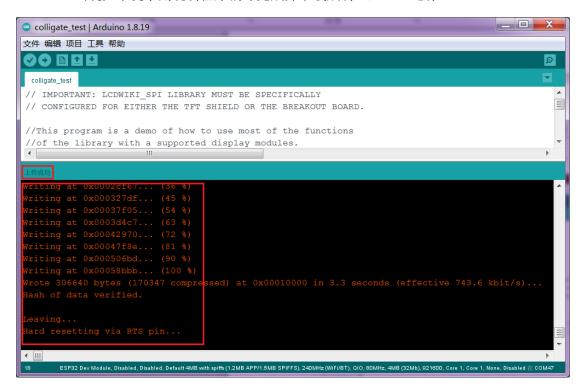
D、进行 ESP32-S3 设备 Flash、PSRAM、端口等配置,如下图所示(此配置为 N16R8 型号,如果是其他型号则需按照实际情况配置):



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
 🗸 🕩 🛅 査 🛂 上传
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 youself, for example
//pin usage as follow:
                     CS DC/RS RESET SDI/MOSI SCK SDO/MISO LED
                                                                        VCC
                                                                                GND
//ESP32-WROOM-32E:
                                  27
                   15
                          2
                                          13
                                                 14
                                                         12
                                                                21
                                                                        5v
                                                                                GND
//Remember to set the pins to suit your display module!
* @attention
^{\dagger}\star the present firmware which is for guidance only aims 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:



G、如果显示模块有内容显示,则说明程序运行成功。