E32R28T&E32N28T
2.8 寸 ESP32-32E
示例程序说明

目　录

1. 软件和硬件平台说明

模块：2.8寸ESP32-32E显示模块，拥有240x320分辨率，采用ILI9341屏驱IC。

模块主控：ESP32-WROOM-32E模组，最高主频240MHz，支持2.4G WIFI+蓝牙。

Arduino IED版本：1.8.19版本和2.3.2版本。

ESP32 Ardunio核心库软件版本：2.0.17版本和3.0.3版本。

2. 引脚分配说明

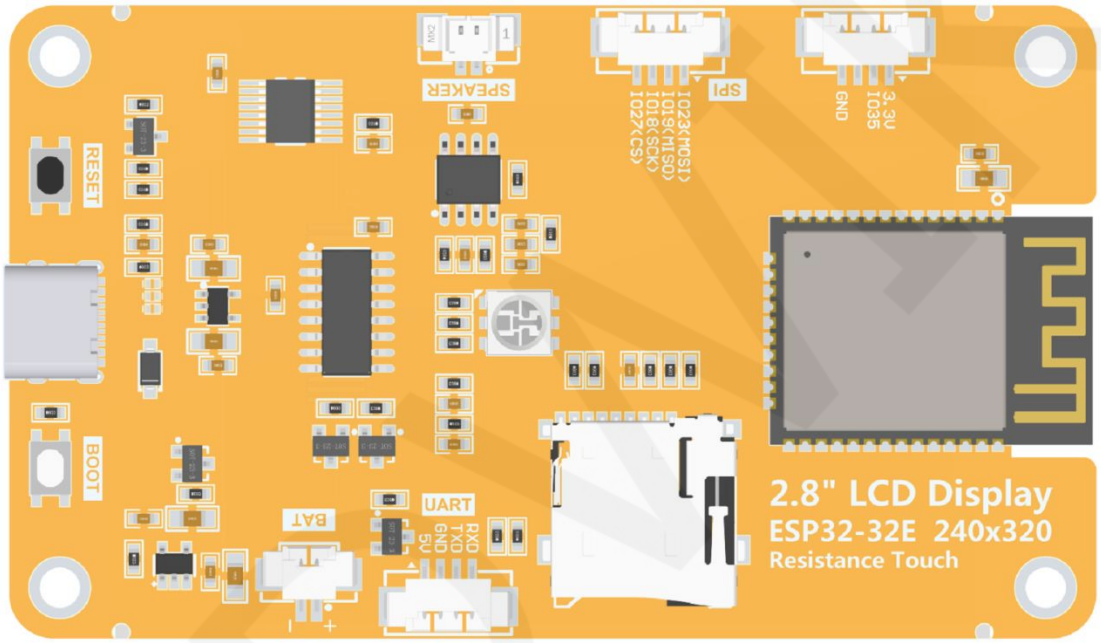

图2.1 2.8寸ESP32-32E显示模块背面图

2.8寸ESP32显示模块主控为ESP32-32E，其连接板载外设的GPIO分配如下表格所示：

ESP32-32E引脚分配说明			
板载设备	板载设备引脚	ESP32-32E连接引脚	说明
LCD	TFT_CS	IO15	液晶屏片选控制信号，低电平有效
	TFT_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
	TFT_SCK	IO14	液晶屏SPI总线时钟信号
	TFT_MOSI	IO13	液晶屏SPI总线写数据信号
	TFT_MISO	IO12	液晶屏SPI总线读数据信号

	TFT_RST	EN	液晶屏复位控制信号，低电平复位（和ESP32-32E主控共用复位引脚）	
	TFT_BL	IO21	液晶屏背光控制信号（高电平点亮背光，低电平关闭背光）	
RTP	TP_SCK	IO25	电阻触摸屏SPI总线时钟信号	
	TP_DIN	IO32	电阻触摸屏SPI总线写数据信号	
	TP_DOUT	IO39	电阻触摸屏SPI总线读数据信号	
	TP_CS	IO33	电阻触摸屏片选控制信号，低电平有效	
	TP_IRQ	IO36	电阻触摸屏触摸中断信号，产生触摸时，输入低电平到主控	
LED	LED_RED	IO22	红色LED灯	RGB三色LED灯，共阳极，低电平点亮，高电平关闭。
	LED_GREEN	IO16	绿色LED灯	
	LED_BLUE	IO17	蓝色LED灯	
SDCARD	SD_CS	IO5	SD卡片选信号，低电平有效	
	SD_MOSI	IO23	SD卡SPI总线写数据信号	
	SD_SCK	IO18	SD卡SPI总线时钟信号	
	SD_MISO	IO19	SD卡SPI总线读数据信号	
BATTERY	BAT_ADC	IO34	电池电压ADC值获取信号（输入）	
Audio	Audio_ENABLE	IO4	音频使能信号，低电平使能，高电平禁止	
	Audio_DAC	IO26	音频信号DAC输出信号	
KEY	BOOT_KEY	IO0	下载模式选择按键（按住该按键上电，然后松开就会进入下载模式）	
	RESET_KEY	EN	ESP32-23E复位按键，低电平复位（和液晶屏复位共用）	
Serial Port	RX0	RXD0	ESP32-32E串口接收信号	
	TX0	TXD0	ESP32-32E串口发送信号	
POWER	TYPE-C_POWER	/	Type-C电源接口，接入5V电压。	

表2.1 ESP32-32E板载外设引脚分配说明

3. 示例程序使用说明

3.1. 搭建ESP32 Arduino开发环境

ESP32 Arduino开发环境搭建详细说明见资料包里“**ESP32_Arduino_IDE1开发环境搭建**”和“**ESP32_Arduino_IDE2开发环境搭建**”说明文档。

3.2. 安装第三方软件库

开发环境搭建好之后，首先需要安装示例程序使用的第三方软件库。步骤如下：

A、打开资料包里“**1-示例程序_Demo\Arduino\Install libraries**”目录，找到第三方软件库，如下图所示：

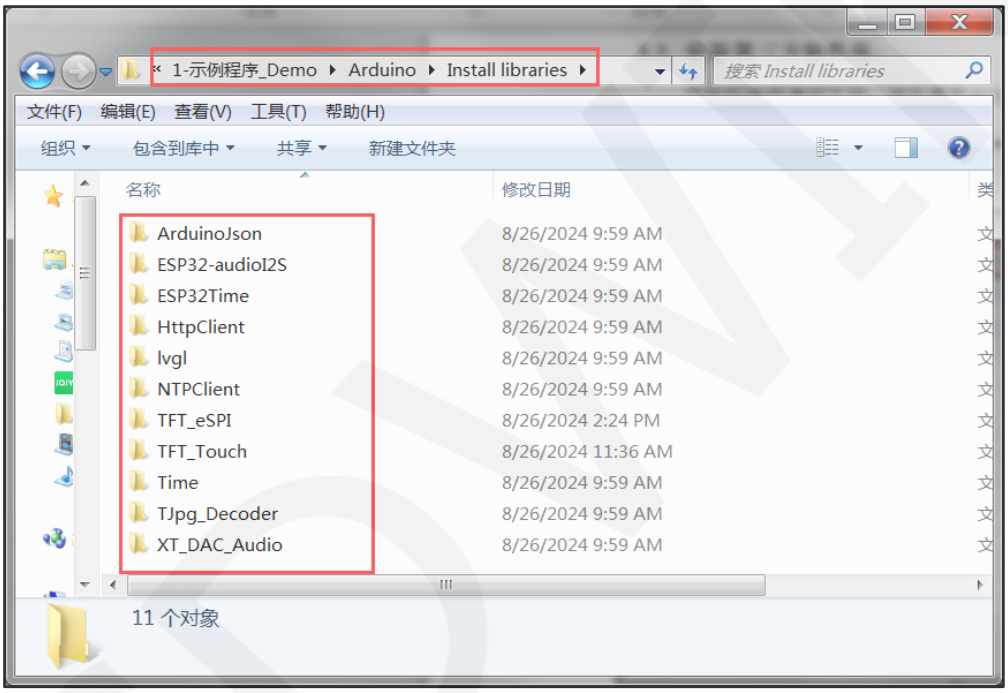

图3.1 示例程序第三方软件库

其中：

ArduinoJson：Arduino和物联网的C++ JSON软件库。

ESP32-audioI2S：ESP32的音频解码软件库，使用ESP32的I2S总线，通过外部音频设备播放SD卡中mp3、m4a以及mav等格式的音频文件。

ESP32Time：用于在ESP32板上设置和检索内部RTC时间的Arduino软件库

HttpClient：与Arduino的web服务器进行交互的Http客户端软件库。

lvgl：高度可裁剪、低资源占用、界面美观且易用的嵌入式系统图形软件库。

NTPClient：连接 NTP 服务器的 NTP 客户端软件库。

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

TFT_Touch：电阻触摸屏驱动芯片XPT2046的Arduino驱动软件库。

Time：为Arduino提供计时功能的软件库。

TJpg_Decoder：Arduino平台JPG格式图片解码库，可将SD卡或者Flash中的JPG文件解码然后显示到LCD上。

XT_DAC_Audio：ESP32 XTronical DAC音频软件库，可支持WAV格式音频文件。

B、将这些软件库拷贝到项目文件夹的库目录下。项目文件夹的库目录默认为“**C:\Users\Administrator\Documents\Arduino\libraries**”（红色部分为电脑的实际用户名）。如果修改了项目文件夹路径，则需拷贝到修改之后的项目文件夹库目录里。

C、第三方软件库安装完成后，就可以打开示例程序使用了。

在第三方软件库里lvgl和TFT_eSPI软件库在使用之前需要配置，资料包里的软件库已经配置好了，可以直接使用。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，步骤如下：

A、找到github的下载地址进行下载，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

附上其他不需要配置的软件包下载地址：

ArduinoJson：https://github.com/bblanchon/ArduinoJson.git

ESP32Time：https://github.com/fbiego/ESP32Time

HttpClient：http://github.com/amcewen/HttpClient

NTPClient：https://github.com/arduino-libraries/NTPClient.git

Time：https://github.com/PaulStoffregen/Time

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

TFT_Touch : https://github.com/Bodmer/TFT_Touch

B、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），然后拷贝到项目文件夹库目录下（默认为“**C:\Users\Administrator\Documents\Arduino\libraries**”（红色部分为电脑的实际用户名））。接下来进行库配置，打开资料包里的“**1-示例程序_Demo**

\Arduino\Replaced files”目录，找到替换文件，如下图所示：

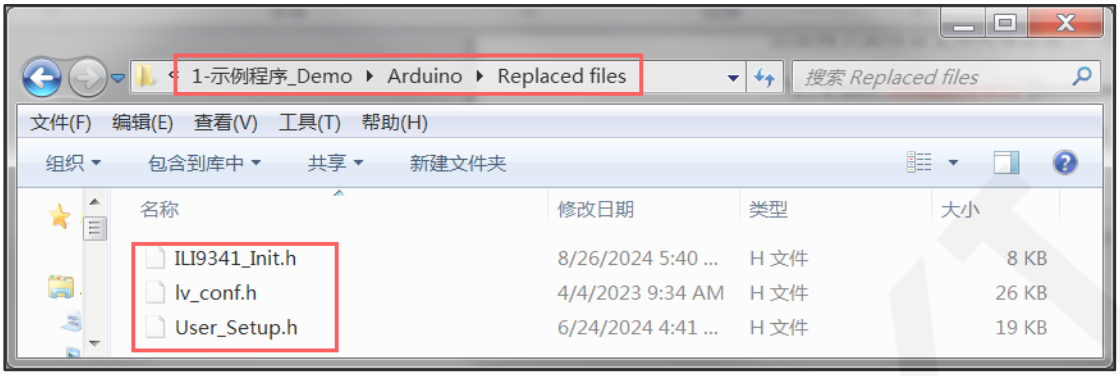

图3.2 第三方软件库替换文件

C、**配置LVGL库：**

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：

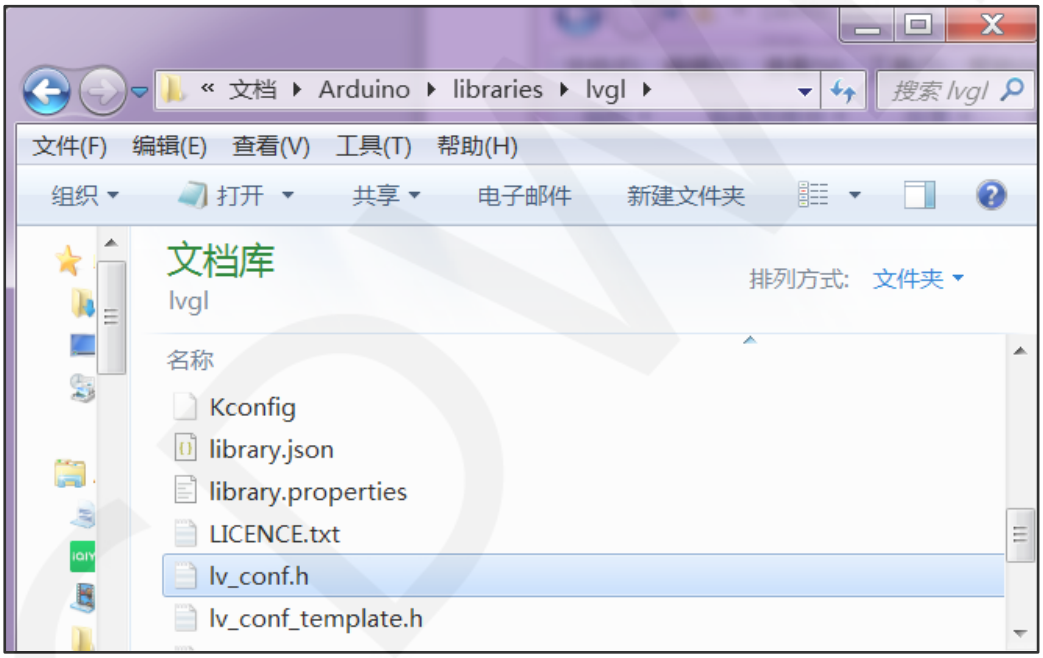

图3.3 配置LVGL库1

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：

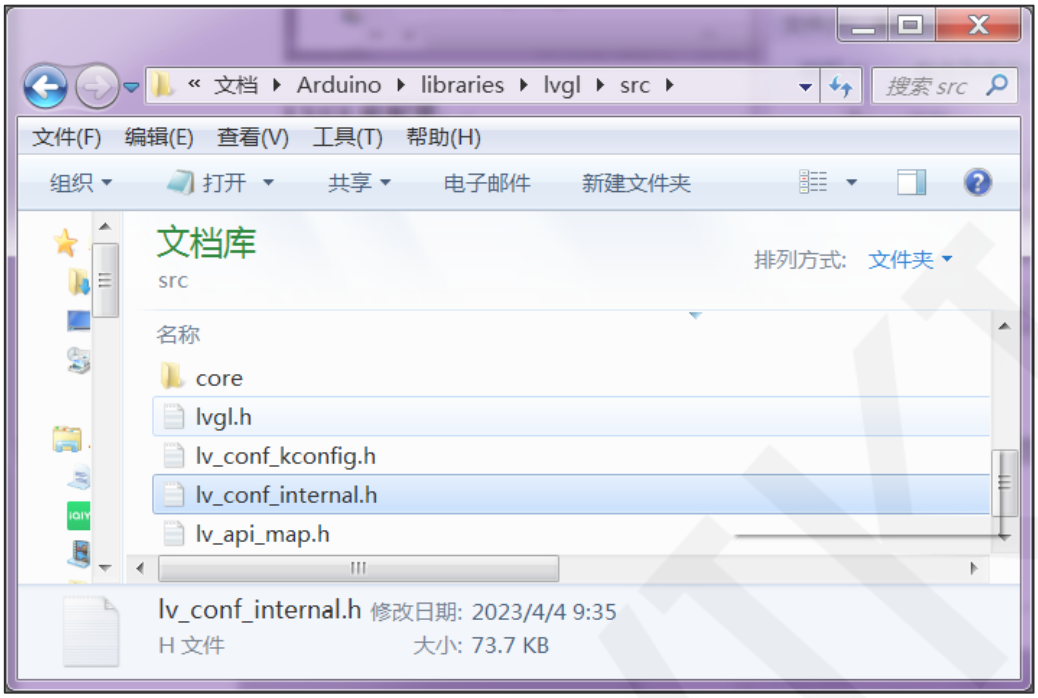

图3.4 配置LVGL库2

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
        #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
```

图3.5 配置LVGL库3

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：

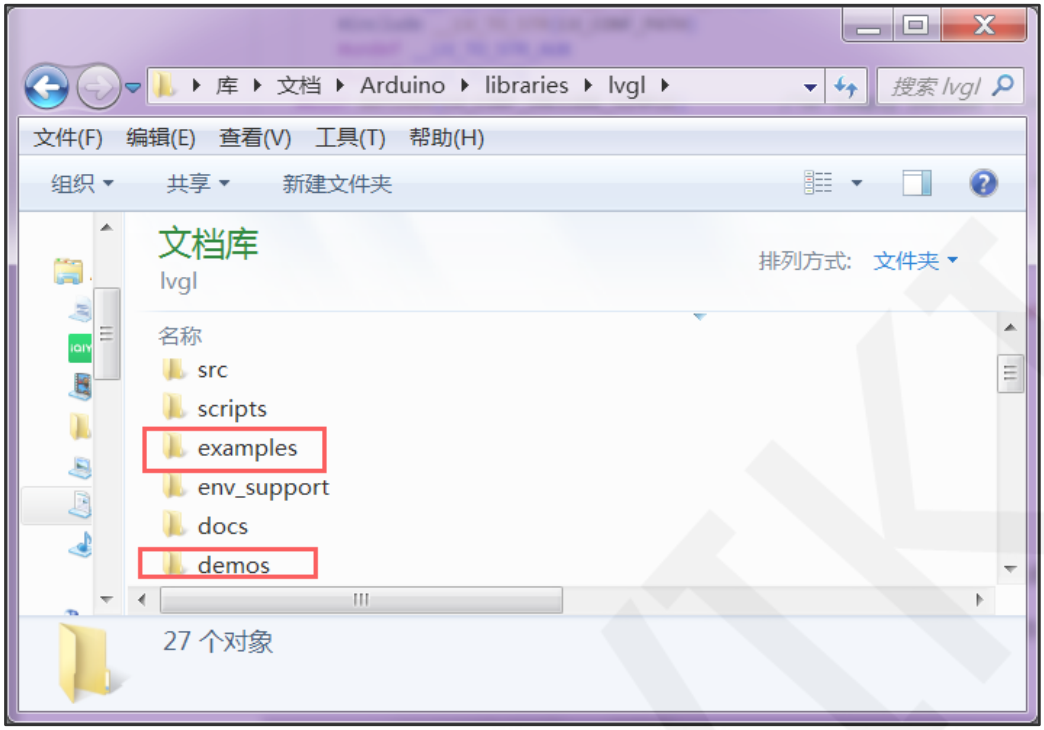

图3.6 配置LVGL库4

拷贝后的目录状态：

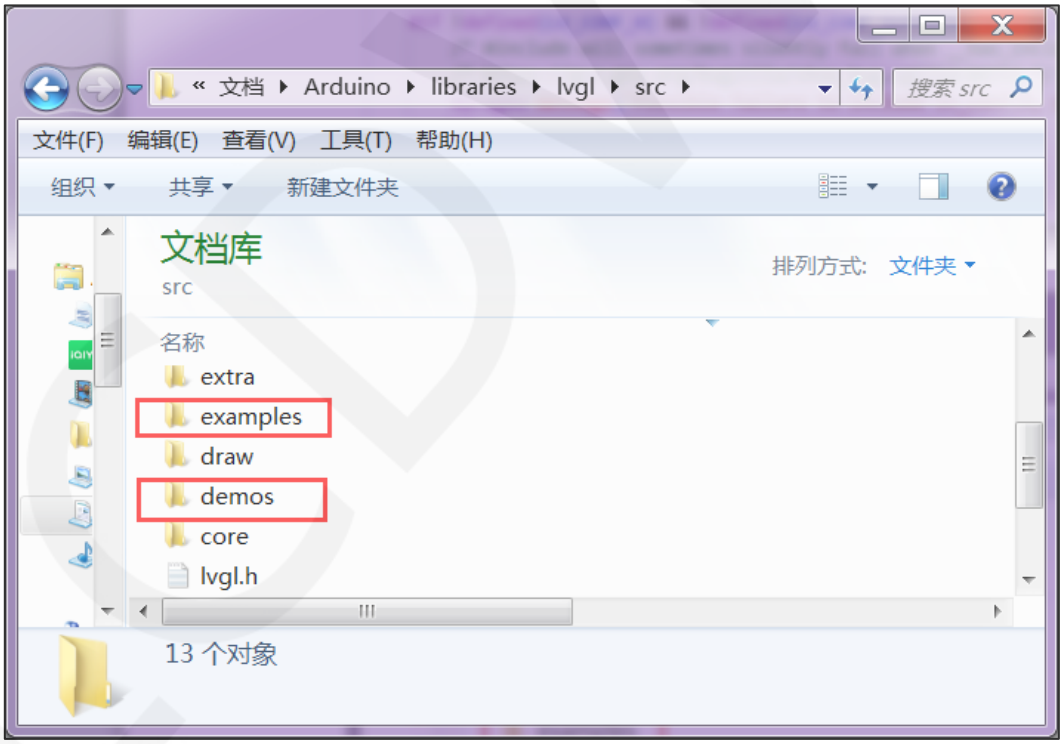

图3.7 配置LVGL库5

D、**配置TFT_eSPI库：**

首先将项目文件夹库目录下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：

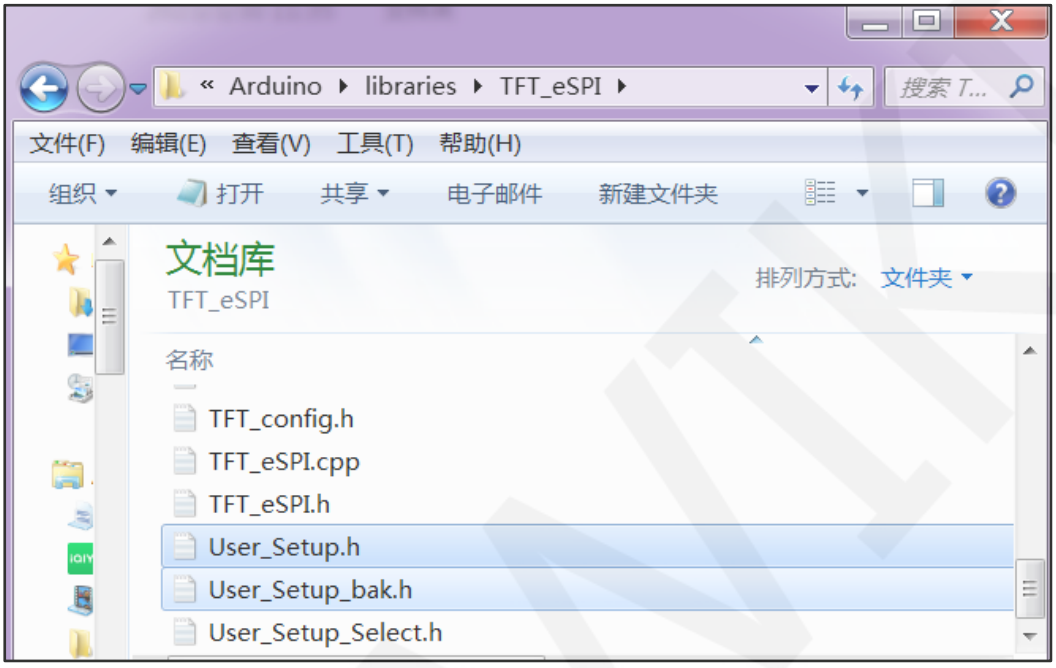

图3.8 配置TFT_eSPI库1

接下来将项目文件夹目录下TFT_eSPI库TFT_Drivers目录下的**ILI9341_Init.h**重命名为**ILI9341_Init_bak.h**，然后将**Replaced files**目录下的**ILI9341_Init.h**拷贝到项目文件夹库目录下TFT_eSPI库**TFT_Drivers**目录，如下图所示：

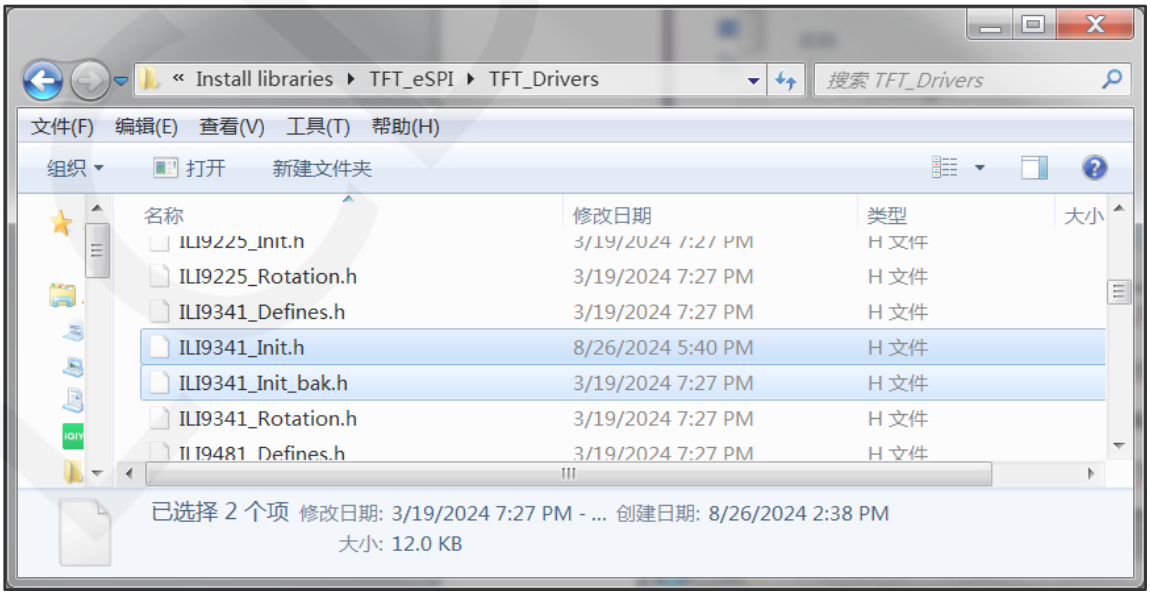

图3.9 配置TFT_eSPI库2

3.3. 示例程序使用说明

示例程序位于资料包的“**1-示例程序_Demo \Arduino\demos**”目录下，如下图所示：

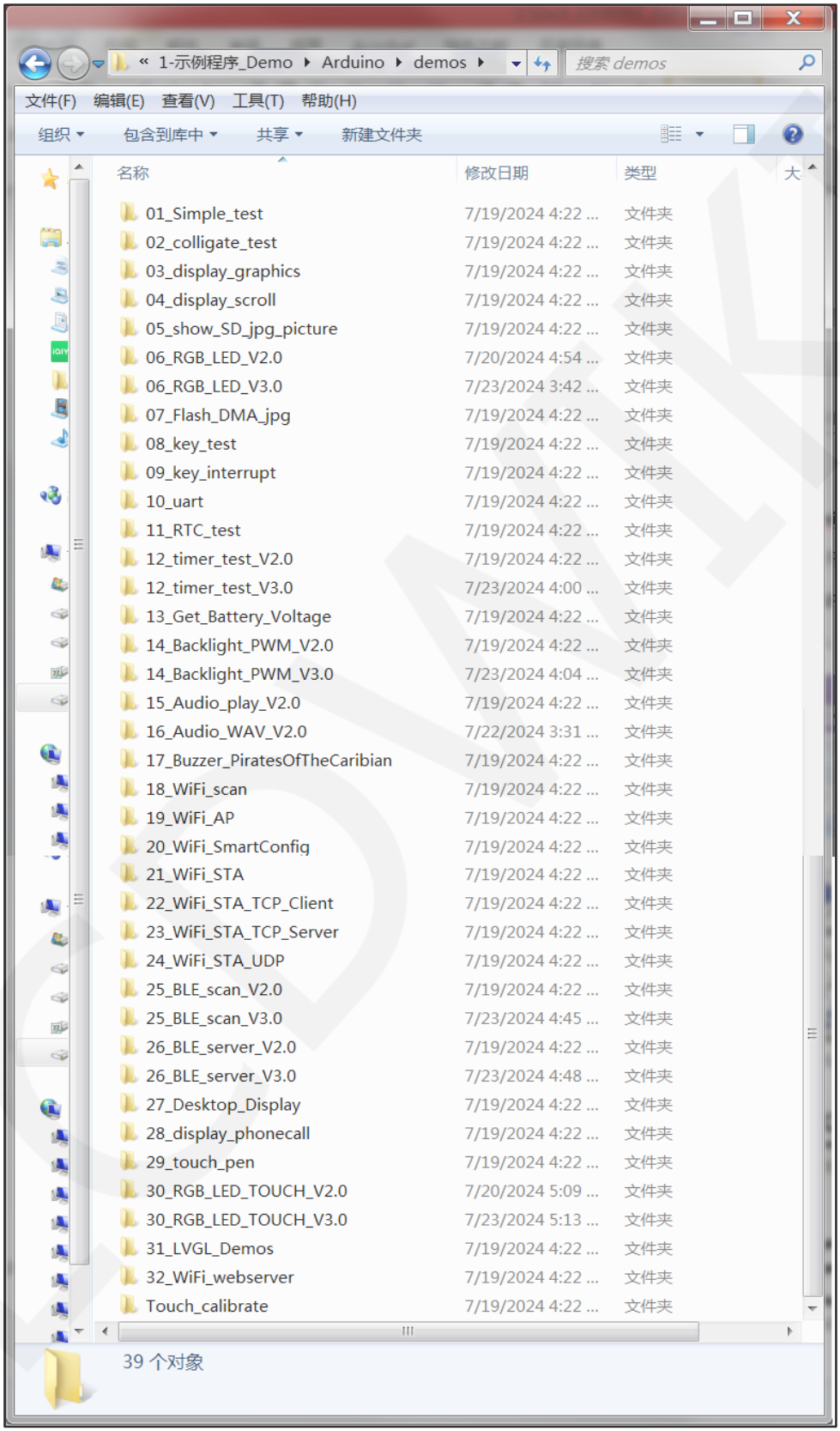

图 3.10 示例程序

各示例程序介绍如下：

01_Simple_test

此示例为基本的示例程序，不依赖任何第三方库。硬件需要用到 LCD 显示屏，显示内容为全屏颜色填充和随机矩形填充。此示例可直接用于检查显示屏是否正常。

02_colligate_test

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏。显示的内为画点、画线、各种图形显示，还有运行时间统计，是一个比较综合的显示示例。

03_display_graphics

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏。显示内容为各种图形绘画和填充。

04_display_scroll

此示例需要依赖 TFT_eSPI 软件库，硬件需用到 LCD 显示屏。显示内容为汉字和图片显示，文字滚动显示，反色显示，四个方向旋转显示。

05_show_SD_jpg_picture

此示例需要依赖 TFT_eSPI 和 TJpg_Decoder 软件库，硬件需要用到 LCD 显示屏和 MicroSD 卡。此示例功能为读取 MicroSD 卡里 JPG 图片并解析，然后在 LCD 上显示图片。示例使用步骤为：

A、通过电脑将示例文件夹里“**PIC_320x480**”目录下的 JPG 图片拷贝到 MicroSD 卡根目录里。

B、将 MicroSD 卡插入到显示模块的 SD 卡槽中；

C、给显示模块上电，编译并下载该示例程序，可以看到 LCD 屏上有图片轮流显示。

06_RGB_LED_V2.0

此示例不依赖任何第三方软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 RGB 三色灯。此示例展示了 RGB 三色灯亮灭控制、闪烁控制以及 PWM 亮度控制。

06_RGB_LED_V3.0

此示例不依赖任何第三方软件库，只能使用 Arduino-ESP32 的 3.0 版本核心软件库（例如 3.0.3 版本）。需要用到的硬件以及功能和 06_RGB_LED_V2.0 示例一致。

07_Flash_DMA_jpg

此示例需要依赖 TFT_eSPI 和 TJpg_Decoder 软件库。硬件需要用到 LCD 显示屏。此示例展示了从 ESP32 模组内部 Flash 读取 JPG 图片取模数据并解析，然后在 LCD 上显示图片。示例使用步骤：

A、通过在线取模工具对需要显示的 jpg 图片取模。

在线取模工具网址：

http://tomeko.net/online_tools/file_to_hex.php?lang=en

B、取模成功后，将取模数据拷贝到示例文件夹里“**image.h**”文件的数组里（可对数组进行重命名，且示例程序里也要同步修改）

C、给显示模块上电，编译并下载该示例程序，可看到 LCD 屏上有图片显示。

08_key_test

此示例不依赖任何第三方软件库。硬件需要用到 BOOT 按键和 RGB 三色灯。此示例展示通过轮询方式来检测按键事件，同时操作按键来控制 RGB 三色灯亮灭。

09_key_interrupt

此示例不依赖任何第三方软件库。硬件需要用到 BOOT 按键和 RGB 三色灯。此示例展示通过中断方式来检测按键事件，同时操作按键来控制 RGB 三色灯亮灭。

10_uart

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到串口和 LCD 显示屏。此示例展示了 ESP32 通过串口和电脑端进行交互。ESP32 通过串口向电脑端发送信息，电脑端也通过串口向 ESP32 发送信息，ESP32 接收后将信息在 LCD 屏上显示。

11_RTC_test

此示例需要依赖 TFT_eSPI 和 ESP32Time 软件库，硬件需要用到 LCD 显示屏。此示例展示了使用 ESP32 的 RTC 模块设置实时时间和日期，并将时间和日期显示到 LCD 显示屏上。

12_timer_test_V2.0

此示例不依赖任何第三方软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 RGB 三色灯。此示例展示了 ESP32 定时器使用，通过设置 1 秒的定时时间来控制绿色 LED 灯亮灭（间隔 1 秒亮，间隔 1 秒灭，一直循环）。

12_timer_test_V3.0

此示例不依赖任何第三方软件库，只能使用 Arduino-ESP32 的 3.0 版本核心软件库（例如 3.0.3 版本）。硬件需要用到 RGB 三色灯。此示例展示功能和 12_timer_test_V2.0 示例一致。

13_Get_Battery_Voltage

此示例依赖 TFT_eSPI 软件库。硬件需要用到 LCD 显示屏和 3.7V 锂电池。此示例展示了使用 ESP32 的 ADC 功能获取外接锂电池的电压，并将它显示到 LCD 显示屏上。

14_Backlight_PWM_V2.0

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 LCD 显示屏和电阻触摸屏。此示例展示了通过显示模块的触摸滑动操作来调节显示屏的背光亮度，同时显示亮度值变化。

14_Backlight_PWM_V3.0

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库，只能使用 Arduino-ESP32 的 3.0 版本核软件库（例如 3.0.3 版本）。硬件需要用到 LCD 显示屏和电阻触摸屏。此示例展示的功能和 14_Backlight_PWM_V2.0 示例一致。

15_Audio_play_V2.0

此示例需要依赖 TFT_eSPI、TFT_Touch、TJpg_Decoder 和 ESP32-audioI2S 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 LCD 显示屏，电阻触摸屏，喇叭以及 MicroSD 卡。此示例展示了读取 SD 卡里的 mp3 音频文件，将文件名称显示到 LCD 上，并进行循环播放。显示屏上有两个触摸按钮图标，操作可以控制音频的暂停和播放，操作另一个可以控制静音和播放声音。示例使用步骤如下：

A、将示例文件夹里“**mp3**”目录下的 mp3 音频文件全部拷贝到 MicroSD 卡里。当然也可以不使用该目录下的音频文件，另外找一些 mp3 音频文件，需要注意的地方就是该示例程序最多只能循环播放 10 首 mp3 歌曲。

B、将 MicroSD 卡插入到显示模块的 SD 卡槽中；

C、给显示模块上电，编译并下载该示例程序，可以看到 LCD 屏上有歌曲名

称显示，外接喇叭有声音播放。操作屏上触摸按钮图标可以控制音频播放。

16_Audio_WAV_V2.0

此示例需要依赖 XT_DAC_Audio 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到喇叭。此示例展示了使用 ESP32 播放 wav 格式的音频文件。此示例使用步骤如下：

A、编辑需要播放的音频文件，将生成的音频数据拷贝到示例文件夹里“**Audio_data.h**”文件的数组里（可对数组进行重命名，且示例程序里也要同步修改）。需要注意的地方是编辑的音频文件不能太大，否则会超过 ESP32 模组内部 Flash 容量。这就意味着需要对音频文件的时长，采样率以及声道数目进行编辑。这里推荐一款音频编辑软件“**Audacity**”，可以自行去网上下载。

B、给显示模块上电，编译并下载该示例程序，可以听到喇叭播放音频。

17_Buzzer_PiratesOfTheCaribian

此示例不依赖任何第三方软件库，硬件需要用到喇叭。此示例展示了使用不同的频率拉高拉低引脚来模拟声波震动，从而引起喇叭发声。

18_WiFi_scan

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏和 ESP32 WIFI 模块。此示例展示了 ESP32 WIFI 模块在 STA 模式下扫描周围无线网络信息。并将扫描到的无线网络信息显示到 LCD 显示屏上。无线网络信息包括 SSID、RSSI、CHANNEL、ENC_TYPE 等内容。无线网络信息扫描结束后，会显示扫描个数，最多只会显示前 17 个扫描到的无线网络信息。

19_WiFi_AP

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏和 ESP32 WIFI 模块。此示例展示了 ESP32 WIFI 模块设为 AP 模式，供 WIFI 终端连接。在显示屏上会显示 ESP32 WIFI 模块 AP 模式下设置的 SSID、密码、主机 IP 地址、主机 MAC 地址等信息，一旦有终端连接成功，显示屏会显示终端连接个数。在示例程序开头位置“ssid”和“password”变量里自行设置 SSID 和密码，如下图所示：

```
19
20 //AP mode SSID and PWD
21 const char *ssid = "ESP32_AP";
22 const char *password = "0123456789";
```

图 3.11 设置 AP 模式下 SSID 和密码

20_WiFi_SmartConfig

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、ESP32 WIFI 模块以及 BOOT 按键。此示例展示了 ESP32 WIFI 模块在 STA 模式下，通过 EspTouch 手机 APP 智能配网的过程。整个示例程序运行流程图如下：

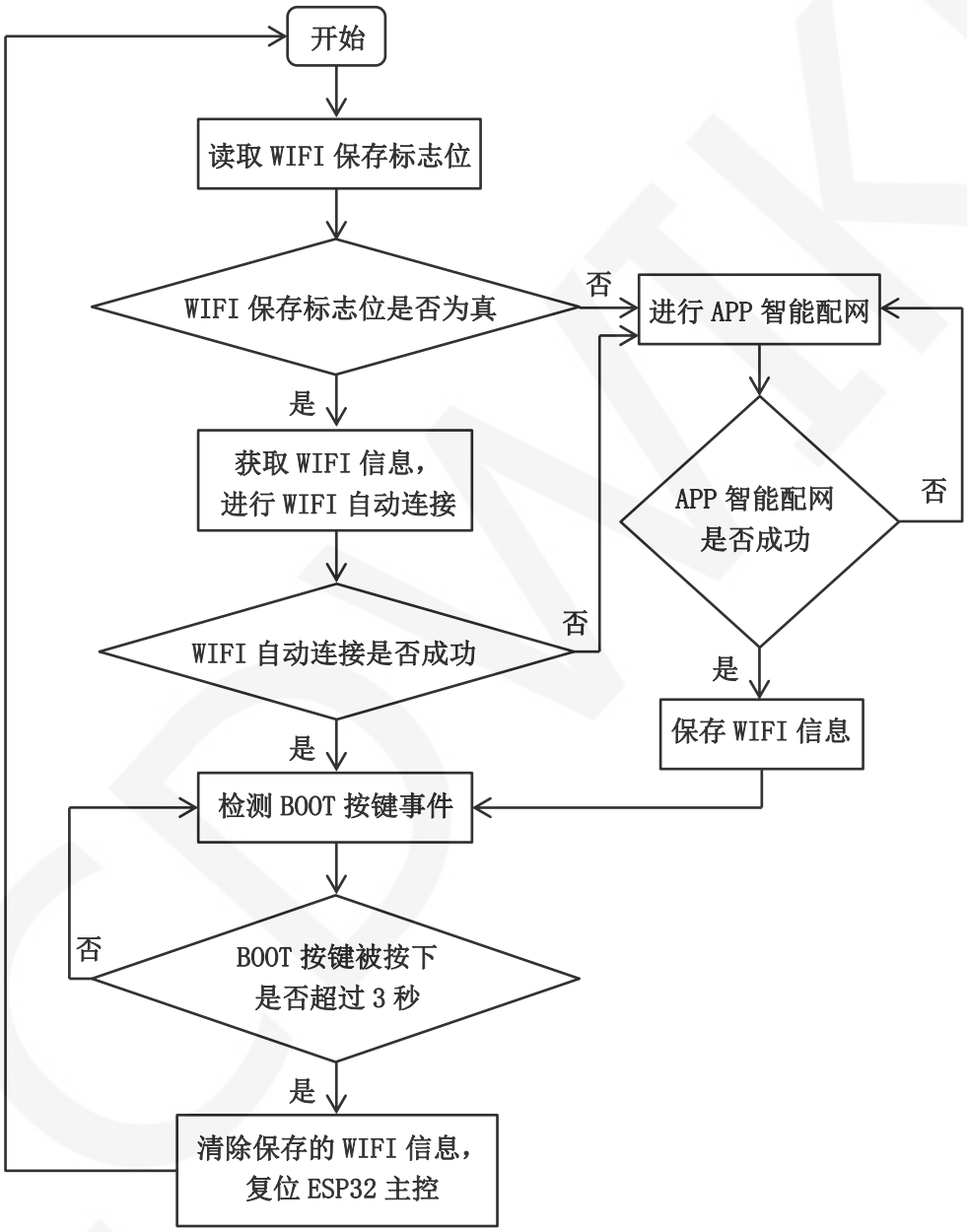

图 3.12 智能配网示例程序运行流程图

此示例程序的操作步骤如下：

A、在手机上下载 EspTouch 应用程序，或者从资料包里“**7-工具软件_Tool_software**”文件夹下拷贝安装程序“**esptouch-v2.0.0.apk**”（只有

安卓系统的安装程序，IOS 系统的应用程序只能从设备上安装），还可从官网下载安装程序。官网下载网址如下：

https://www.espressif.com.cn/en/support/download/apps

B、给显示模块上电，编译并下载该示例程序，如果 ESP32 里没有保存任何 WIFI 信息，则直接进入智能配网模式，此时在手机上打开 EspTouch 应用程序，输入手机所连接 WIFI 的 SSID 和密码，然后以 UDP 广播的方式将相关的信息传播，一旦 ESP32 接收到此信息，就会按照信息里的 SSID 和密码去连接网络，网络连接成功后，会在显示屏上显示 SSID、密码、IP 地址以及 MAC 地址等信息同时保存 WIFI 信息。需要注意的地方就是此配网方式成功率不太高，如果失败了，需要多尝试几次。

C、如果 ESP32 保存有 WIFI 信息，开机则会按照保存的 WiFi 信息去自动连网。如果连网失败，则会进入智能配网模式。网络连接成功后，长按 BOOT 超过 3 秒，则会清除保存的 WIFI 信息，复位 ESP32 重新进行智能配网。

21_WiFi_STA

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、ESP32 WIFI 模块。此示例程序展示了 ESP32 在 STA 模式下，根据提供的 SSID 和密码连接 WIFI 的过程。此示例程序操作步骤如下：

A、在示例程序开头位置“ssid”和“password”变量里写入需要连接的 WIFI 信息，如下如图所示：

```
17 #include <TFT_eSPI.h>
18 #include <WiFi.h>
19
20 //Manually modifying parameters
21 const char *ssid = "yourssid";
22 const char *password = "yourpwd";
23
```

图 3.13 写入 WIFI 信息

B、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到 ESP32 开始连接 WIFI。如果 WIFI 连接成功，则在显示屏上显示成功提示、SSID、IP 地址、MAC 地址等信息；如果连接时间超过 3 分钟，则连接失败，显示失败提示。

22_WiFi_STA_TCP_Client

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、ESP32 WIFI

模块。此示例程序展示了 ESP32 在 STA 模式下，连接 WIFI 后，作为 TCP 客户端连接 TCP 服务端的过程。此示例程序操作步骤如下：

A、在示例程序开头位置“ssid”、“password”、“serverIP”、“serverPort”变量里写入需要连接的 WIFI 信息，TCP 服务端的 IP 地址（电脑的 IP 地址）及端口号，如下如图所示：

```
//Manually modifying parameters
const char *ssid = "yourssid";
const char *password = "yourpwd";

const IPAddress serverIP(192,168,4,52); //The server address to b
uint16_t serverPort = 8080;           //Server port number

char t_buf[100] = {0};
```

图 3.14 写入 WIFI 信息和 TCP 服务端信息 1

B、在电脑上打开“**TCP&UDP 测试工具**”或者“**网络调试助手**”等测试工具（安装包在资料包“**7-工具软件_Tool_software**”目录），在工具里创建一个 TCP 服务 器，端口号要和示例程序里设置一致。

C、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到 ESP32 开始连接 WIFI。如果 WIFI 连接成功，则在显示屏上显示成功提示、SSID、IP 地址、MAC 地址、TCP 服务器端口号等信息，然后开始连接 TCP 服务器，连接成功后，会有相应的提示。此时可以和服务器端进行通信。

23_WiFi_STA_TCP_Server

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、ESP32 WIFI 模块。此示例程序展示了 ESP32 在 STA 模式下，连接 WIFI 后，作为 TCP 服务器端被 TCP 客户端连接的过程。此示例程序操作步骤如下：

A、在示例程序开头位置“ssid”、“password”、“port”变量里写入需要连接的 WIFI 信息，TCP 服务器端端口号，如下如图所示：

```
19
20 //Manually modifying parameters
21 const char *ssid = "yourssid";
22 const char *password = "yourpwd";
23
24 char t_buf[100] = {0};
25 int port = 10000;
26
27 WiFiServer server(port); //Declare server objects
28
```

图 3.15 写入 WIFI 信息和 TCP 服务端信息 2

B、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到 ESP32 开始连接 WIFI。如果 WIFI 连接成功，则在显示屏上显示成功提示、SSID、IP 地址、MAC 地址、TCP 服务器端口号等信息，然后创建 TCP 服务器，等待 TCP 客户端连接。

C、在电脑上打开“**TCP&UDP 测试工具**”或者“**网络调试助手**”等测试工具（安装包在资料包“**7-工具软件_Tool_software**”目录），在工具里创建一个 TCP 客户端（注意 IP 地址以及端口号要和显示屏上显示的内容一致），然后开始连接服务器，如果连接成功则会有相应的提示，此时可和服务器端进行通信。

24_WiFi_STA_UDP

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、ESP32 WIFI 模块。此示例程序展示了 ESP32 在 STA 模式下，连接 WIFI 后，作为 UDP 服务器端被 UDP 客户端连接的过程。此示例程序操作步骤如下：

A、在示例程序开头位置“ssid”、“password”、“localUdpPort”变量里写入需要连接的 WIFI 信息，UDP 服务器端端口号，如下如图所示：

```
//Manually modifying parameters
const char *ssid = "yourssid";
const char *password = "yourpwd";

char t_buf[100] = {0};

AsyncUDP udp;                           //Creating UDP Objects
unsigned int localUdpPort = 10000;  //Local port number
```

图 3.16 写入 WIFI 信息和 UDP 服务端信息

B、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到 ESP32 开始连接 WIFI。如果 WIFI 连接成功，则在显示屏上显示成功提示、SSID、IP 地址、MAC 地址、本地端口号等信息，然后创建 UDP 服务器，等待 UDP 客户端连接。

C、在电脑上打开“**TCP&UDP 测试工具**”或者“**网络调试助手**”等测试工具（安装包在资料包“**7-工具软件_Tool_software**”目录），在工具里创建一个 UDP 客户端（注意 IP 地址以及端口号要和显示屏上显示的内容一致），然后开始连接服务器，如果连接成功则会有相应的提示，此时可和服务器端

进行通信。

25_BLE_scan_V2.0

此示例需要依赖 TFT_eSPI 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 LCD 显示屏、ESP32 蓝牙模块。此示例展示了 ESP32 蓝牙模块扫码周围 BLE 蓝牙设备，并将扫描到的有名称的 BLE 蓝牙设备的名称和 RSSI 显示到 LCD 显示屏上。

25_BLE_scan_V3.0

此示例需要依赖 TFT_eSPI 软件库，只能使用 Arduino-ESP32 的 3.0 版本核软件库（例如 3.0.3 版本）。硬件需要用到 LCD 显示屏、ESP32 蓝牙模块。此示例程序的功能和 25_BLE_scan_V2.0 示例程序一致。

26_BLE_server_V2.0

此示例需要依赖 TFT_eSPI 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 LCD 显示屏、ESP32 蓝牙模块。此示例展示了 ESP32 蓝牙模块创建蓝牙 BLE 服务器端，被蓝牙 BLE 客户端连接，并进行通信的过程。此示例使用步骤如下：

A、在手机上安装蓝牙 BLE 调试工具，例如“BLE 调试助手”、“LightBlue”等。

B、给显示模块上电，编译并下载该示例程序，可以在显示屏上看到蓝牙 BLE 客户端运行提示。如果想自行修改蓝牙 BLE 服务器端设备名称，可在示例程序中“**BLEDevice::init**”函数传参中修改，如下图所示：

```
68 void setupBLE()
69 {
70   BLEDevice::init("ESP32_BT_BLE");    //Create BLE device
71   pServer = BLEDevice::createServer();    //Create BLE server
72   pServer->setCallbacks(new MyServerCallbacks());    //Set the
```

图 3.17 设置蓝牙 BLE 服务端设备名称

C、打开手机端蓝牙和蓝牙 BLE 调试工具，搜索蓝牙 BLE 服务端设备名称（默认为“**ESP32_BT_BLE**”），然后点击该名称进行连接，连接成功后，ESP32 显示模块会有提示。接下来就可以进行蓝牙通信了。

26_BLE_server_V3.0

此示例需要依赖 TFT_eSPI 软件库，只能使用 Arduino-ESP32 的 3.0 版本核软件库（例如 3.0.3 版本）。硬件需要用到 LCD 显示屏、ESP32 蓝牙模块。此示例功能和 26_BLE_server_V2.0 示例一致。

27_Desktop_Display

此示例程序需要依赖 ArduinoJson、Time、HttpClient、TFT_eSPI、TJpg_Decoder、NTPClient 软件库。硬件需要用到 LCD 显示屏、ESP32 WIFI 模块。此示例展示了一个天气时钟桌面，可以显示城市天气情况（包括温度、湿度、天气图标以及滚动显示其他天气信息），当前时间及日期，还有一个太空人动画。天气信息通过网络从天气网获取，时间信息从 NTP 服务器更新。此示例程序使用步骤如下：

A、打开示例后，首先得将**工具->Partition Scheme** 设置为 **Huge APP(3MB No OTA /1MB SPIFFS)**选项，否则编译时会报内存不够的错误。

B、在示例程序开头位置“ssid”和“passwd”变量里写入需要连接的 WIFI 信息，如下如图所示。如果不设置，则进行智能配网（关于智能配网的说明，请参照智能配送示例程序）

```
42 //---------------wifi information---------------------
43 const char* ssid = "yourssid";      //WIFI name
44 const char* passwd = "yourpasswd";    //WIFI password
45 //----------------------------------------------------
```

图 3.17 设置 WIFI 信息

C、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到天气时钟桌面。

28_display_phonecall

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库。硬件需要用到 LCD 显示屏和电阻触摸屏。此示例展示了一个简单的手机拨号界面，通过触摸按钮输入内容。

29_touch_pen

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库。硬件需要用到 LCD 显示屏和电阻触摸屏。此示例展示在显示屏上通过触摸画线，可检测触摸屏功能是否正常。

30_RGB_LED_TOUCH_V2.0

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库，只能使用 Arduino-ESP32 的 2.0 版核心软件库（例如 2.0.17 版本）。硬件需要用到 LCD 显示屏、电阻触摸屏和 RGB 三色灯。此示例展示了通过触摸按钮控制 RGB 灯亮灭，闪烁以及亮度调节。

30_RGB_LED_TOUCH_V3.0

此示例需要依赖 TFT_eSPI、TFT_Touch 软件库，只能使用 Arduino-ESP32 的

3.0 版本核软件库（例如 3.0.3 版本）。硬件需要用到 LCD 显示屏、电阻触摸屏和 RGB 三色灯。此示例展示的功能和 30_RGB_LED_TOUCH_V2.0 测试示例一致。

31_LVGL_Demos

此示例需要依赖 TFT_eSPI、TFT_Touch、lvgl 软件库，硬件需要用到 LCD 显示屏、电阻触摸屏。此示例展示了 lvgl 嵌入式 UI 系统 5 个自带的 Demo 功能。通过此示例，可以学习怎么在 ESP32 平台移植 lvgl 以及怎么配置显示屏和触摸屏等底层设备。在示例程序里一次只能编译一个 demo，把需要编译的 demo 注释去掉，其他的 demo 需要加上注释，如下图所示：

```
111     // uncomment one of these demos
112      lv_demo_widgets();
113     // lv_demo_benchmark();
114     // lv_demo_keypad_encoder();
115     // lv_demo_music();
116     // lv_demo_stress();
```

图 3.18 选择 lvgl demo

lv_demo_widgets：各种小控件测试 demo

lv_demo_benchmark：性能基准测试 demo

lv_demo_keypad_encoder：键盘编码器测试 demo

lv_demo_music：音乐播放器测试 demo

lv_demo_stress：压力测试 demo

注意：此示例第一次编译时，时间较长，大概 15 分钟左右。

32_WiFi_webserver

此示例需要依赖 TFT_eSPI 软件库，硬件需要用到 LCD 显示屏、RGB 三色灯。此示例展示了建立一个 web 服务器，然后电脑上访问该 web 服务器，在 web 界面上操作图标控制 RGB 三色灯。此示例使用步骤如下：

A、在示例程序开头位置“ssid”和“password”变量里写入需要连接的 WIFI 信息，如下如图所示：

```
28 //Manually modifying parameters
29 const char *ssid = "yourssid";
30 const char *password = "yourpwd";
```

图 3.19 设置 WIFI 信息

B、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到 ESP32

开始连接 WIFI。如果 WIFI 连接成功，则在显示屏上显示成功提示、SSID、IP 地址、MAC 地址等信息。

C、在电脑上的浏览器网址输入栏输入上述步骤显示的 IP 地址，此时可以访问 web 界面，点击界面上相应的图标就可以控制 RGB 三色灯。

Touch_calibrate

此程序需要依赖 TFT_eSPI、TFT_Touch 软件库，其专门用于电阻触摸屏校准，校准步骤如下：

A、打开校准程序，设置触摸屏 SPI 总线引脚定义，如下图所示。一定要保持定义的引脚和实际用到引脚一致。

```
#define RTP_DOUT 39
#define RTP_DIN  32
#define RTP_SCK  25
#define RTP_CS   33
```

图 3.20 设置触摸屏 SPI 总线引脚

B、给显示模块上电，编译并下载该示例程序，在显示屏上可以看到校准界面，此时根据提示依次点击四个红色十字。

C、校准完成后，校准结果通过串口输出，如下图所示，同时进入校准检测界面，在此界面通过画点画线检测校准结果是否准确，如不准确则需重新校准。

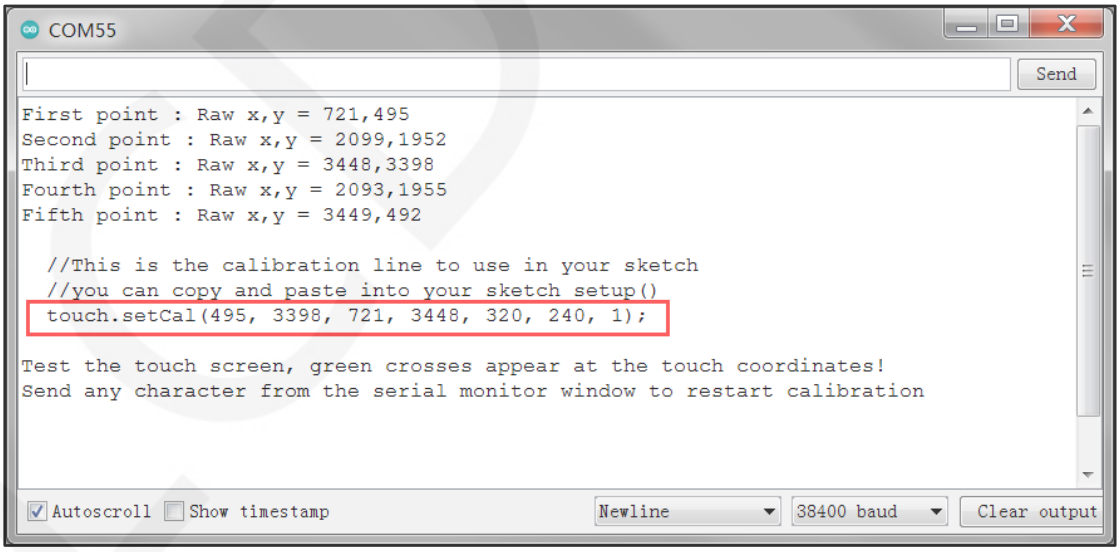

图 3.21 串口输出电阻触摸屏校准参数

D、校准结果准确后，将串口端校准参数拷贝到使用的示例程序里。