

# 5.0 寸 RGB Arduino 示例程序说明

# 目 录

1. 软件和硬件平台说明.....	3
2. 引脚分配说明.....	3
3. 示例程序使用说明.....	5
3.1. 搭建 ESP32 Arduino 开发环境.....	5
3.2. 安装第三方软件库.....	5
3.3. 示例程序使用说明.....	11

## 1. 软件和硬件平台说明

**模块：**5.0寸ESP32-S3 RGB显示模块，拥有800x480分辨率。

**模块主控：**ESP32-S3芯片，最高主频240MHz，支持2.4G WIFI+蓝牙。

**Arduino IED版本：**2.3.4版本。

**ESP32 Arduino核心库软件版本：**3.2.0版本。

## 2. 引脚分配说明

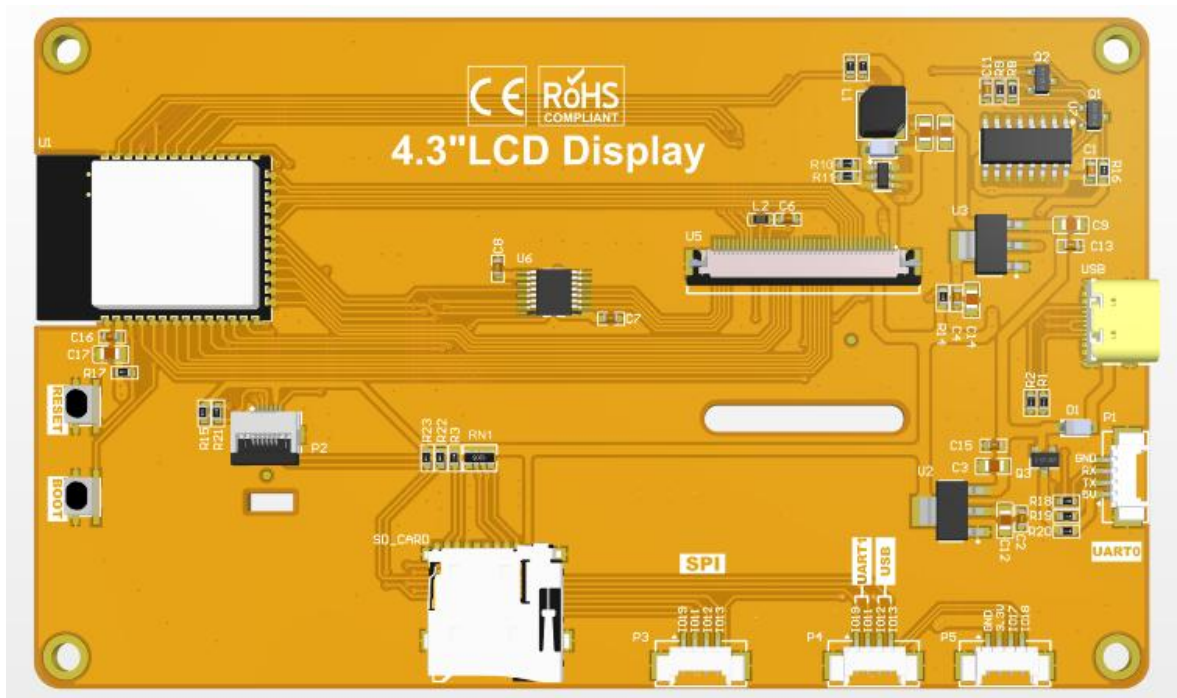


图2.1 5.0寸ESP32-S3 显示模块背面图

5.0寸ESP32-S3显示模块主控为ESP32-S3，其连接板载外设的GPIO分配如下表格所示：

ESP32-S3芯片引脚分配说明			
板载设备	板载设备引脚	ESP32-S3 连接引脚	说明
LCD	LCD_DE	IO40	数据使能控制引脚
	LCD_VSYNC	IO41	垂直同步信号控制引脚
	LCD_HSYNC	IO39	水平同步信号控制引脚
	LCD_PCLK	IO42	像素时钟控制引脚
	PIN_R0	IO45	5位RED数据引脚
	PIN_R1	IO48	

	PIN_R2	IO47	
	PIN_R3	IO21	
	PIN_R4	IO14	
	PIN_G0	IO5	6位GREEN数据引脚
	PIN_G1	IO6	
	PIN_G2	IO7	
	PIN_G3	IO15	
	PIN_G4	IO16	
	PIN_G5	IO4	
	PIN_B0	IO8	5位BLUE数据引脚
	PIN_B1	IO3	
	PIN_B2	IO46	
	PIN_B3	IO9	
	PIN_B4	IO1	
	<b>RTP</b>	TP_SCK	IO12
TP_MISO		IO13	电阻屏SPI总线数据写入控制引脚
TP_MOSI		IO11	电阻屏SPI总线数据读取控制引脚
TP_CS		IO38	电阻屏SPI总线片选控制引脚
TP_INT		IO18	电阻屏SPI总线中断控制引脚
<b>SD</b>	SD_CS	IO10	SD卡SPI总线片选控制引脚
	SD_MOSI	IO11	SD卡SPI总线数据读取控制引脚
	SD_MISO	IO13	SD卡SPI总线数据写入控制引脚
	SD_SCLK	IO12	SD卡SPI总线时钟控制引脚
<b>KEY</b>	KEY_BOOT	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）
	KEY_RESET	EN	ESP32-S3复位按键，低电平复位
<b>UART0</b>	TX0	IO43	ESP32-S3串口0发送信号引脚
	RX0	IO44	ESP32-S3串口0接收信号引脚

表2.1 ESP32-S3板载外设引脚分配说明

### 3. 示例程序使用说明

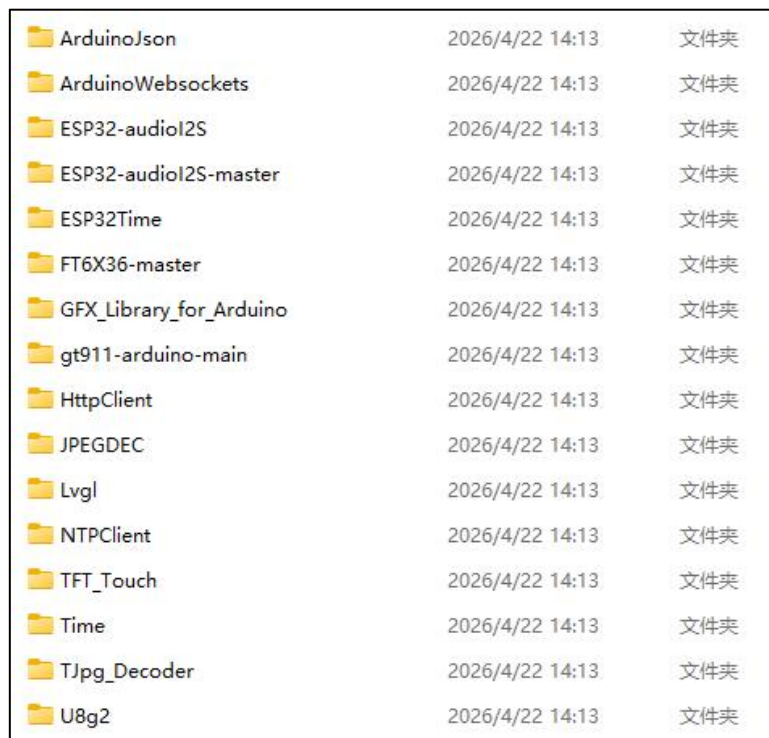
#### 3.1. 搭建ESP32 Arduino开发环境

ESP32 Arduino开发环境搭建详细说明见资料包里的“ESP32\_Arduino\_IDE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：



ArduinoJson	2026/4/22 14:13	文件夹
ArduinoWebsockets	2026/4/22 14:13	文件夹
ESP32-audioI2S	2026/4/22 14:13	文件夹
ESP32-audioI2S-master	2026/4/22 14:13	文件夹
ESP32Time	2026/4/22 14:13	文件夹
FT6X36-master	2026/4/22 14:13	文件夹
GFX_Library_for_Arduino	2026/4/22 14:13	文件夹
gt911-arduino-main	2026/4/22 14:13	文件夹
HttpClient	2026/4/22 14:13	文件夹
JPEGDEC	2026/4/22 14:13	文件夹
Lvgl	2026/4/22 14:13	文件夹
NTPClient	2026/4/22 14:13	文件夹
TFT_Touch	2026/4/22 14:13	文件夹
Time	2026/4/22 14:13	文件夹
TJpg_Decoder	2026/4/22 14:13	文件夹
U8g2	2026/4/22 14:13	文件夹

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

**ArduinoWebsockets:** 用 Arduino 编写现代 websockets 应用程序的库

**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。

**GFX\_Library\_for\_Arduino:** 液晶屏的Arduino图形库,支持多种平台和多种LCD驱动IC。

**Time:** 为Arduino提供计时功能的软件库。

**JPEGDEC:** 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。

**U8g2:** Arduino平台字库，可将中文汉字显示到LCD上。

**gt911\_arduino\_main:** 电容触摸驱动库，可以实现触摸功能

- 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、上述资料包里的第三方软件库已经配置好了，可以直接使用。安装完成后，打开示例程序编译运行就可以看到效果了。

### 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 \Arduinodemos”目录下，如下图所示：

1.HelloWorld	2026/1/22 16:53	文件夹
2.PDQgraphicstest	2026/1/22 16:53	文件夹
3.ImgViewerJpeg	2026/1/22 16:53	文件夹
4.RTC_test	2026/1/22 16:53	文件夹
5.UART	2026/1/22 16:53	文件夹
6.PWM_Backlight	2026/1/22 16:53	文件夹
7.BLE_Scan	2026/1/22 16:53	文件夹
8.BLE_Server	2026/1/22 16:53	文件夹
9.WIFI_AP	2026/1/22 16:53	文件夹
10.WIFI_Scan	2026/1/22 16:53	文件夹
11.WIFI_STA	2026/1/22 16:53	文件夹
12.WiFi_STA_TCP_Client	2026/1/22 16:53	文件夹
13.WiFi_STA_TCP_Server	2026/1/23 13:41	文件夹
14.WiFi_STA_Udp	2026/1/23 14:29	文件夹
15.WiFi_webserver	2026/1/23 17:15	文件夹
16.LVGL_Touch	2026/1/22 16:53	文件夹
17.Touch_Pen	2026/1/22 16:53	文件夹
18.Calculator	2026/1/22 16:53	文件夹
19.DesktopDisplay	2026/1/22 16:53	文件夹

图 3.2 示例程序

打开例程后在工具->开发板中选择 ESP32S3 开发板



图 3.3

然后选择接入的端口

其余设置如下

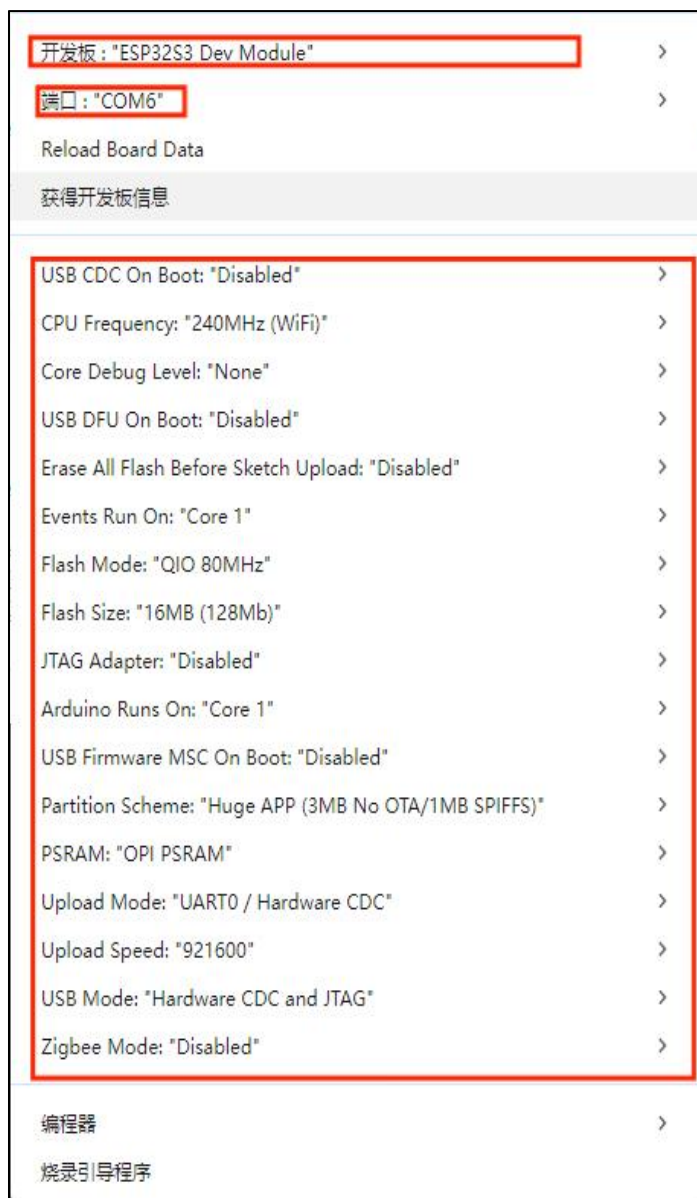


图 3.4

各示例程序介绍如下：

### 01\_HelloWorld

此示例为基本的示例程序，需要依赖 `GFX_Library_for_Arduino` 库。硬件需要用到 LCD 显示屏，显示内容为“HelloWorld”字符随机填充。此示例用于检查显示屏是否正常。

### 02\_PDQgraphicstest

此示例需要依赖 `GFX_Library_for_Arduino` 库，硬件需要用到 LCD 显示屏。显示的为全屏填充颜色、画点、画线、各种图形显示，是一个比较综合的显示示例。

### 03\_ImgViewerJpeg

此示例需要依赖 `GFX_Library_for_Arduino` 库 和 `JPEGDEC` 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“**DATA**”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。

### 04\_RTC\_test

此示例需要依赖 `GFX_Library_for_Arduino` 库和 `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 模块设置时间和日期，并将时间和日期显示到 LCD 显示屏上

### 05\_Uart

此示例需要依赖 `GFX_Library_for_Arduino` 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。

### 06\_PWM\_Backlight

此示例需要依赖 `GFX_Library_for_Arduino` 库和 `gt911_arduino_main` 库，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。

## 07\_BLE\_Scan

此示例需要依赖 `GFX_Library_for_Arduino` 库，硬件需要用到 LCD 显示屏。此示例展示了 ESP32 蓝牙模块扫描周围 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。

## 08\_BLE\_server

此示例需要依赖 `GFX_Library_for_Arduino` 库和 `U8g2` 字库，硬件需要用到 LCD 显示屏。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：

```
// ===== BLE初始化函数 =====  
void setupBLE()  
{  
    // 初始化BLE设备  
    BLEDevice::init("ESP32_BT_BLE");  
    // 创建BLE服务器  
    pServer = BLEDevice::createServer();  
    pServer->setCallbacks(new MyServerCallbacks());  
    // 创建BLE服务
```

图 3.6 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。

## 09\_WiFi\_scan

此示例需要依赖 `GFX_Library_for_Arduino` 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。

## 10\_WiFi\_AP

此示例需要依赖 `GFX_Library_for_Arduino` 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：

```
// ===== WiFi AP配置 =====
#define AP_SSID "ESP32-S3_AP" // AP热点名称
#define AP_PASSWORD "12345678" // AP密码（至少8位，否则无法启动AP）
#define AP_CHANNEL 6 // AP信道（1~13）
#define AP_MAX_CONNECT 4 // 最大连接数（ESP32最多支持4个）
```

图 3.7 设置 AP 模式下 SSID 和密码

## 11\_WiFi\_STA

此示例需要依赖 `GFX_Library_for_Arduino` 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：

```
// ===== WiFi STA配置 =====
#define WIFI_SSID "ESP" // 要连接的WiFi名称
#define WIFI_PASSWORD "12345678" // 要连接的WiFi密码
#define WIFI_CONNECT_RETRY 5 // 连接失败重试次数
#define WIFI_CONNECT_DELAY 3000 // 重试间隔（毫秒）
```

图 3.8 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；如果连接失败超过 5 次，则连接失败，显示失败提示。

## 12\_WiFi\_STA\_TCP\_Client

此示例需要依赖 `GFX_Library_for_Arduino` 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：

```
// ===== WiFi STA配置 =====
#define WIFI_SSID      "ESP"          // 参考代码的WiFi名称
#define WIFI_PASSWORD  "12345678"    // 参考代码的WiFi密码
#define WIFI_CONNECT_RETRY 5         // WiFi连接失败重试次数
#define WIFI_CONNECT_DELAY 3000      // WiFi重试间隔（毫秒）

// ===== TCP Client配置 =====
#define TCP_SERVER_IP  "192.168.1.126" // 参考代码的TCP服务器IP
#define TCP_SERVER_PORT 8080          // 参考代码的TCP服务器端口
#define TCP_CONNECT_RETRY 3          // TCP连接失败重试次数
#define TCP_CONNECT_DELAY 2000       // TCP重试间隔（毫秒）
#define TCP_RECV_BUFFER_SIZE 1024     // 接收缓冲区大小
```

图 3.9 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。

### 13\_WiFi\_STA\_TCP\_Server

此示例需要依赖 GFX\_Library\_for\_Arduino 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：

```
// ===== WiFi STA配置 =====
#define WIFI_SSID      "yourssid"    // WiFi名称
#define WIFI_PASSWORD  "yourpass"    // WiFi密码
#define WIFI_CONNECT_RETRY 5         // WiFi连接失败重试次数
#define WIFI_CONNECT_DELAY 3000      // WiFi重试间隔（毫秒）

// ===== TCP Server配置 =====
#define TCP_SERVER_PORT 10000         // TCP服务器监听端口
#define TCP_CLIENT_TIMEOUT 5000      // 客户端超时时间（毫秒）
#define TCP_RECV_BUFFER_SIZE 1024    // 接收缓冲区大小
#define MAX_CLIENTS 1                // 最大支持客户端数
```

图 3.11 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，此时可和服务端进行通信。

#### 14\_WiFi\_STA\_UDP

此示例需要依赖 GFX\_Library\_for\_Arduino 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：

```
// ===== WiFi配置 =====  
const char *ssid = "yourssid";  
const char *password = "yourpass";  
  
// ===== UDP配置 =====  
WiFiUDP udp; // 同步UDP对象  
unsigned int localUdpPort = 10000; // 本地监听端口  
unsigned int broadcastPort = localUdpPort;  
const char *broadcastData = "broadcast UDP data";  
char udpRecvBuffer[256] = {0}; // 接收缓冲区（避免内存溢出）
```

图 3.12 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，此时可和服务端进行通信。

## 15\_WIFI\_Webserver

此示例需要依赖 `GFX_Library_for_Arduino` 库, 硬件需要用到 LCD 显示屏、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, 连接该热点后输入以下网址“[192.168.4.1](http://192.168.4.1)”可进入网页, 可实现在网页上控制 LCD 亮度与背光颜色。示例使用步骤如下:

A: 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:

```
// ===== WiFi AP 配置 =====  
const char *ap_ssid = "ESP";           // 设置热点的 SSID  
const char *ap_password = "12345678"; // 设置热点的密码  
WebServer server(80);
```

B: 烧录代码后使用手机连接该热点, 连接成功后进入网址“[192.168.4.1](http://192.168.4.1)”, 可以在网页控制 LCD 亮度与背光颜色

## 16\_LVGL\_Touch

此示例需要依赖 `lvgl` 软件库、`GFX_Library_for_Arduino` 库和 `gt911_arduino_main` 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 4 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:

```
lv_demo_widgets();  
//lv_demo_benchmark();  
//lv_demo_music();  
//lv_demo_stress();
```

图 3.5 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_music`: 音乐播放器测试 demo

`lv_demo_stress`: 压力测试 demo

## 17\_Touch\_pen

此示例需要依赖 `GFX_Library_for_Arduino` 库和 `gt911_arduino_main` 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。

## 18\_Calculator

此示例需要依赖 GFX\_Library\_for\_Arduino 库、XT\_DAC\_Audio 库和 U8g2 软件库，硬件需要用到 LCD 显示屏、电阻触摸屏。此示例展示了一个简易的触摸计算器，可以进行简单的加减乘除运算。

## 19\_Desktop\_Display

此示例程序需要依赖 ArduinoJson、Time、HttpClient、GFX\_Library\_for\_Arduino 库、U8g2 软件库、NTPClient 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、以及显示其他天气信息），当前时间及日期，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。

```
// ===== 网络配置（请修改为你的WiFi名称和密码）
// =====
const char* WIFI_SSID = "ESP"; // WiFi名称
const char* WIFI_PASSWORD = "12345678"; // WiFi密码
const uint32_t CONNECT_TIMEOUT = 10000; // 连接超时时间：10秒（10000毫秒）
const uint32_t PROGRESS_UPDATE_INTERVAL = 100; // 进度条更新间隔：100毫秒
```

图 3.10 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。

**注意：**此示例第一次编译时，时间较长，大概 15 分钟左右。