

E-Paper ESP32 Driver Board

来自Waveshare Wiki
跳转至： 导航、 搜索

支持型号

本Wiki主要介绍该产品的具体操作，如需获取该产品支持的墨水屏型号请前往官网产品详情最底部获取
官网链接：<https://www.waveshare.net/shop/e-Paper-ESP32-Driver-Board.htm>

说明

参数、特点说明

自我介绍

我是电子墨水屏无线网络驱动板，我可以通过WiFi或者蓝牙从PC机或智能手机上获取图片信息，并将信息通过电子墨水屏显示出来。我支持Arduino开发，你可以像使用其它Arduino板一样使用我。

产品参数

- WiFi 标准：802.11b/g/n
- 通信接口：SPI/IIC
- 蓝牙标准：4.2，包含传统蓝牙(BR/EDR)和低功耗蓝牙(BLE)
- 通信接口：3-wire SPI、4-wire SPI(默认)
- 工作电压：5V
- 工作电流：50mA ~ 150mA
- 外形尺寸：29.46mm x 48.25mm

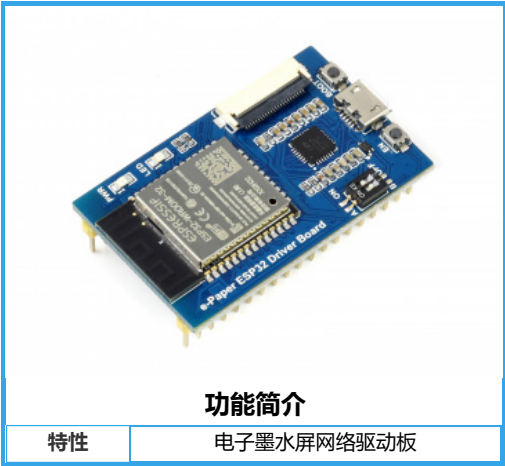
功能引脚

功能引脚	ESP32	描述
VCC	3V3	电源正(3.3V电源输入)
GND	GND	电源地
SCK	P13	SPI的CLK，时钟信号输入
DIN	P14	SPI的MOSI，数据输入
CS	P15	片选，低电平有效
BUSY	P25	忙状态输出引脚（表示忙碌）
RST	P26	复位，低电平有效
DC	P27	数据/命令，低电平表示命令，高电平表示数据

PS：以上为板子固定连接，无需用户额外操作

产品特点

- 板载 ESP32，支持 Arduino 开发
- 提供安卓手机 APP 程序，可通过蓝牙 EDR 更新显示内容，方便使用
- 提供 HTML 上位机程序，可通过网页远程更新显示内容，方便集成到各种网络应用中
- 支持 Floyd-Steinberg 抖动算法，以获得更多的颜色组合，对原始图片进行更好的阴影渲染



- 支持多种常用图片格式(BMP、JPEG、GIF 和 PNG 等)
- 出厂内置电子墨水屏驱动程序(开源)
- 5V管脚支持3.6V到5.5V电压输入，可使用锂电池供电
- 提供完善的配套资料手册

产品应用

本产品配合墨水屏，适用于无线刷图的应用场景。

- 超市电子价签
- 电子名片
- 串口信息显示牌等

使用准备

硬件操作、环境搭建和程序说明等

硬件连接

本产品出货的时候配有一个 ESP32 网络驱动板，一个转接板和 FFC 延长线。
使用的时候你可以直接将屏幕接入到驱动板，或者是通过延长线和转接板接入。

- 将屏幕接入 ESP32 驱动板：

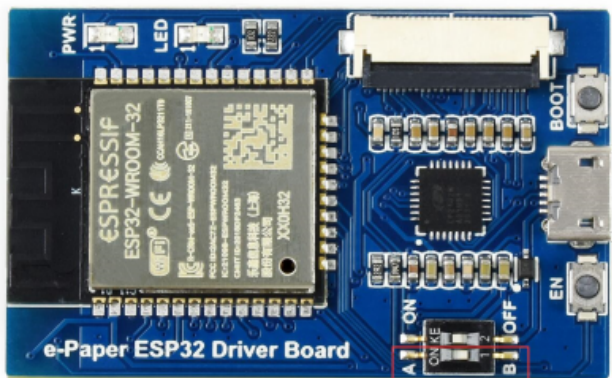
直接接入驱动板：



通过延长线接入：



- 设置型号开关：根据使用的墨水屏型号设置1号开关，屏幕较多，如有未列出请优先使用 ‘A’ 尝试，如果显示效果差或者无法驱动，请尝试切换开关



Trigger state	E-Paper
B	1.54inch e-Paper, 2.13inch e-Paper, 2.13inch e-Paper (D), 2.9inch e-Paper
A	1.54inch e-Paper (B), 1.54inch e-Paper (C), 2.13inch e-Paper (B), 2.13inch e-Paper (C), 2.7inch e-Paper (B), 2.9inch e-Paper (B), 2.9inch e-Paper (C), 4.2inch e-Paper (B), 4.2inch e-Paper (C), 5.83inch e-Paper (B), 5.83inch e-Paper (C), 7.5inch e-Paper (B), 7.5inch e-Paper (C)

- 开启串口模块：将2号开关拨动到“ON”，此开关控制CP2102（USB to UART模块）的供电。不需要使用时，您可以手动关闭使模块更加省电（如果2号开关处于OFF状态，则无法上传程序）
- 使用一条 micro USB 线将 ESP32 驱动板接入到电脑或者 5V 电源

下载例程

我们有提供本地、蓝牙、WiFi三种例程，本页面的“资料”标签内可以找到示例程序，或者点击 [示例程序](#) 下载 将下载下来的压缩包解压出来，可以得到以下文件：

E-Paper_ESP32_Driver_Board_Code			
名称	修改日期	类型	大小
ePape_Esp32_Loader_APP	2020/8/12 18:42	文件夹	
examples	2020/7/22 17:33	文件夹	
Loader_esp32bt	2020/8/10 18:55	文件夹	
Loader_esp32wf	2020/8/29 14:01	文件夹	
app-release.apk	2020/8/12 18:42	APK 文件	1,433 KB

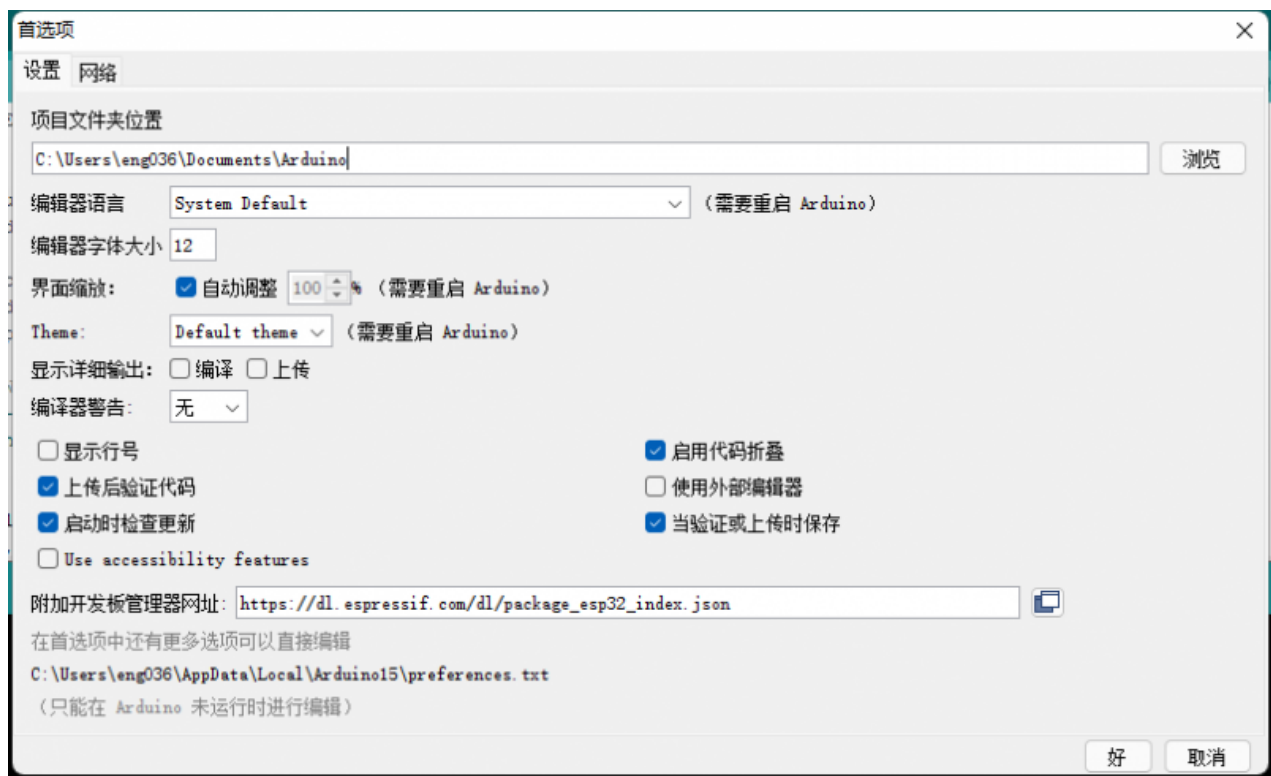
- ePape_Esp32_Loader_APP：蓝牙App源码（Android Studio）
- examples：本地例程
- Loader_esp32bt：蓝牙例程
- Loader_esp32wf：WiFi例程
- app-release.apk：蓝牙例程App安装包

环境配置

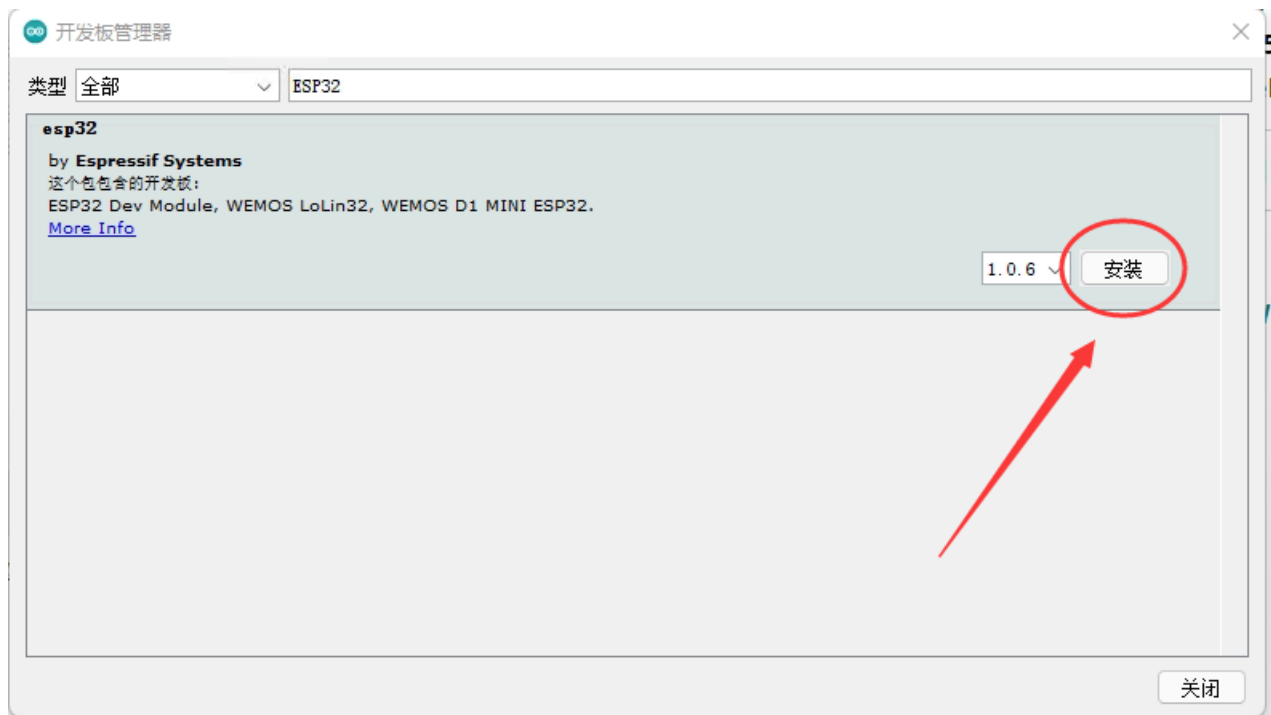
- 如果您电脑之前没有安装 Arduino IDE，或者 IDE 的版本比较老。建议到 [Arduino 官方网站](https://www.arduino.cc/en/Main/Software)根据自己的系统下载最新的 IDE 并安装。
官方链接：<https://www.arduino.cc/en/Main/Software>

下载 ESP32 Dev Module 开发板管理工具：

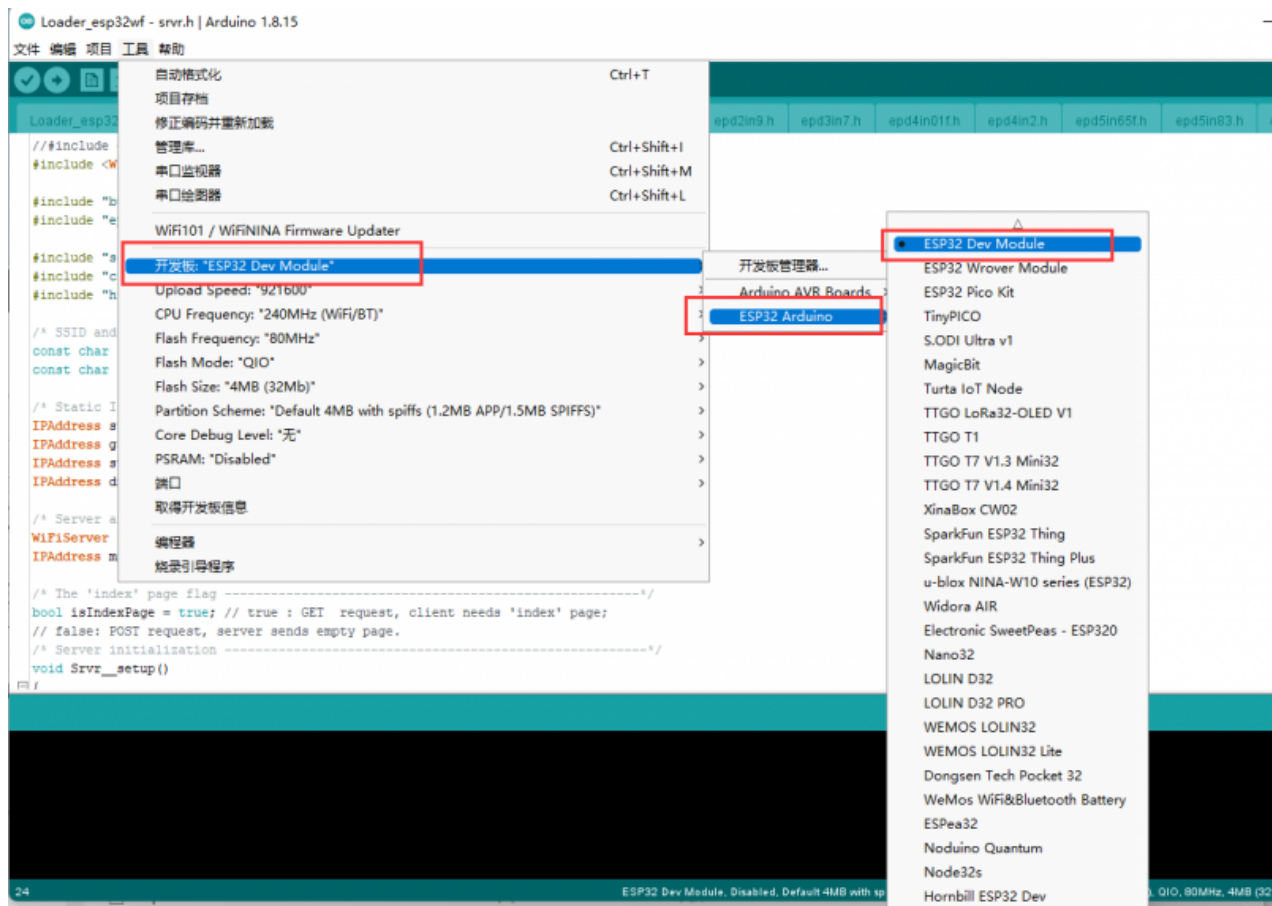
- 1.打开arduino软件的首选项，在附加开发板管理器网址中加入链接：
https://dl.espressif.com/dl/package_esp32_index.json



- 2.添加完成后打开工具-> 开发板 -> 开发板管理工具，手动搜索“ESP32”下载安装最新版本。



- 3.等待安装完成后，可以在 IDE 的 Tools - Boards 里面找到 ESP32 Dev Module 的型号选项即可。



- 若多次出现报错无法完成安装，请联系售后技术人员。

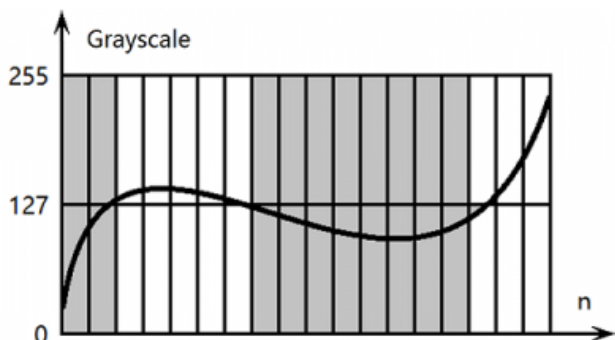
图像处理算法

在蓝牙和WiFi例程中，提供了两种图像处理算法，分别是 Level（色阶法）以及 Dithering（抖动法）

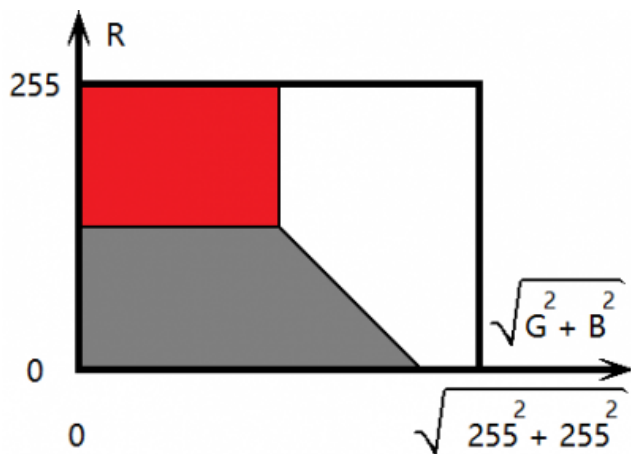
色阶法

一张图像，我们可以把它划分为几个大的颜色域，图像上的每个像素点根据颜色跟这几个色域的趋近程度，被划分到这些颜色域中去。这种方法比较适用于颜色不多的图像，例如亮色或者三色的形状或者文字图像。以黑白红三色墨水屏为例，处理图像的时候我们希望把它处理成黑白红三色，因此对于一张图像来说，我们可以把图像的所有颜色划分三个大的颜色区域：黑色区域，白色区域，红色区域。

比如根据下图，如果灰度图中的某个像素点的值等于或者小于127的话，我们把这个像素点视为黑色像素，否则，就是白色



对于彩色图来说，我们都知道RGB有三色通道，相对于红色通道来说，我们可以把蓝色和绿色统称为蓝-绿通道，或者是非红通道。根据下面的图，彩色图像上的某个像素点，如果它红色通道的值很高，但是蓝-绿通道的值很低的话，我们将它归为红色像素；如果说它红色通道和蓝-绿通道的值都很低的话，我们将它归为黑色像素；红色和蓝-绿通道值都很高的话我们把它归为白色。



算法中，对于颜色定义是根据RGB值以及预期颜色值的平方和的差值计算的。其中预期颜色值是指的像素点最趋近的那个颜色值，这些值被保存在 curPal 数组中。

```
// Returns the discrepancy between given (r, g, b)
// and available colors
function getErr(r, g, b, avlCol)
{
    r -= avlCol[0];
    g -= avlCol[1];
    b -= avlCol[2];
    return r*r + g*g + b*b;
}

// Returns the index of available color
// which has minimal discrepancy with the given one
function getNear(r,g,b)
{
    var ind=0;
    var err=getErr(r,g,b,curPal[0]);

    for (var i=1;i<curPal.length;i++)
    {
        var cur=getErr(r,g,b,curPal[i]);
        if (cur<err){err=cur;ind=i;}
    }

    return ind;
}
```

抖动法

对于那些颜色比较多，或者渐变区域比较多的图像，上面的色阶法并不太合适，很多时候图像里面的渐变区域的像素可能跟所有颜色域都很接近，如果用色阶法的画就会让图像丢失很多图像细节。很多摄像头拍摄的图片，通过混合颜色的方法来绘画阴影和过度区域，这些图像中，渐变区域占了大部分

对于人眼来说，很容易把特别小的颜色混淆了，比如两种颜色红和蓝并列，如果把它缩小到足够小的手，在人眼看来会变成一种由红和蓝混合而成的颜色。人眼的缺陷意味着我们可以通过欺骗人眼，利用“混合”的方法来获取更多可以表现的颜色，抖动算法就是采用了这一种现象。

我们提供的例程中使用了Floyd-Steinberg 抖动算法-基于错误扩散（由Robert Floyd 和Louis Steinberg在1976年发表）。公式是根据下面的图像的方式进行错误扩散

$$\begin{array}{ccc} & X & 7 \\ 3 & 5 & 1 \\ (1/16) \end{array}$$

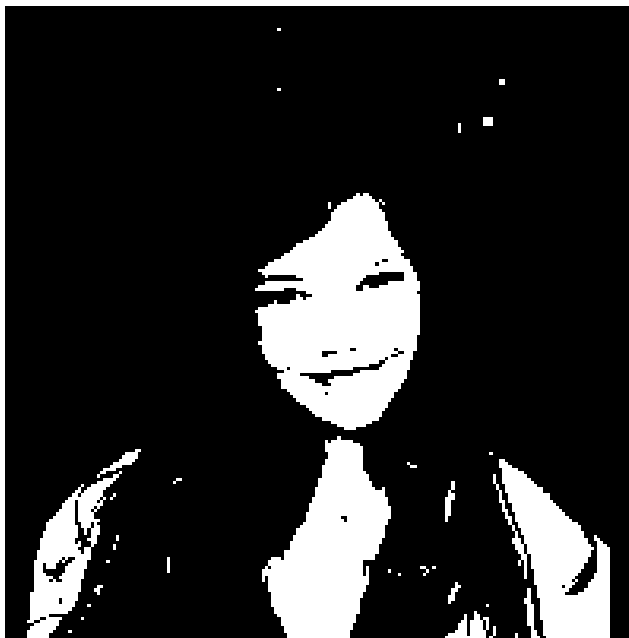
X 就是错误（原始颜色和灰度值（颜色值）之间的一个标量（矢量）差值），这个错误会向右边，右下，下边，和左下四个方向扩散，分别以7/16， 1/16， 5/16和3/16的权重添加到这四个像素点的值中去。感兴趣的用户可以去了解该算法，网络上有很多资源。

两种算法的处理效果比较

原图



“黑白色阶处理” 和 “多色色阶处理”



“黑白抖动处理” 和 “多色抖动处理”

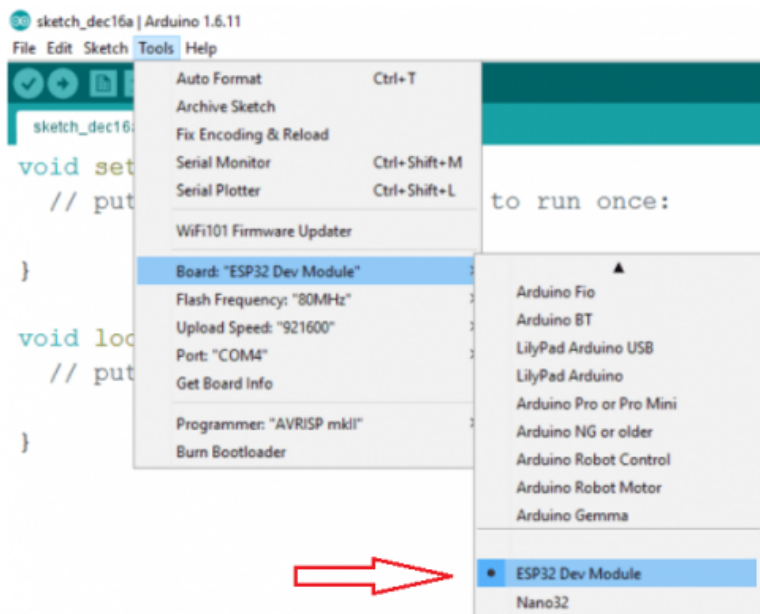


蓝牙例程

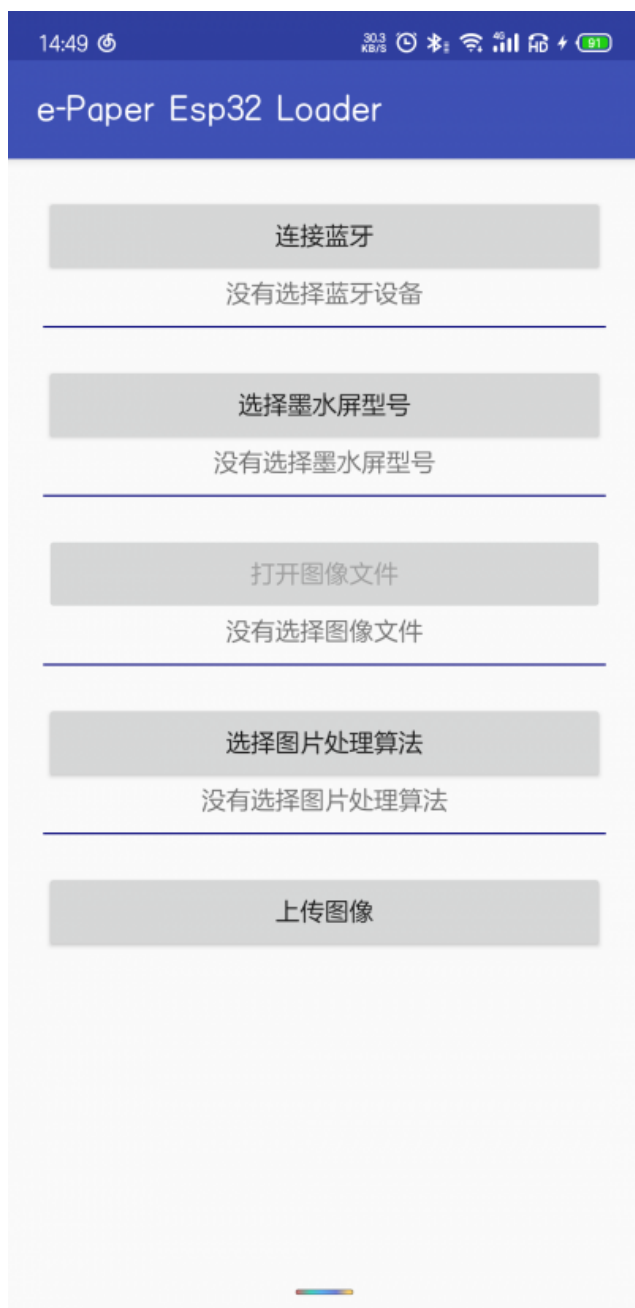
提供带安卓APP的蓝牙通信的例程

程序使用

- 打开Loader_esp32bt目录，双击Loader_esp32bt.ino文件打开Arduino工程
- 选择Tools->Boards->ESP32 Dev Module，并且选择好对应的串口：Tools->Port



- 然后点击上传，把程序编译并下载到 ESP32 驱动板上
- 手机安装并打开 APP:



■ APP 主页一共有五个按钮：

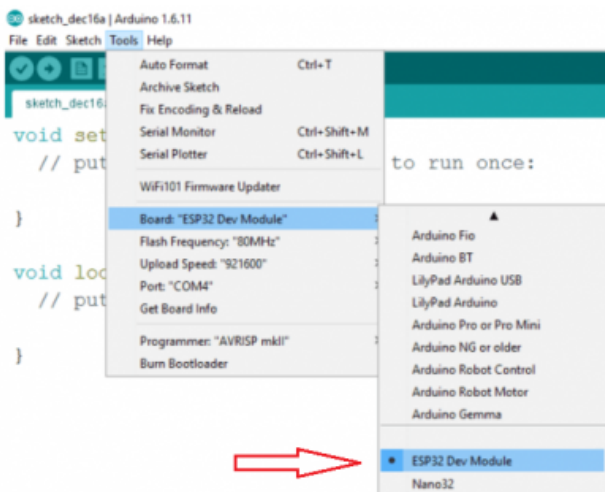
- 连接蓝牙：用来连接ESP32设备
 - 选择墨水屏型号：选择你接入到驱动板的墨水屏型号
 - 打开图像文件：点击可以选择手机里面的一张图片打开，必须选型号确定尺寸才能选择图片
 - 选择图片处理算法：由于手机中的图片并不一定符合墨水屏型号的需求，所以要先处理一下图片
 - 上传图片：将处理之后的图片上传到墨水屏，并刷新到屏幕上去
- 首先确保你已经打开手机蓝牙。点击 “连接蓝牙” -> 点击右上角的 “SCAN” 进行蓝牙设备扫描。
- 找到ESP32设备，点击进行连接。如果你是第一次连接这个设备，会弹出配对信息，点击确认完成配对。（注意：如果设备没有进行配对，将无法上传图片，并且可能出现APP闪退的问题）
- 点击 “选择墨水屏型号” 选择你连接的墨水屏对应型号
- 点击 “打开图像文件” 选择图库并打开一张图片，会根据上一步选择的型号出现矩形裁剪框，拖动裁剪到你想要的部分即可
- 点击 “选择图片处理算法” 选择对应的处理算法，并确认
- 黑白色阶算法（将图片处理成黑白两色，并根据墨水屏分辨率切割图片大小）
 - 彩色色阶算法（将图片处理成多色，并根据墨水屏分辨率切割图片大小，只适用于多色墨水屏）
 - 黑白抖动算法（将图片处理成黑白两色，并根据墨水屏分辨率切割图片大小）
 - 彩色抖动算法（将图片处理成多色，并根据墨水屏分辨率切割图片大小，只适用于多色墨水屏）
- 点击 “上传图片”，把处理过后的图像上传到墨水屏中显示

WiFi例程

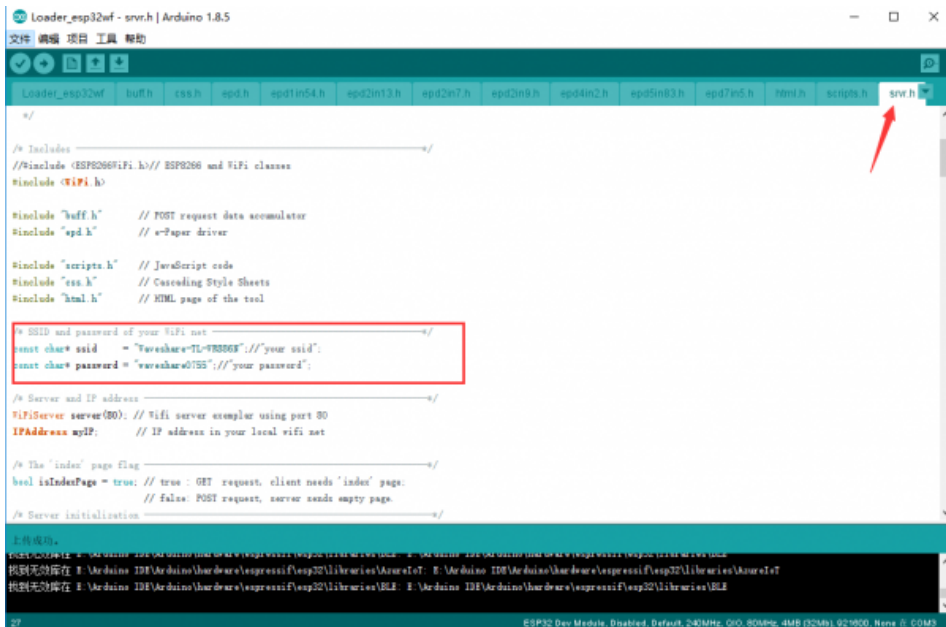
提供带HTML上位机的WiFi例程

程序使用

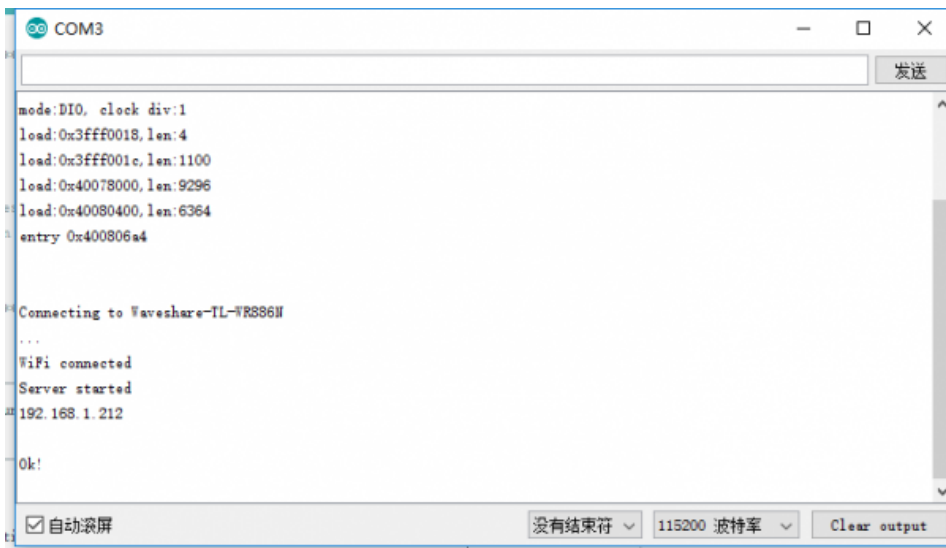
- 进入Loader_esp32wf文件夹，双击Loader_esp32wf.ino文件打开工程
- 选择Tools->Boards->ESP32 Dev Module，并且选择好对应的串口：Tools->Port



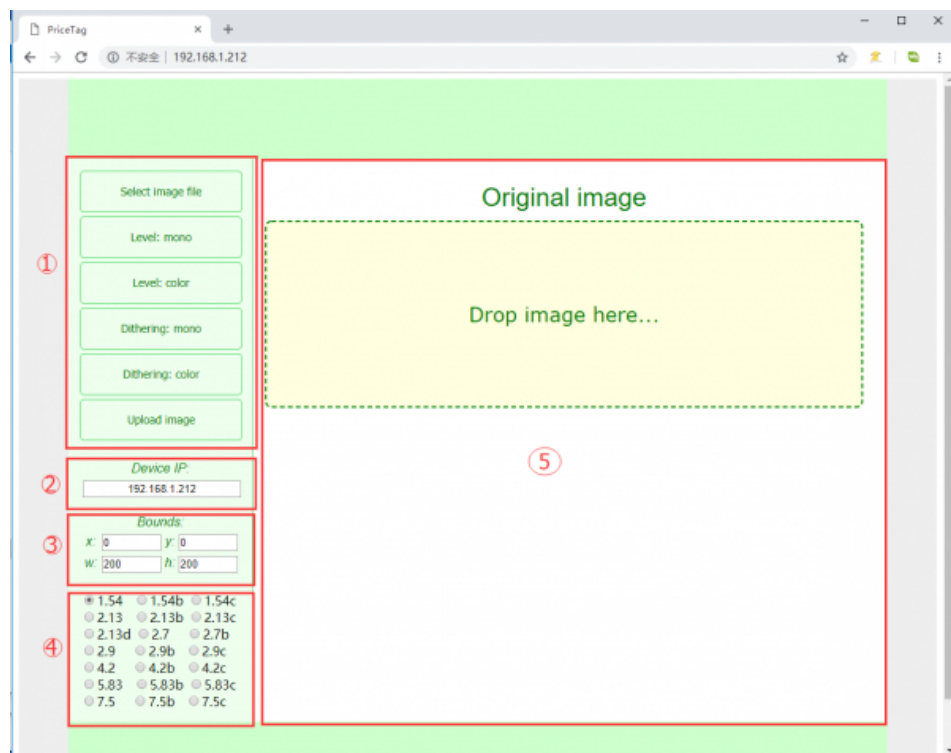
- 打开srvr.h文件，将ssid和password改成实际使用的WiFi用户名和密码



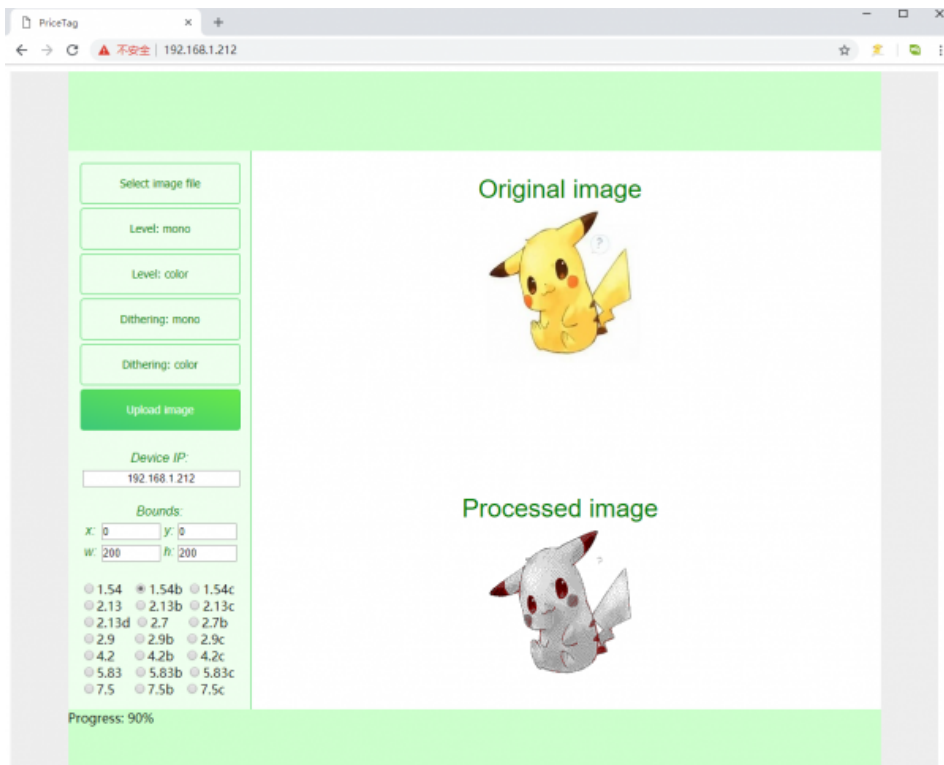
- 然后点击上传，把程序编译并下载到ESP32驱动板上
- 打开串口监视器，设置波特率为115200，可以看到串口将ESP32驱动板的IP地址打印出来：



- 电脑或者手机（注意电脑/手机接入的网络需要跟ESP32接入的wifi是同一个网段的才可以）打开浏览器，在网址输入栏输入ESP32的IP地址并打开，可以看到操作界面如下：



- 整个操作界面分为五个区域：
 - **图像操作区域：**
 - Select Image file: 点击在电脑或者手机里面选择一张图片
 - Level: mono: 黑白色阶图像处理算法
 - Level: color: 多色色阶图像处理算法（只对多色屏幕生效）
 - Dithering: mono: 黑色抖动图像处理算法
 - Dithering: color: 多色抖动图像处理算法（只对多色屏幕生效）
 - Update image: 上传图像
 - **IP信息显示区域：** 这里显示的是你当前连接的模块的IP地址信息
 - **图像大小设置区域：** 这里x和y可以设置你要显示的起始位置，这个设置是相对于你选择的图片文件的，比如选择一张800x480的图片，但是连接的墨水屏是2.9寸的，这时候墨水屏并无法显示整张图片的信息，所以在选择图像处理算法的时候，算法会自动从左上角开始截取一部分图片传到墨水屏显示，这里设置x和y可以自定义截取的起始位置。w和h是当前墨水屏的分辨率大小。
注意：如果修改了x和y的指的话，需要重新点击一下处理算法生成新的图像
 - **型号选择区域：** 这里可以选择你接入的墨水屏型号
 - **图像显示区域：** 这里会显示你选择的图片以及处理之后的图像
 - PS:在上传图像的时候，底部会显示上传的数据进度
- 区域①点击Select image file 选择一张图片，或者直接将图片拖拽至Original image的区域内
- 区域④选择对应的墨水屏型号，例如：1.54b
- 区域①点击一种图像处理算法，例如：Dithering: color
- 区域①点击Upload image将图片上传到墨水屏显示。

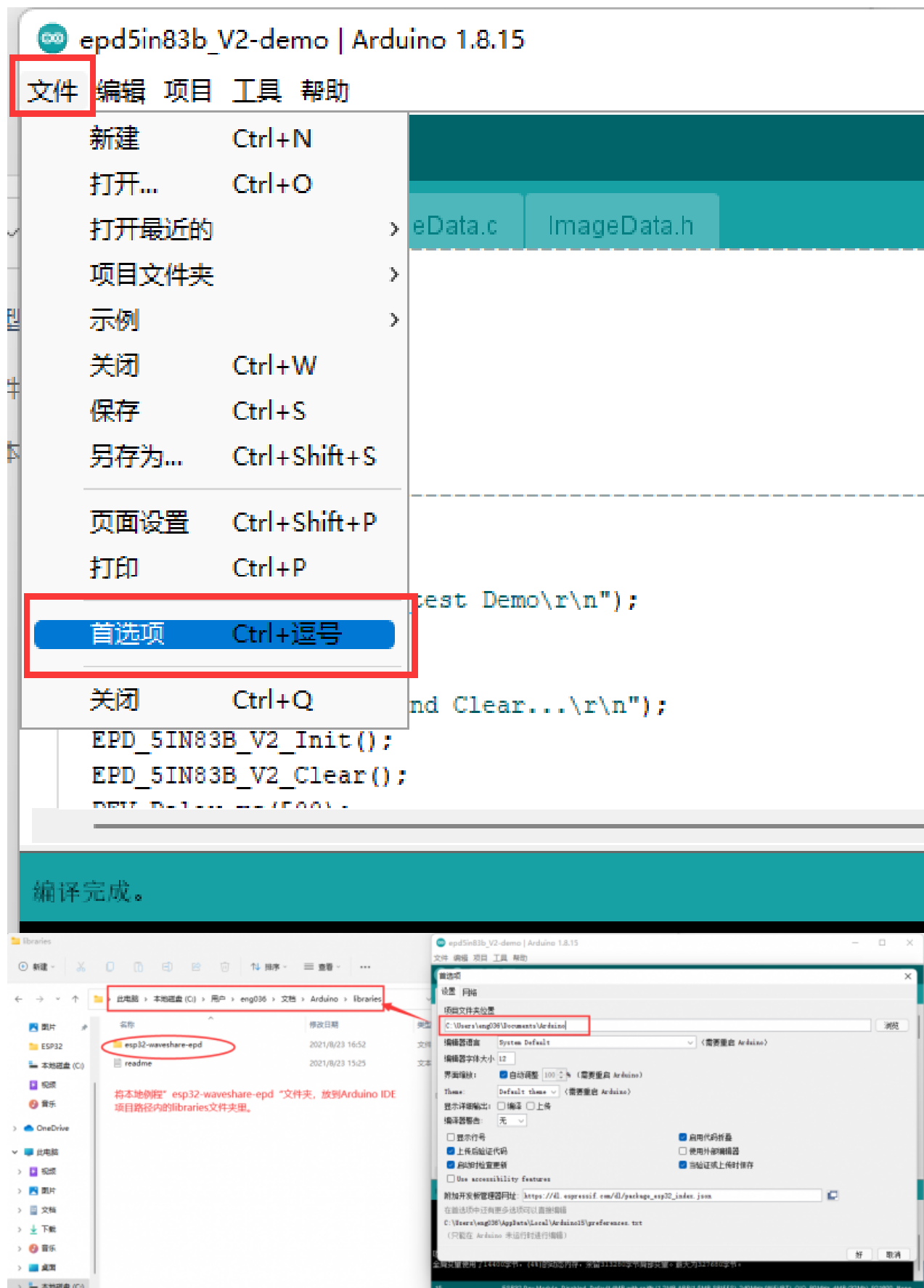


本地例程

提供基于ESP32的本地例程，无需WiFi、蓝牙和其他设备

程序使用

- 进入 examples\ 目录，把整个 esp32-waveshare-epd 文件夹复制到处于"Arduino IDE首选项" -> "文件项目路径" -> "libraries" 文件夹中



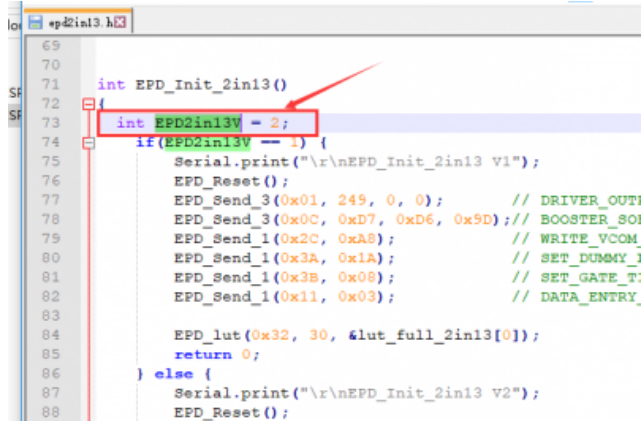
- 然后打开 Arduino IDE，在文件 - 示例中找到 waveshare-e-Paper，然后点击对应型号的示例程序

FAQ

E-Paper ESP32 Driver Board常见问题与解答

问题：2.13寸墨水屏（2.13inch e-paper）无法刷新，确认屏幕背面有没有V2标识，没有的话需要修改程序如下部分

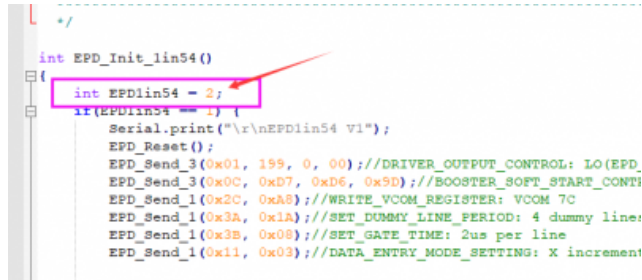
打开工程中的epd1in13.h,把如下值改成1



```
69
70
71 int EPD_Init_2in13()
72 {
73     int EPD2in13V = 2;
74     if (EPD2in13V == 1) {
75         Serial.print("\r\nEPD_Init_2in13 V1");
76         EPD_Reset();
77         EPD_Send_3(0x01, 249, 0, 0); // DRIVER_OUT1
78         EPD_Send_3(0x0C, 0xD7, 0xD6, 0x9D); // BOOSTER_SOI
79         EPD_Send_1(0x2C, 0xA8); // WRITE_VCOM
80         EPD_Send_1(0x3A, 0x1A); // SET_DUMMY_1
81         EPD_Send_1(0x3B, 0x08); // SET_GATE_T
82         EPD_Send_1(0x11, 0x03); // DATA_ENTRY_
83
84         EPD_lut(0x32, 30, &lut_full_2in13[0]);
85         return 0;
86     } else {
87         Serial.print("\r\nEPD_Init_2in13 V2");
88         EPD_Reset();
```

问题：1.54寸墨水屏（1.54inch e-paper）无法刷新，确认屏幕背面有没有V2标识，没有的话需要修改程序如下部分

打开工程中的epd1in54.h,把如下值改成1



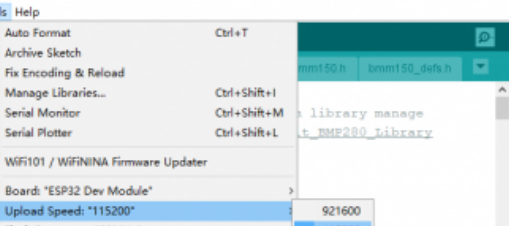
```
1 /*
2
3 int EPD_Init_1in54()
4 {
5     int EPD1in54 = 2;
6     if (EPD1in54 == 1) {
7         Serial.print("\r\nEPD1in54 V1");
8         EPD_Reset();
9         EPD_Send_3(0x01, 199, 0, 0); //DRIVER_OUTPUT_CONTROL: LO(EPD
10         EPD_Send_3(0x0C, 0xD7, 0xD6, 0x9D); //BOOSTER_SOFT_START_CONTI
11         EPD_Send_1(0x2C, 0xA8); //WRITE_VCOM_REGISTER: VCOM 7C
12         EPD_Send_1(0x3A, 0x1A); //SET_DUMMY_LINE_PERIOD: 4 dummy line:
13         EPD_Send_1(0x3B, 0x08); //SET_GATE_TIME: 2us per line
14         EPD_Send_1(0x11, 0x03); //DATA_ENTRY_MODE_SETTING: X increment
```

问题：ESP32下载蓝牙例程，模块报错：“Guru Meditation Error: Core 0 panic'ed (LoadProhibited). Exception was unhandled.” 且无法成功打开蓝牙，需要怎么解决？

下载Arduino-ESP32 支持包，并将压缩包里面的文件解压到 Arduino IDE 安装目录下的hardware\espressif\esp32 路径，选择“确定覆盖文件”（记得将原先文件备份），然后断电重新运行一次例程。（注意：如果在安装目录没有该路径，可以手动创建）

问题：用Arduino下载ESP32程序有时成功，有时失败，怎么解决？

尝试降低波特率，可尝试调为115200，如下图所示



The screenshot shows the Arduino IDE 1.8.9 interface. The 'Tools' menu is open, and the 'Upload Speed' option is selected, which has opened a sub-menu. In this sub-menu, '115200' is highlighted. Other options in the sub-menu include 921600, 256000, 230400, and 512000. The background code in the editor shows a C++ sketch for an ESP32, including headers like 'ESP32.h' and 'Adafruit_BME280.h', and defining pins and variables.

问题： wifi例程上传正常，串口输出了ip地址，但是电脑输入IP地址无法访问,需要检查ip的网段与wifi的网段值一致，且IP不冲突

修改ip网段，如下图所示

问题： 电脑识别不到驱动板，先确认串口驱动是否已经安装，再尽可能地更换USB线和USB接口尝试

mac串口驱动: https://www.silabs.com/documents/public/software/Mac_OSX_VCP_Driver.zip
window串口驱动: https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip

mac串口驱动: https://www.silabs.com/documents/public/software/Mac_OSX_VCP_Driver.zip
window串口驱动: https://www.silabs.com/documents/public/software/CP210x_Universal_Windows_Driver.zip

[illegible]

需要在出现Connecting...提示时，长按ESP32底板上的boot按键

问题： 蓝牙例程卡在0%

需要确认硬件连接正确，并选择对应的墨水屏型号

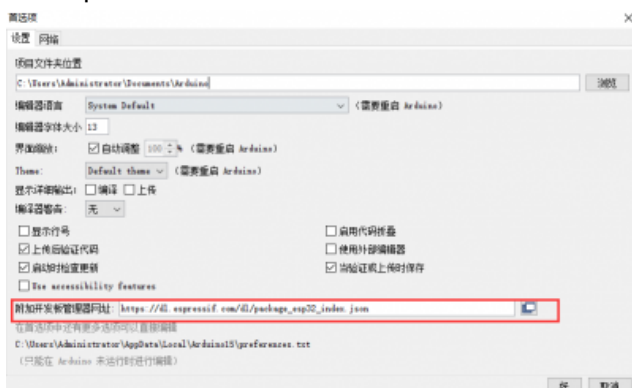
问题： 上传程序，报错开发板不存在或为空，需要确认端口和开发板选择正确，需要确认硬件连接正确，并选择对应的墨水屏型号

选择端口和驱动板，如下图所示



问题： 开板管理器搜索不到esp32，需要在菜单栏：文件->首选项里填入esp32开发板管理网址
https://dl.espressif.com/dl/package_esp32_index.json (esp8266:
http://arduino.esp8266.com/stable/package_esp8266com_index.json)

填入esp32开发板管理网址，如下图所示



问题： E-Paper ESP32驱动板A，B键的作用

兼容更多的墨水屏型号，可以根据显示效果调节

问题： E-Paper ESP32驱动板J3 J4之间的间距是多少

间距是22.65mm

