

E-Paper ESP32 Driver Board User Guide

This Wiki mainly introduces the specific operation of this product. If you want to get the type of the e-paper screen model supported by this product, please check the bottom of the official website for product details and purchase.

Introduction

Universal e-Paper Driver HAT features ESP32, supports various Waveshare SPI interface in e-Paper raw panels. It also supports refreshing image to e-paper via WIFI or Bluetooth and Arduino.

20220728: The serial port chip is changed from CP2102 to CH343, please pay attention to the driver selection.

Parameter

- WiFi Standard: 802.11b/g/n
- Communication Interface: SPI
- Bluetooth Standard: 4.2, BR/EDR and BLE included
- Communication Interface: 3-Wire SPI, 4-wire SPI (default)
- Working voltage: 5V
- Working Current: 50mA-150mA
- Outline dimension: 29.46mm x 48.25mm
- Flash size: 4 MB
- SRAM size: 520 KB
- ROM size: 448 KB

Pin

Pin	ESP32	Description
VCC	3.3V	Power input

GND	GND	Power ground
SCK	P13	SPI CLK, Clock Signal Input
DIN	P14	SPI MOSI, Data Input
CS	P15	Chip Select, active-low
BUSY	P25	Busy Output Pin (means busy)
RST	P26	Reset, active-low
DC	P27	Reset, Data/Demand, low level means to demand, high level means data

PS: The above is the board fixed connection, no additional operation by the user.

Feature

- Onboard ESP32, support Arduino development.
- Provide Android mobile APP program, which can update the display content through Bluetooth EDR, easy to use.
- Provide HTML host computer program, which can remotely update the display content through the web page, which is convenient to integrate into various network applications.
- Supports Floyd-Steinberg dithering algorithm for more color combinations and better shadow of the original image.

- Supports many common image formats (BMP, JPEG, GIF, PNG, etc.).
- Factory built-in e-ink screen driver (open source).
- 5V pin supports 3.6V to 5.5V voltage input, and can be powered by a lithium battery.
- Provide complete supporting information manual.

Application

This product cooperates with the ink screen and is suitable for the application scenario of wireless refreshing.

- Supermarket electronic price tag
- Electronic name card
- Serial information display board, etc.

Preparation

Hardware Connection

This product is shipped with an ESP32 network driver board, an adapter board, and an FFC extension cable.

When using it, you can directly connect the screen to the driver board, or connect it through extension cables and adapter boards.

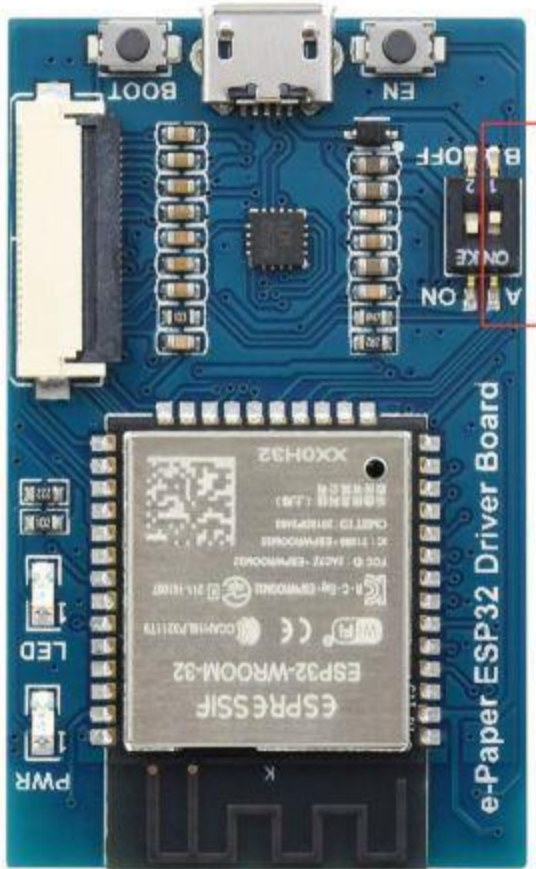
Direct access to the driver board:



Access via extension cord:



- Set the mode switch: Set the No. 1 switch according to the model of the EPD used. There are many screens. If it is not listed, please use 'A' to try. If the display effect is poor or cannot be driven, please try switching the switch.



Resistor (Display Config)

Screen

0.47R (A)

2.13inch e-Paper (D), 2.7inch e-Paper, 2.9inch e-Paper (D)
 3.7inch e-Paper, 4.01inch e-Paper (F), 4.2inch e-Paper
 4.2inch e-Paper (B), 4.2inch e-Paper (C), 5.65inch e-Paper (F)
 5.83inch e-Paper, 5.83inch e-Paper (B), 7.3inch e-Paper (G)
 7.3inch e-Paper (F), 7.5inch e-Paper, 7.5inch e-Paper (B)
 1.64inch e-Paper (G), 2.36inch e-Paper (G), 3inch e-Paper (G)
 4.37inch e-Paper (G)

3R (B)

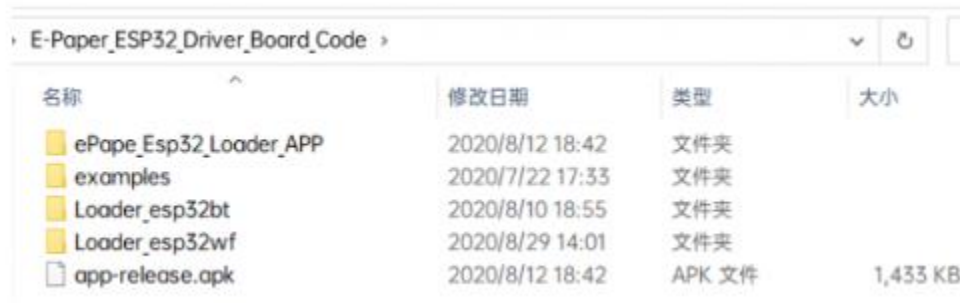
1.54inch e-Paper, 1.54inch e-Paper(B), 2.13inch e-Paper
 2.13inch e-Paper (B), 2.66inch e-Paper, 2.66inch e-Paper (B)
 2.9inch e-Paper, 2.9inch e-Paper (B)

- Turn on the serial port module: Toggle the No. 2 switch to "ON", this switch controls the power supply of the USB to the UART module. When you don't need to use it, you can manually turn off the module to save power (if switch 2 is in the OFF state, you cannot upload the program.)
- Use a micro USB cable to connect the ESP32 driver board to a computer or 5V power supply.

Download Demo

We provide three kinds of demos: local, bluetooth, and WiFi. The sample program can be found in [#Resources](#), or click [the sample program](#) to download.

Unzip the downloaded compressed package, you can get the following files:



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: Bluetooth App source code (Android Studio)
- examples: local demo
- Loader_esp32bt: Bluetooth demos
- Loader_esp32wf: WiFi demo
- app-release.apk: Bluetooth demo App installation package

Environment Configuration

- [Arduino ESP32/8266 Online Installation](#)

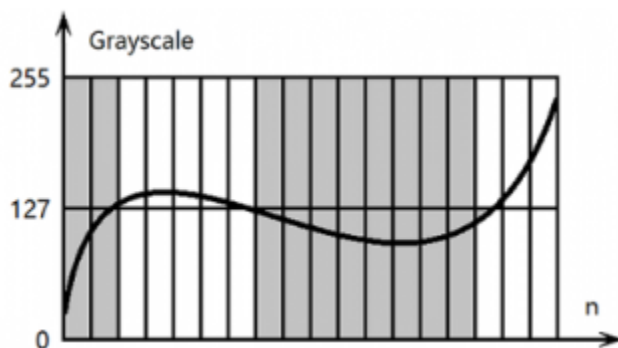
Image Processing Algorithms

In the Bluetooth and WiFi demos, two image processing algorithms are provided, namely Level and Dithering.

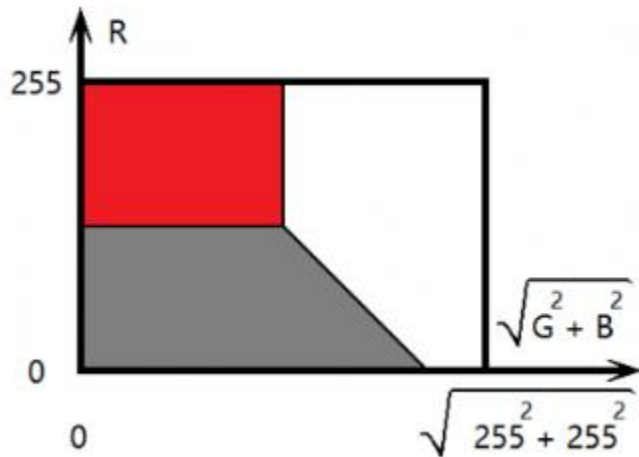
- **color scale method**

An image can be divided into several large color gamuts, and each pixel on the image is divided into these color gamuts according to how close the color is to these color gamuts. This method is more suitable for images with few colors, such as bright or tri-color shapes or text images. Taking the black and white and red ink screen as an example, when processing the image, we hope to process it into black, white and red, so for an image, we can divide all the colors of the image into three large color areas: the black area, white area, red area.

For example, according to the figure below, if the value of a pixel in the grayscale image is equal to or less than 127, we regard this pixel as a black pixel, otherwise, it is white



For color images, we all know that RGB has three color channels. Compared with the red channel, we can refer to blue and green as the blue-green channel or the non-red channel. According to the figure below, a pixel on a color image, if it has a high value in the red channel, but a low value in the blue-green channel, we classify it as a red pixel; if its red channel and blue- If the green channel has low values, we classify it as a black pixel; if the red and blue-green channel values are high, we classify it as white.



In the algorithm, the color definition is calculated based on the difference between the RGB value and the sum of the squares of the expected color value. The expected color value refers to the color value that the pixel is closest to, and these values are stored in the curPal array

```
// 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;
}
```

- ## Dithering

For those images with more colors or more gradient areas, the above gradation method is not suitable. In many cases, the pixels in the gradient area in the image may be very close to all color gamuts. If you use the gradation method to draw the image will lose a lot of image details. Many images taken by cameras, by mixing colors to paint shadows and transition

areas, in these images, the gradient area accounts for the majority. For the human eye, it is easy to confuse a particularly small color. For example, two colors, red and blue, are juxtaposed. If you reduce it to a small enough hand, it will appear to the human eye as a mixture of red and blue. The defect of the human eye means that we can deceive the human eye and use the "mixing" method to obtain more colors that can be expressed. The dithering algorithm uses this phenomenon. The routine we provide uses the Floyd-Steinberg dithering algorithm - based on error diffusion (published by Robert Floy and Louis Steinberg in 1976). The formula is for error diffusion according to the image below



X is the error (a scalar (vector) difference between the original color and the gray value (color value)), this error will spread to the right, lower right, lower, and lower left in four directions, respectively $7/16$, $1/16$, $5/16$ and $3/16$ weights are added to the values of these four pixels. Interested users can go to understand the algorithm, there are many resources on the Internet.

- **Comparison**
-

Original image



"Black and white grading" and "Multicolor grading"



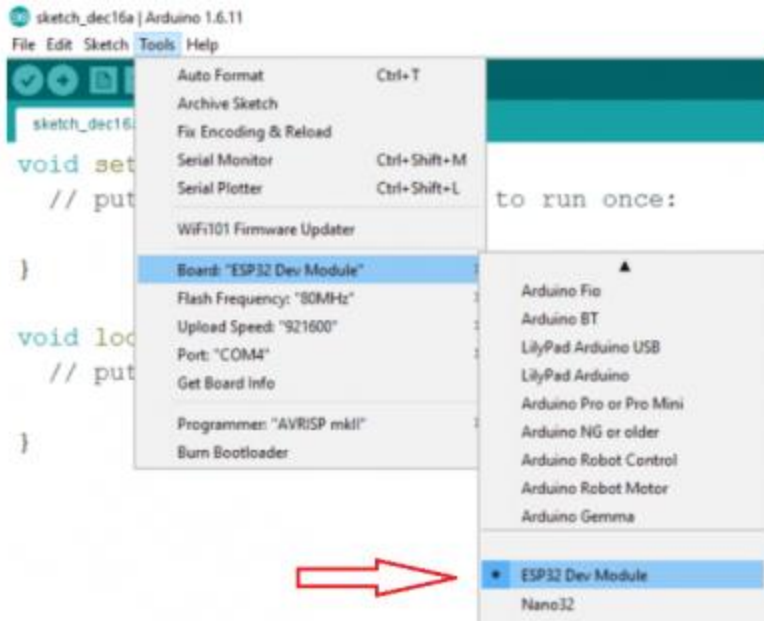
"Black and White Dithering" and "Multicolor Dithering"



Bluetooth Demo

Download example

- Go to the Loader_esp32bt directory, double click the Loader_esp32bt.ino file to open the example.
- Choose Tools->Boards->ESP32 Dev Module and select the correct Port according to Device Manager: Tools->Port



- Click the Upload icon to build the project and upload it to the ESP32 driver board.
- Install the APP to the Android board and open it:

e-Paper Esp32 Loader

BLUETOOTH CONNECTION

Bluetooth device is not selected

LOAD IMAGE FILE

Image file is not loaded

SELECT DISPLAY TYPE

Display type is not selected

SELECT IMAGE FILTER

Image filter is not loaded

UPLOAD IMAGE

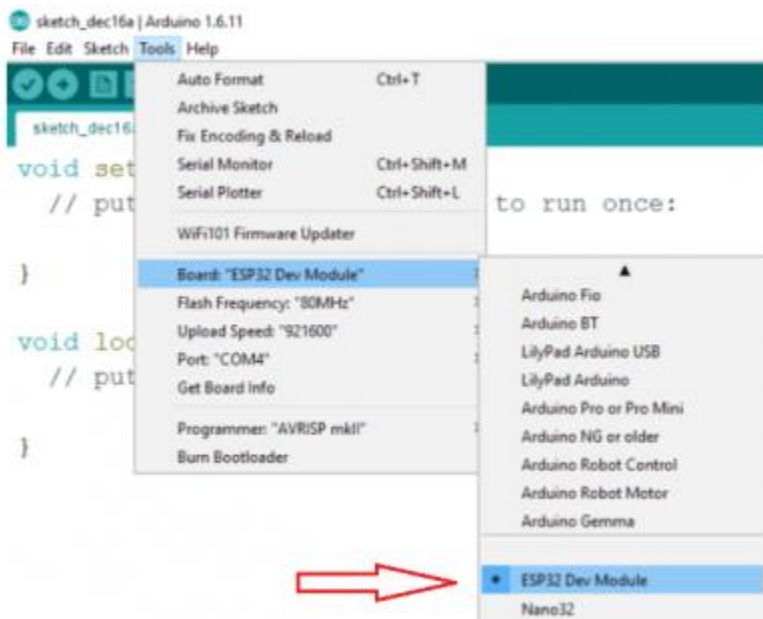
- APP has five buttons on the main page:
 - BLUETOOTH CONNECTION: This button is used to connect the ESP32 device via Bluetooth.
 - SELECT DISPLAY TYPE: This button is used to select the type of display according to what you buy.
 - LOAD IMAGE FILE: Click it and choose a picture to open. It is only available after choosing the display type.
 - SELECT IMAGE FILTER: This button is used to choose the image process method.
 - UPLOAD IMAGE: Upload the processed image to ESP32 driver board and update to e-Paper display.
- Please first open the Bluetooth function of your phone. Click the BLUETOOTH CONNECTION button -> Click the SCAN icon on the top-right to scan the Bluetooth device.
- Find the ESP32 device and connect. If your phone is the first time to connect this device, it requires pairing, complete the pairing process according to the prompt. (Note: The APP cannot work with pairing.)
- Click "SELECT DISPLAY TYPE" to choose the display type.
- Click "LOAD IMAGE FILE" To choose a picture from your phone and cut it.
- Click " SELECT IMAGE FILTER" to choose a process algorithm and confirm.
 - "LEVEL: MONO": This option will process the picture to a monochrome image.
 - "LEVEL" COLOR": This option will process the picture to the tricolor image according to the display colors of the display (only valid for colorful displays).

- "DITHERING: MONO": This option will process the picture to a monochrome image.
- "DITHERING: COLOR": This option will process the picture to the tricolor image according to the display colors of the display (only valid for colorful displays).
- Click "UPLOAD IMAGE" to upload the image to the ESP32 device and display it.

WiFi Demo

How to Use

- Go to the Loader_esp32wf directory, double click Loader_esp32wf.ino file to open the project.
- Choose Tools->Boards->ESP32 Dev Module in IDE menu, and select the correct COM port: Tools->Port



- Open the srvr.h file, modify the ssid and password to your actual WiFi used.

```
Loader_esp32wf - srvr.h | Arduino 1.8.5
文件 编辑 项目 工具 帮助

/*
 * Includes
 */
//Include (ESP8266WiFi.h) // ESP8266 and WiFi classes
#include <WiFi.h>

#include "buff.h" // POST request data accumulator
#include "spd.h" // e-paper driver

#include "scripts.h" // JavaScript code
#include "css.h" // Cascading Style Sheets
#include "html.h" // HTML page of the tool

/* SSID and password of your WiFi net */
const char* ssid = "wvsnharc-TL-9R3068";//your ssid;
const char* password = "wvsnharc0155";//your password;

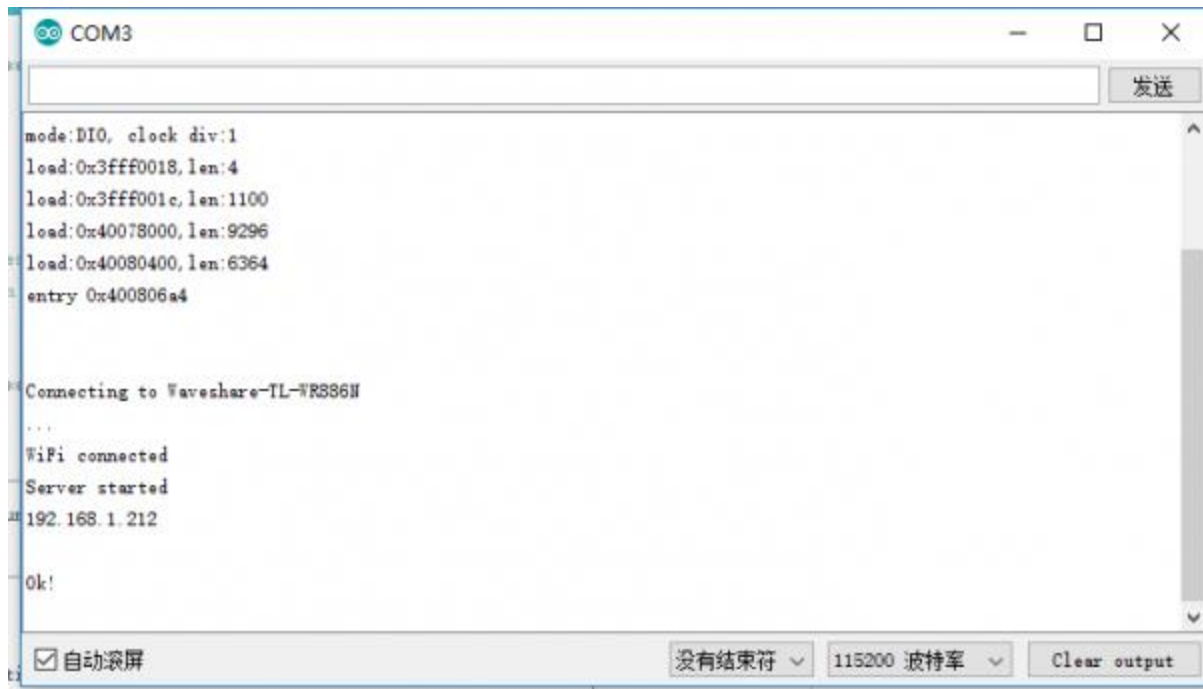
/* Server and IP address */
WiFiServer server(80); // WiFi server exemplar using port 80
IPAddress myIP; // IP address in your local wifi net

/* The 'index' page flag */
bool isIndexPage = true; // true : GET request, client needs 'index' page.
// false: POST request, server sends empty page.

/* Server initialization */


```

- Click the Upload icon to build the project and upload it to the ESP32 driver board.
- Open the serial monitor, set the baud rate to 115200. Then you can get the information printed like the IP of the ESP32 board.



```
mode:DIO, clock div:1
load:0x3fff0018, len:4
load:0x3fff001c, len:1100
load:0x40078000, len:9296
load:0x40080400, len:6364
entry 0x400806a4

Connecting to Waveshare-TL-WR886H
...
WiFi connected
Server started
192.168.1.212

Ok!
```

COM3

发送

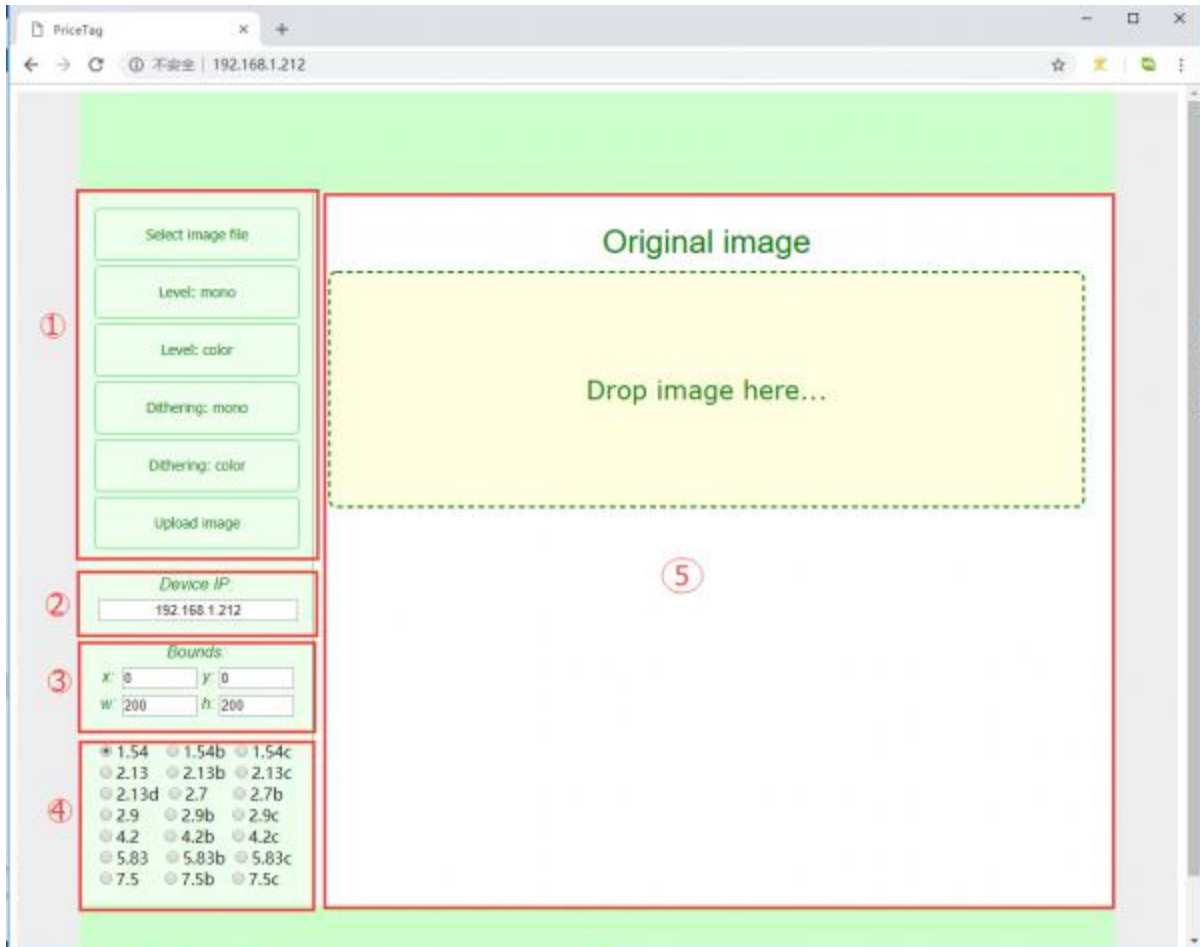
自动滚屏

没有结束符

115200 波特率

Clear output

- Open a browser on the PC or mobile phone (Note that you should connect your PC/Phone to the same network as the ESP32 board), go to the webpage with the IP address of the ESP32 board:



- There are five areas in the webpage:
 - **Image Operation:**
 - Select Image file: Click the option to choose a picture from PC/phone

Level: mono: Black/White image level processing.

Level: color: Colorful image level processing (only valid for colorful display)

Dithering: mono: Black/White image Dithering processing

Dithering: color: Colorful image Dithering processing (only valid for colorful display)

Update image: Upload the image to the ESP32 board and display it.

-

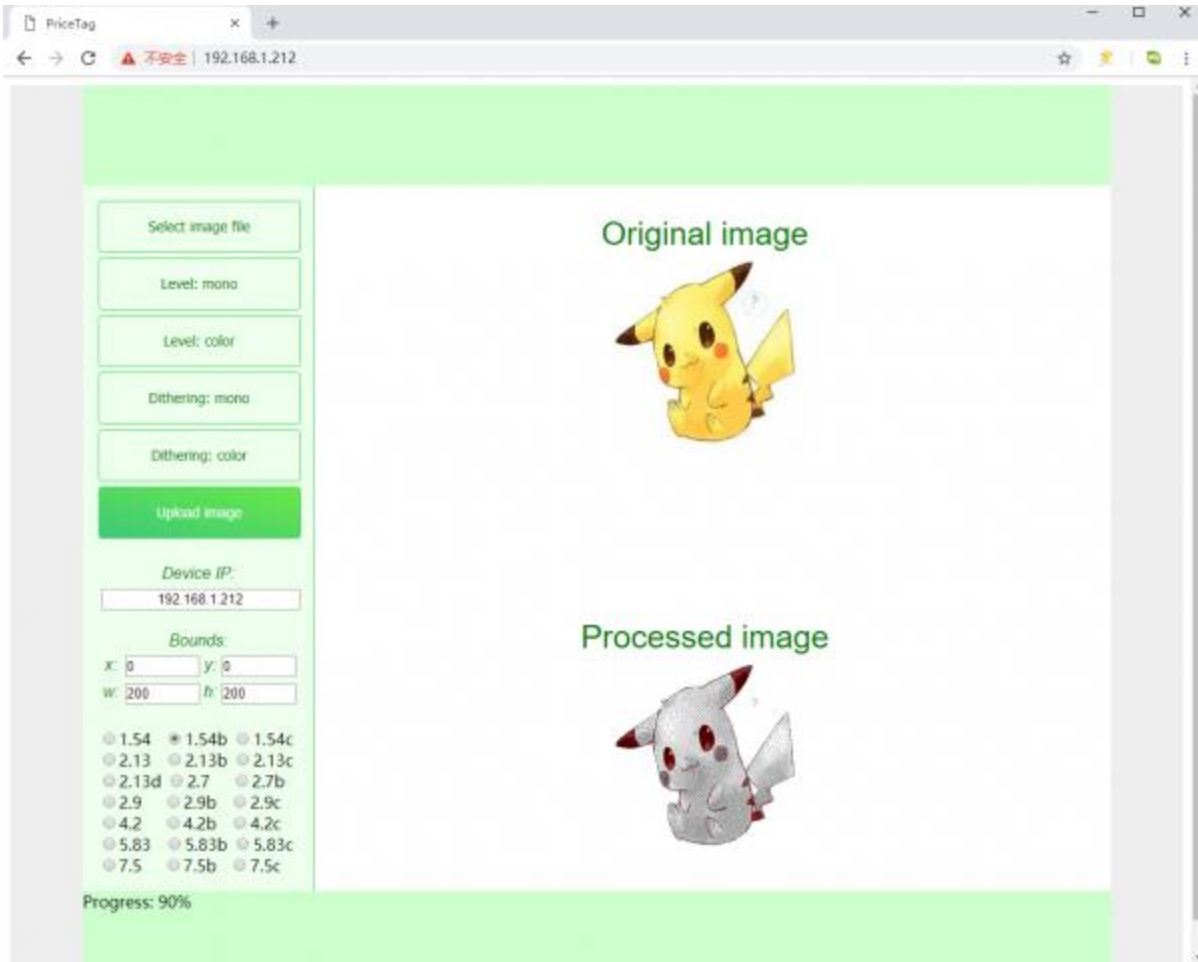
- **IP Information:** In this are, the IP information of the ESP32 board connected is listed.
- **Image Size Setting:** x and y are the beginning position of the image. For example, if you select an image with 800x480 resolution, and the display you connect is 2.13. The 2.13inch display cannot display the whole picture, the codes will crop the picture according to the resolution of displays. If you set the x and y, the codes will set the (x, y) as the being pixel where it starts to crop. w and h is the resolution of the displays chosen.

Note: If you modify the x and y value, you need to click the Level or Dithering options to process the image again.

- - **Display Type:** You should first select the display size here according to the display used.
 - **Image Preview:** it will display the original image selected and the image processing.
 - PS: The upload progress value is printed at the bottom of the webpage when uploading.

For examples:

- In area ① click Select image file to choose a picture, you can also drag the picture to the Original image area directly.
- In area ④ choose the type according to the display, for example, 1.54b
- In area ① choose an algorithm to process the image, for example, Dithering: color
- In area ① click Upload image to upload the image and display it.



Local Demo

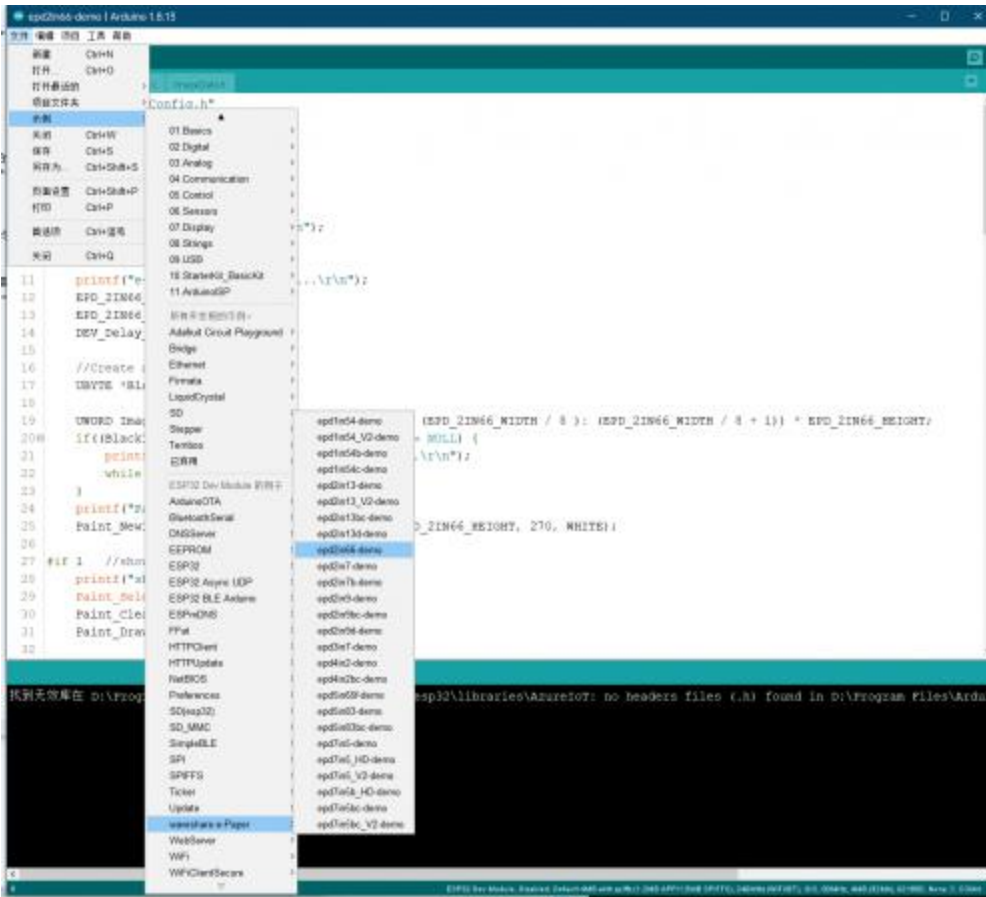
Download example

- Go to examples\ directory. copy the whole esp32-waveshare-epd folder to [Installation directory of Arduino IDE]\hardware\espressif\esp32\libraries.

› Arduino › hardware › espressif › esp32 › libraries ›

名称	修改日期	类型	大小
ArduinoOTA	2020/7/22 17:32	文件夹	
AsyncUDP	2020/7/22 17:32	文件夹	
AzureIoT	2020/7/19 7:21	文件夹	
BLE	2020/7/22 17:32	文件夹	
BluetoothSerial	2020/7/22 17:32	文件夹	
DNSServer	2020/7/22 17:32	文件夹	
EEPROM	2020/7/22 17:32	文件夹	
ESP32	2020/7/22 17:32	文件夹	
esp32-waveshare-epd	2020/9/1 15:14	文件夹	
ESPmDNS	2020/7/22 17:32	文件夹	
FFat	2020/7/22 17:32	文件夹	
FS	2020/7/22 17:32	文件夹	
HTTPClient	2020/7/22 17:32	文件夹	
HTTPUpdate	2020/7/22 17:32	文件夹	
NetBIOS	2020/7/22 17:32	文件夹	
Preferences	2020/7/22 17:32	文件夹	
SD	2020/7/22 17:32	文件夹	
SD_MMC	2020/7/22 17:32	文件夹	
SimpleBLE	2020/7/22 17:32	文件夹	
SPI	2020/7/22 17:32	文件夹	
SPIFFS	2020/7/22 17:32	文件夹	
Ticker	2020/7/22 17:32	文件夹	
Update	2020/7/22 17:32	文件夹	
WebServer	2020/7/22 17:32	文件夹	
WiFi	2020/7/22 17:32	文件夹	
WiFiClientSecure	2020/7/22 17:32	文件夹	
Wire	2020/7/22 17:32	文件夹	
README.md	2020/7/19 7:21	MD 文件	3 KB

- Open the Arduino IDE, Find the waveshare-e-Paper examples from File->Example, and select one according to the display type.



- Click Upload to build the project and upload it to the ESP32 board. You can open the Serial Monitor to check the debug information.

Resources

Documentation

- [Schematic](#)

ESP32 Resources

- [ESP32 Resouces](#)
- [ESP32 Datasheet](#)

Demo Code

- [Demo code](#)

Related Resources

This is a post in Arduino Form about our SPI e-Paper thanks to ZinggJM, maybe you want to refer to.

- [Waveshare e-Paper display with SPI](#)

Software Driver

CP2102 (Old version, used before July 2022)

- [CH343 VCP driver for Windows](#)
- [CH343 driver for MacOS](#)
- [MacOS guide](#)

CH343 (New version, used after July 2022)

- [Windows VCP driver](#)
- [MAC driver](#)

FAQ

[Question:If you don't have a V2 logo on the back of your 2.13inch e-paper screen, how do I use it?](#)

Answer:

- Open epd2in13.h in the project and change the following value to 1

```
epd2in13.h
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_OUTPUT
78         EPD_Send_3(0x0C, 0xD7, 0xD6, 0x9D); // BOOSTER_SOFT_START_CONTI
79         EPD_Send_1(0x2C, 0xA8); // WRITE_VCOM
80         EPD_Send_1(0x3A, 0x1A); // SET_DUMMY_LINE_PERIOD: 4 dummy line
81         EPD_Send_1(0x3B, 0x08); // SET_GATE_TIME: 2us per line
82         EPD_Send_1(0x11, 0x03); // DATA_ENTRY_MODE_SETTINGS: X increment
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();
```

[Question:If you don't have a V2 logo on the back of your 1.54inch e-paper screen, how do I use it?](#)

Answer:

* Open epd1in54.h in the project and change the following value to 1

```

/*
int EPD_Init_1in54()
{
    int EPD1in54 = 2;
    if(EPD1in54 == 1) {
        Serial.print("\r\nEPD1in54 V1");
        EPD_Reset();
        EPD_Send_3(0x01, 199, 0, 0); //DRIVER_OUTPUT_CONTROL: 10(EPD
        EPD_Send_3(0x0C, 0xD7, 0xD6, 0x9D); //BOOSTER_SOFT_START_CONTI
        EPD_Send_1(0x2C, 0xA8); //WRITE_VCOM REGISTER: VCOM 7c
        EPD_Send_1(0x3A, 0x1A); //SET_DUMMY_LINE_PERIOD: 4 dummy line
        EPD_Send_1(0x3B, 0x08); //SET_GATE TIME: 2us per line
        EPD_Send_1(0x11, 0x03); //DATA_ENTRY_MODE_SETTINGS: X increment
    }
}
*/
```

[Question:ESP32 downloads the Bluetooth demo, and the module reports an error: "Guru Meditation Error: Core 0 panic'ed \(LoadProhibited\). Exception was unhandled." and the Bluetooth cannot be turned on successfully. What should I do?](#)

Answer:

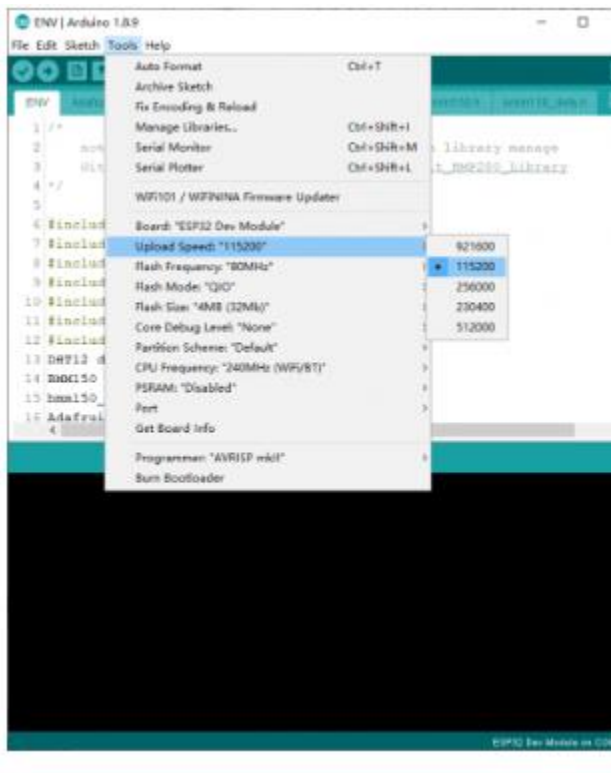
Download [Arduino-ESP32 Package](#), Unzip the files in the compressed package to the hardware\espressif\esp32 path in the Arduino IDE installation directory, select "OK to overwrite the file" (remember to back up the original file), and then

re-run the routine after power off. (Note: If the path does not exist in the installation directory, you can create it manually)

Question: [Downloading ESP32 program with Arduino sometimes succeeds and sometimes fails, how to solve it?](#)

Answer:

Try to reduce the baud rate, you can try to adjust to 115200, as shown in the figure below



Question: [The wifi routine upload is normal, the serial port outputs the ip address, but the computer input IP address cannot be accessed, it is necessary to check that the network segment of the ip is consistent with the network segment value of the wifi, and the IP does not conflict](#)

Answer:

Modify the ip network segment, as shown in the following figure

```
Loader_esp32wf  buff.h  css.h  epd.h  epd1in54.h  epd2in13.h  epd2in66.h  epd2in7.h  epd2in9.h  epd3in7.h  epd4in01f.h  epd4in7.h

#include "buff.h" // POST request data accumulator
#include "epd.h" // e-Paper driver

#include "scripts.h" // JavaScript code
#include "css.h" // Cascading Style Sheets
#include "html.h" // HTML page of the tool

/* SSID and password of your WiFi net -----*/
const char *ssid = "JSBPI"; // "your ssid";
const char *password = "waveshare0755"; // "your password";

/* Static IP address Settings -----*/
IPAddress staticIP(192, 168, 10, 113);
IPAddress gateway(192, 168, 10, 113);
IPAddress subnet(255, 255, 255, 0);
IPAddress dns(223, 5, 5, 5);

/* Server and IP address -----*/
WiFiServer server(80); // Wifi server exemplar using port 80
IPAddress myIP; // IP address in your local wifi net
```

Question: If the computer does not recognize the driver board, first confirm whether the serial port driver has been installed, and then try to replace the USB cable and USB interface as much as possible.

Answer:

- [CH343 VCP driver for Windows](#)
- [CH343 driver for MacOS](#)
- [MacOS guide](#)

Question: Burning and uploading program error:

Answer:

Connecting.....
....._____....._____....._____Error uploading project_A fatal error occurred:
Failed to connect to ESP32: Timed out waiting for packet header

You need to press and hold the boot button on the ESP32 baseboard when the Connecting... prompt appears

Question:[Bluetooth demo stuck at 0%](#)

Answer:

It is necessary to confirm that the hardware connection is correct, and select the corresponding ink screen model

Question:[When uploading the program, an error is reported that the development board does not exist or is empty, you need to confirm that the port and development board are selected correctly, you need to confirm that the hardware connection is correct, and select the corresponding ink screen model](#)

Answer:

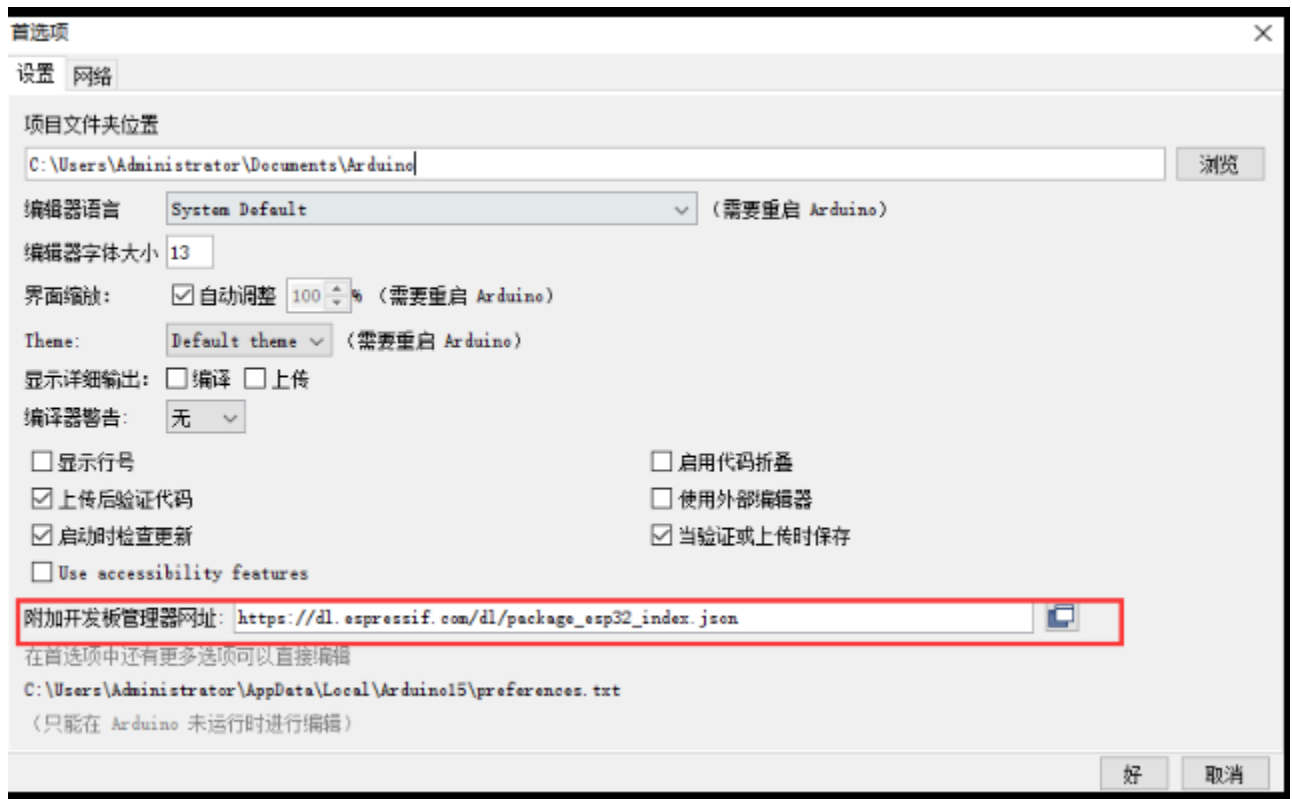
Select the port and driver board as shown below



Question:[The board manager cannot search for esp32, you need to fill in the esp32 development board management URL](#)

Answer:

https://dl.espressif.com/dl/package_esp32_index.json (esp8266: <http://arduino>) in the menu bar: File -> Preferences
.esp8266.com/stable/package_esp8266com_index.json)



Question: E-Paper ESP32 driver board A, B key function

Answer:

Compatible with more ink screen models, which can be adjusted according to the display effect

Question: What is the spacing between J3 and J4 of the E-Paper ESP32 driver board

Answer:

Spacing is 22.65mm

