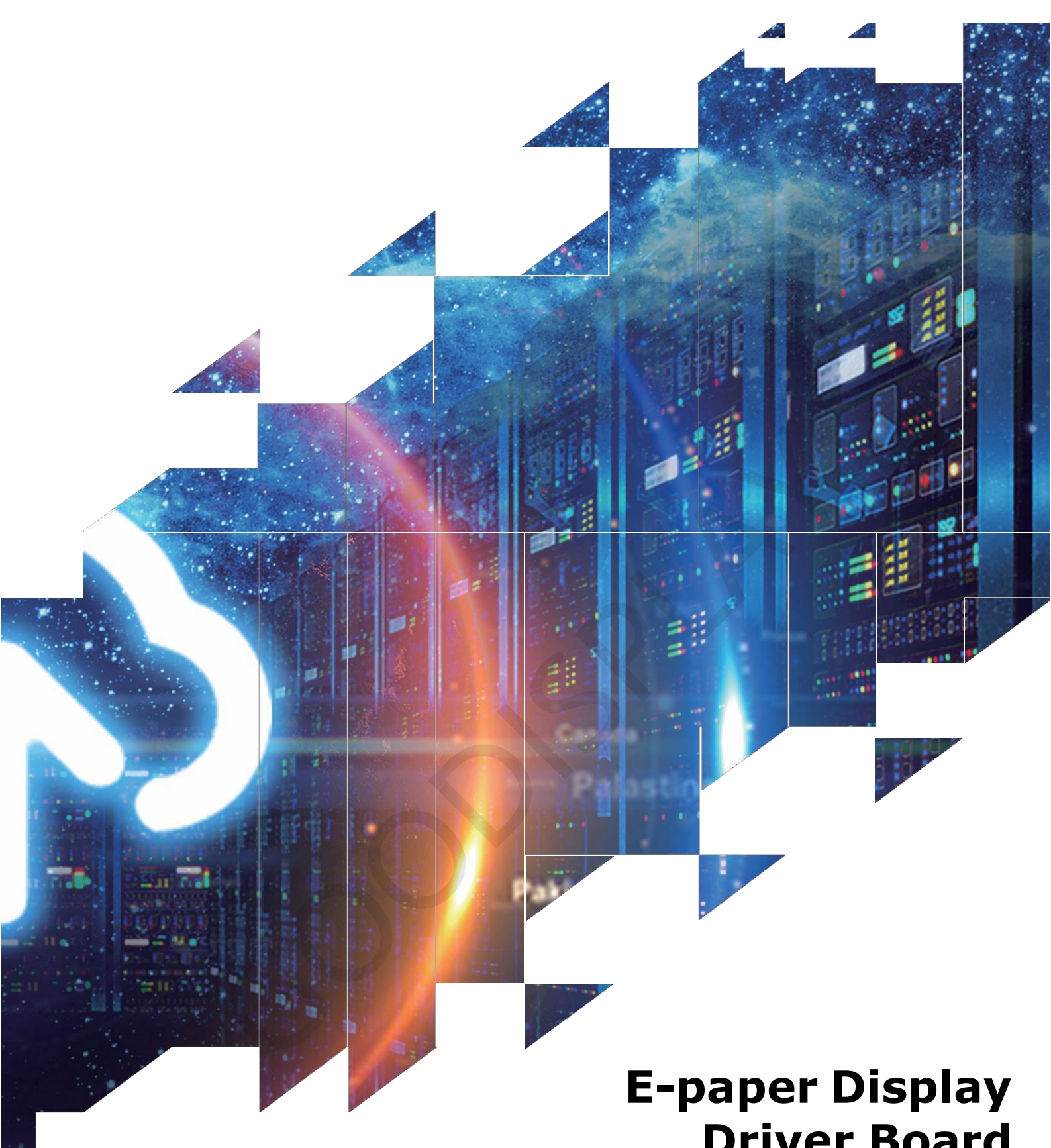




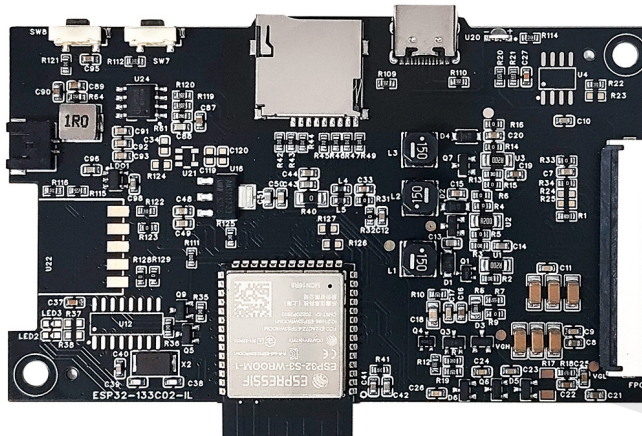
E-paper Display Driver Board



ESP32-133C02-IL

Dalian Good Display Co., Ltd.

Product Specifications



Customer	Standard
Description	E-paper display driver board
Model Name	ESP32-133C02-IL
Date	2026/03/12
Revision	2.0

	Design Engineering		
	Approval	Check	Design
			

Zhongnan Building, No.18, Zhonghua West ST,Ganjingzi DST,Dalian,CHINA

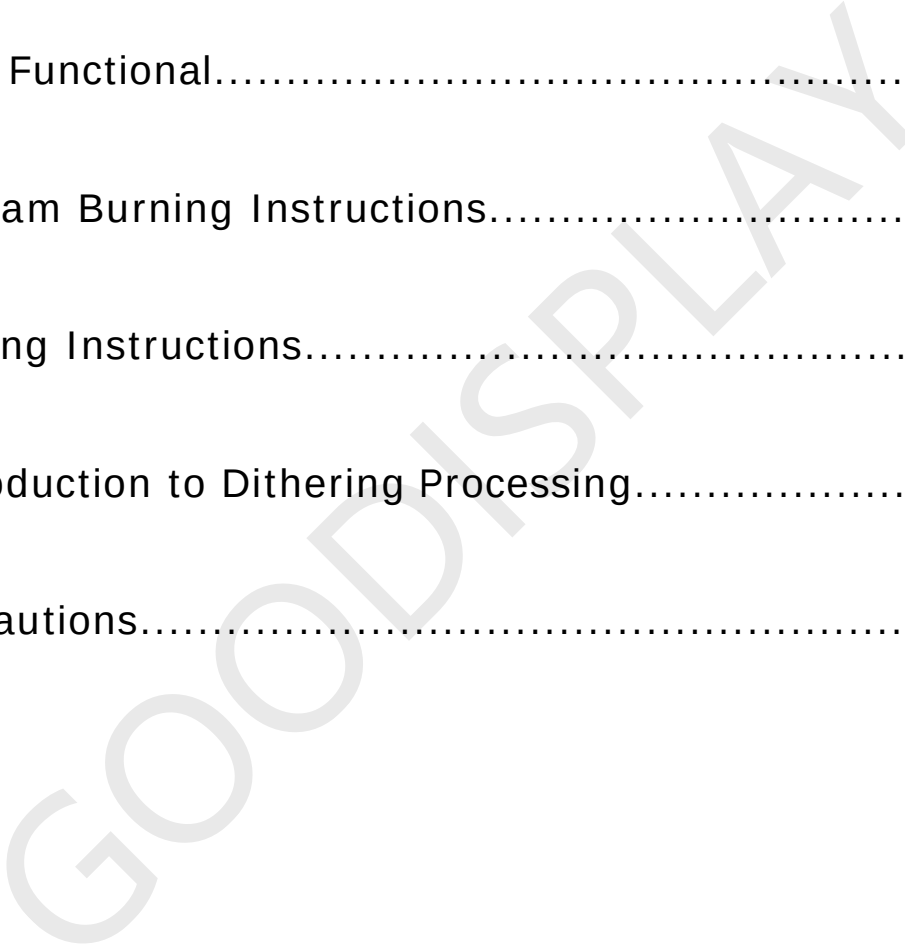
Tel: +86-411-84619565

Email: info@good-display.com

Website: www.good-display.com

Contents

1. Overview.....	4
2. Parameters.....	4
3. Main Functional.....	5
4. Program Burning Instructions.....	7
5. Driving Instructions.....	17
6. Introduction to Dithering Processing.....	20
7. Precautions.....	23



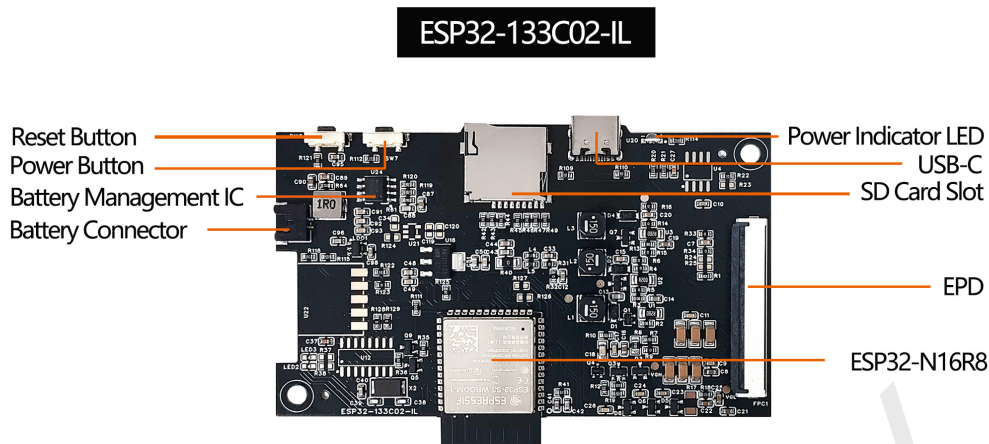
1. Overview

The ESP32-133C02-IL development board is specifically designed to drive the 13.3-inch color E Ink Spectra 6 display, model GDEP133C02. It supports image demonstration, image updates via PC software, wireless image transmission through Wi-Fi, and various other functions. Our intention in designing this development board goes beyond simply assisting developers in creating e-paper display projects more conveniently. We also aim for users to directly use this board for prototype development, significantly reducing the time required for development and production.

2. Specifications

Model Name	ESP32-133C02-IL
Platform Used	ESP-IDF/Arduino IDE
Dimensions(mm)	98.298mm x 69.215mm
Supported Size	13.3-inch e-paper
Power Supply	Powered via a Type-C adapter or a Type-C lithium battery.
Example Programs	Available
Operating Temperature	0°C to +50°C
Auxiliary Functions	Indicator light, lithium battery power supply, etc.
Main Functions	Learn how to drive an e-paper display; Test and evaluate the e-paper display; Perform secondary development based on this board: update images via PC software; Store and update images from an SD card; Update images via Wi-Fi wireless communication; Display fixed image content.

3. Main Functional



3.1 Function Buttons

Four function buttons (SW2, SW3, SW4) are reserved on the development board. All three buttons are triggered by a high level. The pin definitions are as follows:

SW2-----IO12;

SW3-----IO13;

SW4-----IO14.

3.2 SD card

An SD card slot is reserved on the development board. Users can expand the storage space of the development board via a memory card. The pin definitions are as follows:

SD CS-----IO15;

SD MOSI-----IO3;

SD MISO-----IO5;

SD SCK-----IO8.

3.3 EPD

EPD_MOSI-----IO41;

EPD_MISO-----IO40;

EPD_CS_M-----IO18;

EPD_CS_S-----IO17;

EPD_RST-----IO6;

EPD_DC-----IO2;

EPD_BUSY-----IO7;

EPD_SCK-----IO9;

This development board provides four functional models, and the corresponding firmware needs to be burned:

Model	Function Description
ESP32-133C02-IL-USB	Image transmission via USB upper computer
ESP32-133C02-IL-SD	Image update via SD card
ESP32-133C02-IL-WIFI	Wireless image transmission via WiFi (Web/ mobile terminal)
ESP32-133C02-IL-CH	Fixed image sequence demonstration

4. Program Burning Instructions

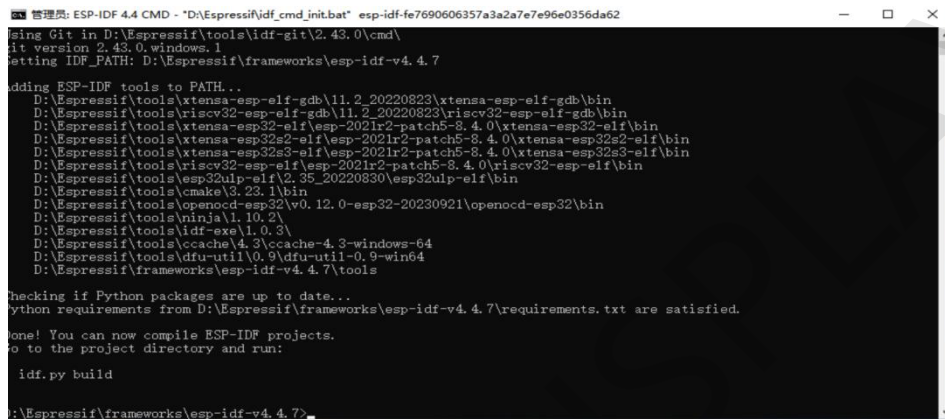
4.1 Download Software (ESP-IDF 5.5.2 CMD) (Provided on Espressif Official Website)

Download Links:

<https://dl.espressif.com/dl/idf-installer/esp-idf-tools-setup-offline-5.5.2.exe>

<https://dl.espressif.cn/dl/esp-idf/?idf=4.4>

After installing the program downloaded from the official website, run `idf_cmd_init.bat`.



```
管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62
Using Git in D:\Espressif\tools\idf-git\2.43.0\cmd\
Git version 2.43.0.windows.1
Setting IDF_PATH: D:\Espressif\frameworks\esp-idf-v4.4.7

Adding ESP-IDF tools to PATH...
D:\Espressif\tools\xtensa-esp-elf-gdb\11.2.20220823\xtensa-esp-elf-gdb\bin
D:\Espressif\tools\riscv32-esp-elf-gdb\11.2.20220823\riscv32-esp-elf-gdb\bin
D:\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32-elf\bin
D:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s2-elf\bin
D:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s3-elf\bin
D:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch5-8.4.0\riscv32-esp-elf\bin
D:\Espressif\tools\esp32ulp-elf\2.35.20220830\esp32ulp-elf\bin
D:\Espressif\tools\cmake\3.23.1\bin
D:\Espressif\tools\openocd-esp32\v0.12.0-esp32-20230921\openocd-esp32\bin
D:\Espressif\tools\ninja\1.10.2\
D:\Espressif\tools\idf-exe\1.0.3\
D:\Espressif\tools\ccache\4.3\ccache-4.3-windows-64
D:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
D:\Espressif\frameworks\esp-idf-v4.4.7\tools

Checking if Python packages are up to date...
Python requirements from D:\Espressif\frameworks\esp-idf-v4.4.7\requirements.txt are satisfied.

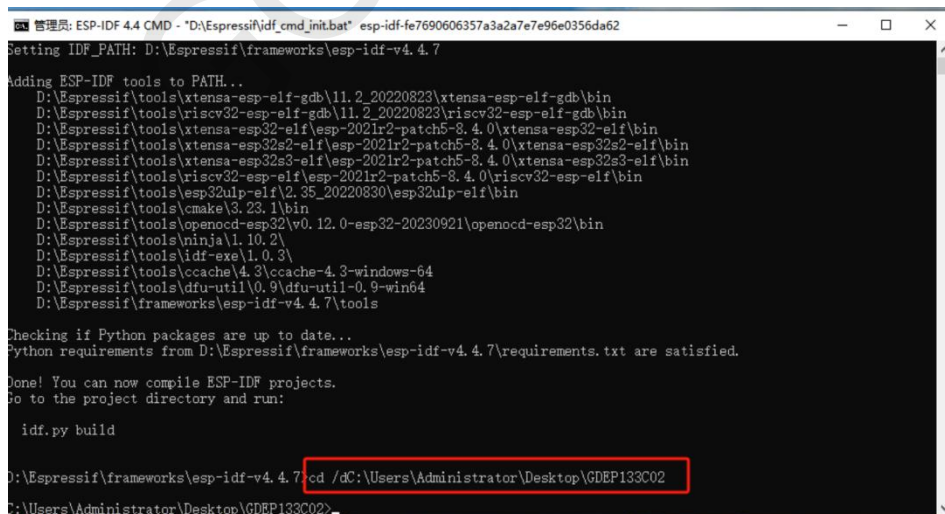
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7>
```

4.2 Navigate the Compiler Path

First, enter "`cd /d` " + the program storage path and press Enter to navigate the compiler to the target directory. (Please download the relevant programs for E-paper displays from our official website. Note that the program path should not be too long or contain Chinese characters, otherwise the compiler will report an error.)



```
管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62
Setting IDF_PATH: D:\Espressif\frameworks\esp-idf-v4.4.7

Adding ESP-IDF tools to PATH...
D:\Espressif\tools\xtensa-esp-elf-gdb\11.2.20220823\xtensa-esp-elf-gdb\bin
D:\Espressif\tools\riscv32-esp-elf-gdb\11.2.20220823\riscv32-esp-elf-gdb\bin
D:\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32-elf\bin
D:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s2-elf\bin
D:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s3-elf\bin
D:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch5-8.4.0\riscv32-esp-elf\bin
D:\Espressif\tools\esp32ulp-elf\2.35.20220830\esp32ulp-elf\bin
D:\Espressif\tools\cmake\3.23.1\bin
D:\Espressif\tools\openocd-esp32\v0.12.0-esp32-20230921\openocd-esp32\bin
D:\Espressif\tools\ninja\1.10.2\
D:\Espressif\tools\idf-exe\1.0.3\
D:\Espressif\tools\ccache\4.3\ccache-4.3-windows-64
D:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
D:\Espressif\frameworks\esp-idf-v4.4.7\tools

Checking if Python packages are up to date...
Python requirements from D:\Espressif\frameworks\esp-idf-v4.4.7\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7> cd /d C:\Users\Administrator\Desktop\GDEP133C02
C:\Users\Administrator\Desktop\GDEP133C02>
```

4.3 Compile the Program

Enter "`idf.py build` " and press Enter to compile the program.

```

选择 管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62 - python.exe "D:\Espressif\frameworks\esp-idf-v4.4.7\cd /d C:\Users\Administrator\Desktop\GDEP133C02
idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7>cd /d C:\Users\Administrator\Desktop\GDEP133C02
C:\Users\Administrator\Desktop\GDEP133C02>idf.py build
Executing action: all (aliases: build)
Running cmake in directory C:\Users\Administrator\Desktop\GDEP133C02\build
Executing "cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 -DIDF_TARGET=esp32s3 -DOCACHE_ENABLE=1 C:\Users\Administrator\Desktop\GDEP133C02\..."
-- Found Git: D:/Espressif/tools/idf-git/2.43.0/cmd/git.exe (found version "2.43.0.windows.1")
-- ccache will be used for faster recompilation
-- The C compiler identification is GNU 8.4.0
-- The CXX compiler identification is GNU 8.4.0
-- The ASM compiler identification is GNU
-- Found assembler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-gcc.exe
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-g++.exe - skipped
-- Detecting CXX compile features

```

4.4 Burn the Program

Enter "idf.py flash monitor" to start burning the program and wait for the completion of burning.

```

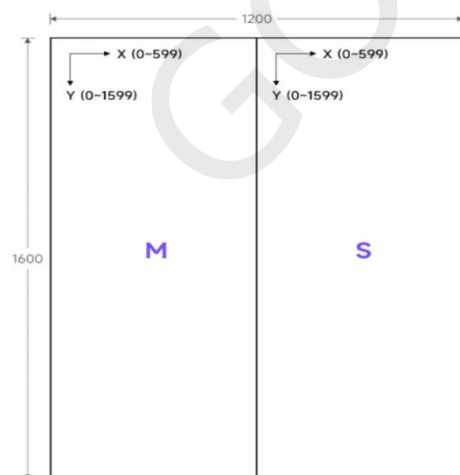
选择 管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62 - python.exe "D:\Espressif\frameworks\esp-idf-v4.4.7\cd /d C:\Users\Administrator\Desktop\GDEP133C02
idf.py flash monitor

project build complete. To flash, run this command:
D:\Espressif\python_env\idf4.4_py3.11\env\Scripts\python.exe D:\Espressif\frameworks\esp-idf-v4.4.7\components\esptool_py\esptool.py -p (PORT) -b 460800 --before=default_reset --after=hard_reset --chip esp32s3 write_flash --flash_mode dio --flash_size detect --flash_freq 80m 0x0 build/bootloader/bootloader.bin 0x8000 build/partition_table/partition-table.bin 0x10000 build\133_ReferenceDesign_SampleCode.bin
or run 'idf.py -p (PORT) flash'

C:\Users\Administrator\Desktop\GDEP133C02>idf.py flash monitor
Executing action: flash
Serial port COM55
Connecting...
Detecting chip type... ESP32-S3
Running minija in directory C:\Users\Administrator\Desktop\GDEP133C02\build
Executing "ninja flash"...
[1/5] cmd.exe /C "cd /D C:\Users\Administrator\Desktop\GDEP133C02\build\133_ReferenceDesign_SampleCode.bin"
133_ReferenceDesign_SampleCode.bin binary size 0x1234b0 bytes. Smallest app partition is 0x1f0000 bytes. 0xc50 bytes (0.1%) free.
[2/5] Performing build step for 'bootloader'
[1/1] cmd.exe /C "cd /D C:\Users\Administrator\Desktop\GDEP133C02\build\bootloader\esp-idf\esptool_py && D:\Espressif\python_env\idf4.4_py3.11\env\Scripts\python.exe D:\Espressif\frameworks\esp-idf-v4.4.7\components\partition_table\check_size.py --offset 0x8000 bootloader 0x0 C:\Users\Administrator\Desktop\GDEP133C02\build\bootloader\bootloader.bin"
bootloader binary size 0x52b0 bytes. 0x2d50 bytes (35%) free.
[2/3] cmd.exe /C "cd /D D:\Espressif\frameworks\esp-idf-v4.4.7\components\esptool_py\run_serial_tool.cmake"
esptool.py esp32s3 -p COM55 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 80m --flash_size 16MB 0x0 bootloader/bootloader.bin 0x10000 133_ReferenceDesign_SampleCode.bin 0x8000 partition_table/partition-table.bin
esptool.py v3.3.4-dev
Serial port COM55
Connecting...

```

4.5 Dual-IC Programming Analysis



GDEP133C02 is an E-ink screen controlled by dual ICs. Please note the following points:

- As shown in the figure above, each IC controls half of the screen, and two CS pins are used to control the corresponding M area (main area) and S area (slave area). Except for the CS pins, other control pins are shared.
- Except for common commands (e.g., initialization commands and refresh commands in the routine), only one CS pin is allowed to be pulled low at a time. Special attention should be paid during image transmission: the screen is displayed in landscape orientation (X-axis direction).
- After reaching the end of one area, proceed to the next row for display; therefore, the image needs to be split into two images for separate transmission, otherwise the display will be misaligned.

5. Driving Instructions

For the ESP32-133C02-IL development board, our company has developed three image refresh methods as follows:

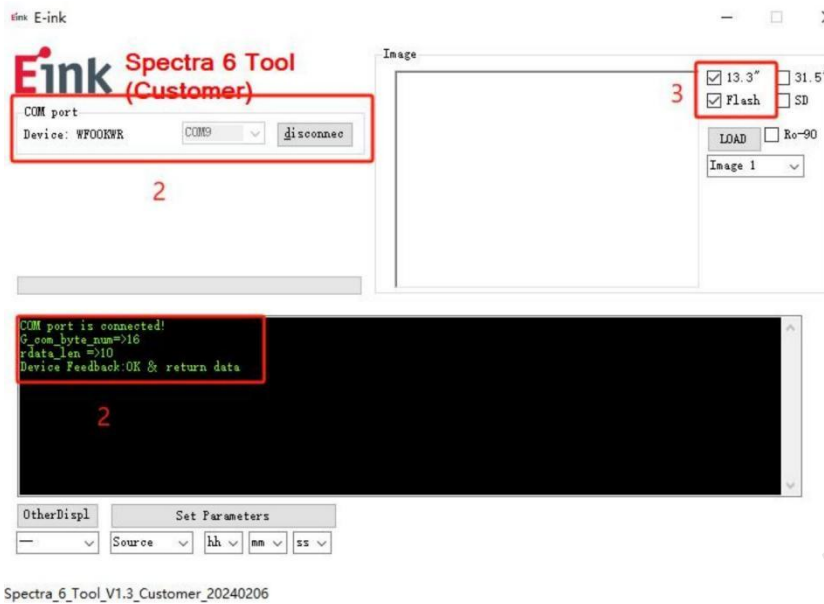
1. Update images via USB communication between host computer and development board;
2. Update images via ESP32 parsing images stored in SD card;
3. Update images via WiFi connection between mobile phone and development board through web application.

Method 1: Update Images via USB Communication Between Host Computer and Development Board

First, connect the MCU development board to the EPD, and power it via the Type-C port or battery interface. Then connect the development board to the PC port via the USB communication interface.

Install and open Spectra 6_Image_Tool.exe (available for download on our official website). After successful installation, open the software and click "connect".

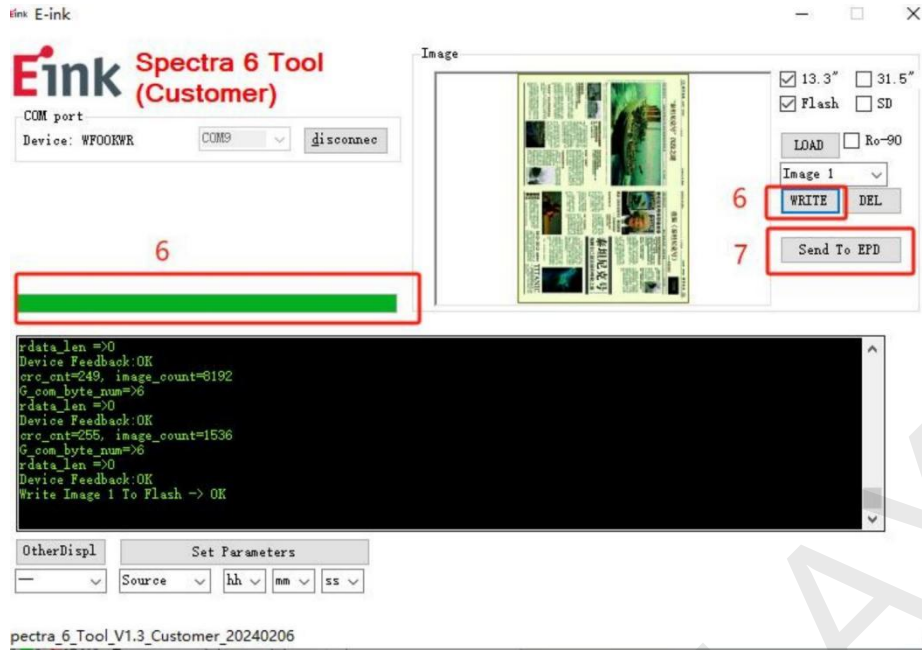
- (1) If the connection is successful, the message "WF00KWR" will appear on the device.
- (2) If no message appears on the device, check the Device Manager to confirm the successful connection of the COM port.
- (3) After a successful connection, check "13.3" and Flash.



- (4) Click "LOAD" and select an image from the folder (Resolution: 1200X1600, and the image must be pre-dithered).
- (5) Select the image storage location (Image1-15 optional).



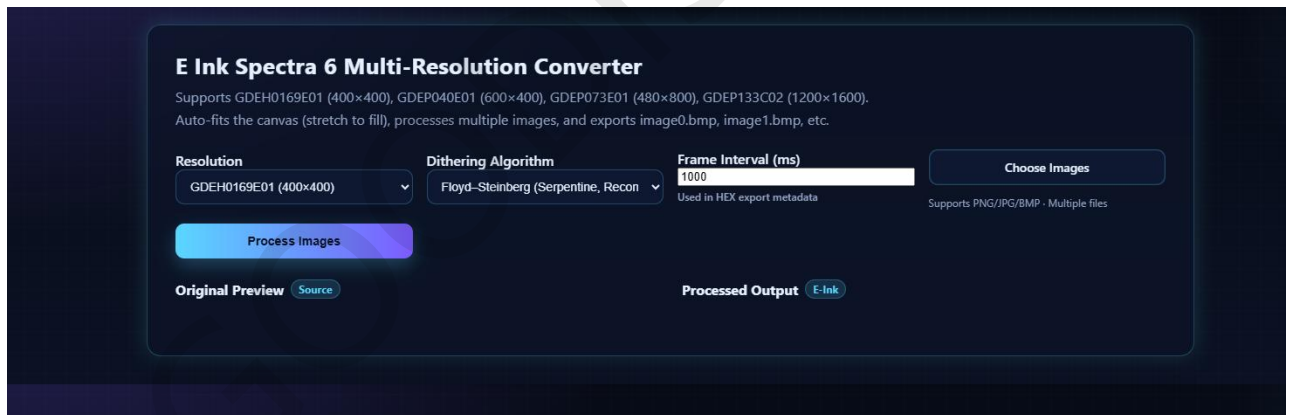
- (6) Click "WRITE" to burn the image information into the MCU and wait for the progress bar to complete.
- (7) After the burning is completed, click "Send To EPD", and the EPD will display the image at the Imagexx location and start refreshing.



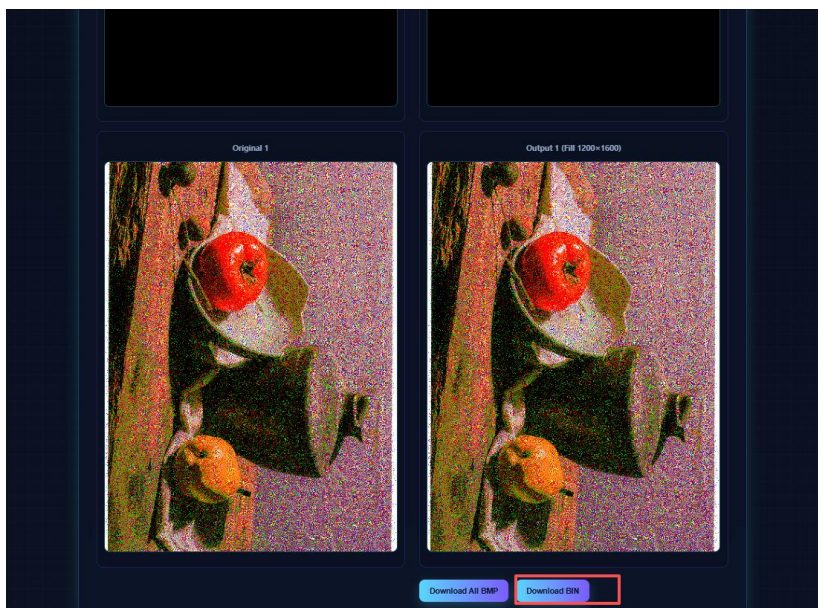
Method 2: Update Images via ESP32 Parsing Images Stored in SD Card

Open the image processing website:

<https://www.e-paper-display.com/bmp.html>



- 1.First, select the corresponding E6 product model by size. You can select the dithering type (Dithering Algorithm) and set the image refresh interval (Frame Interval (ms)).
- 2.Click "Choose Images" to select the target images, then click "Process Images" to generate dithered images.



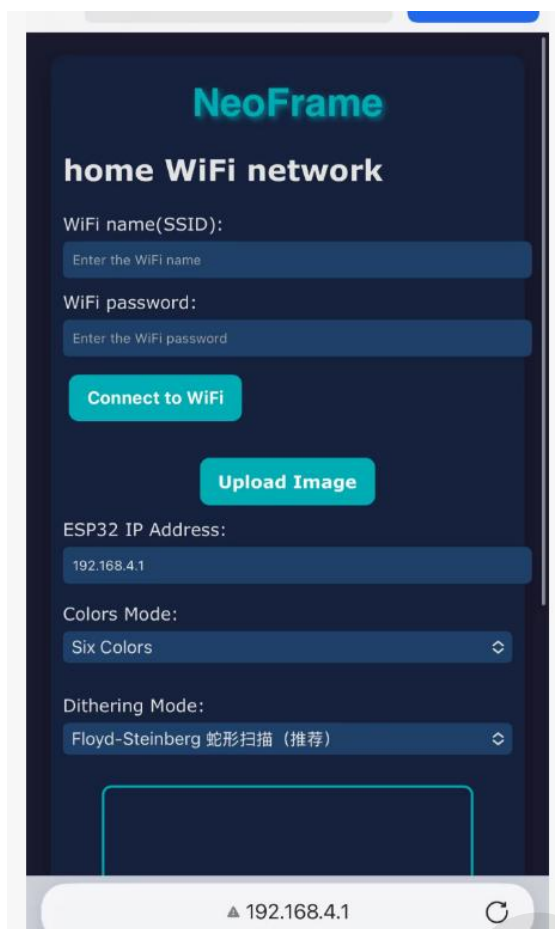
3. Click to download the BIN file of the images and copy it to the SD card.

4. Insert the SD card into the development board's card slot, turn on the power switch of the development board, and the images will be updated according to the set time.

Method 3: Update Images via WiFi Connection Between Mobile Phone and Development Board Through Web Application

Version 1 AP mode short-distance real-time update

1. The development board will create a self-built WiFi hotspot, and the mobile phone must connect to this hotspot.
2. Then scan the QR code on the image displayed on the EPD with a browser, and the mobile phone will enter the operation page of NeoFrame (actual URL: 192.168.4.1; you can also bookmark this URL in the browser for direct access when needed).



3. While powered by a USB cable or battery, the development board will create a WiFi named "NeoFrame". In the WLAN settings of the mobile phone, find the WiFi network named "NeoFrame", click to connect, and enter the password "123456789". Ensure the mobile phone is connected only to this network. Note: If the mobile phone prompts that the current WLAN has no internet access and other WLANs are available, and asks whether to switch, be sure to select "Keep Connected".

4. Upload Images

(1) Click "Upload Image" to select images taken by the mobile phone or from the mobile phone album.

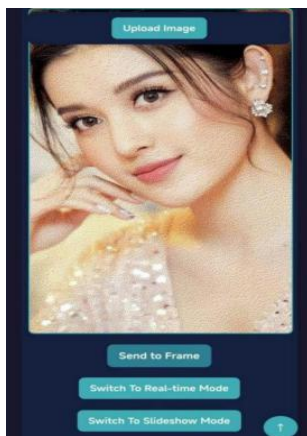
(2) Four function options are available:

① Colors Mode: Six Colors/Four Colors/Black & White / Three Colors Since the E Ink Spectra 6 screen is used, select "Six Colors".

② Dithering Mode: Floyd-Steinberg / Atkinson / Stucki / Jarvis-Judice-Ninke You can adjust the Dithering Strength and Contrast by sliding the screen

(3) After the image processing is completed, click "Send to Frame" and wait for the E-paper to refresh.

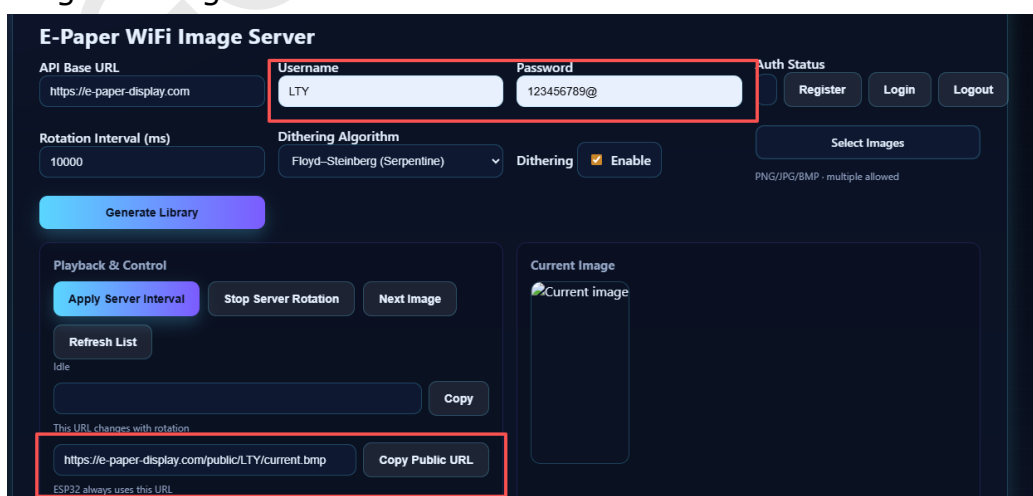
- (4) Click the "Switch To Real-time Mode" button to switch to the real-time image update mode.
- (5) Click the "Switch To Slideshow Mode" button to switch to the image carousel refresh mode (two or more images required).



Version 2

Function Introduction of This Page Via WiFi, images can be pushed with one click and refreshed remotely anytime and anywhere through mobile APP or computer without approaching the device. The operation is simple and the update is convenient, making beautiful moments present in real time.

1. Open this link: <https://e-paper-display.com/backend/image-host.html>
2. First, register an account on this page, enter the username and password, and click Register to generate a new URL.



3. After power-on, the ESP32 development board will create a self-built WiFi hotspot named "EPAPER-SETUP". After a successful connection, open a browser

- (1) Enter the WiFi name and password (note that a 2.4GHz network is required).
- (2) Enter the URL generated in the first step.
- (3) Set the network wake-up interval of the development board.
- (4) Click Save to complete the configuration.

E-Paper WiFi Config

WiFi SSID:

WiFi Password:

Image URL:

Interval (seconds):

Save

After saving, device will reboot and use new settings.

4. After the configuration is completed, open this page for image refresh operation:
 - (1) Click "Select Images" to select the images to be uploaded, then click "Generate Library" to create an image bed.
 - (2) You can select the image interval and dithering method.
 - (3) Click "Apply Server Interval" to play and wait for the images to refresh.
 - (4) Click Copy to copy the link and view the updated image content.

E-Paper WiFi Image Server

API Base URL

https://e-paper-display.com

Username

Username

Password

Password

Auth Status

Register

Login

Logout

Rotation Interval (ms)

1500

Dithering Algorithm

Floyd–Steinberg (Serpentine)

Dithering

Enable

Select Images

PNG/JPG/BMP · multiple allowed

Generate Library

Playback & Control

Apply Server Interval

Stop Server Rotation

Next Image

Refresh List

Current: 1/3

https://e-paper-display.com/uploads/user_11/46ac79b3540f4af5b91

Copy

This URL changes with rotation

https://e-paper-display.com/public/LTY/current.bmp

Copy Public URL

ESP32 always uses this URL

Current Image



5. To reset the WiFi configuration information, turn on the power and press and hold the "SW4" button for five seconds to complete the reset.

Method 4: Fixed Image Demonstration Mode (ESP32-133C02-IL-4)

After connecting GDEP133C02-1 to the development board and powering it via Type-C or battery, turn on the power switch. The default display content is as follows:

1. Black, white, red, yellow, blue and green color bars
2. A color image
3. A black, white, red, yellow, blue and green checkerboard pattern
4. White screen (no display content)
5. Corresponding firmware: [GDEP133C02-ESP-IDF Code Example GooDisplay](#)

www.good-display.com

16/23

Driver Board Series

5. Dithering Processing

When refreshing E Ink Spectra 6 series e-paper displays, RGB images must be processed with dithering to convert them into dotted patterns consisting of six colors: black, white, red, yellow, blue, and green. These processed images can then be converted into data arrays. To assist users, we have developed two software tools.

Method 1: Web-based application – [E-paper Dithering Tool](#)

1) First, open the web application and select “ E6 Color” .

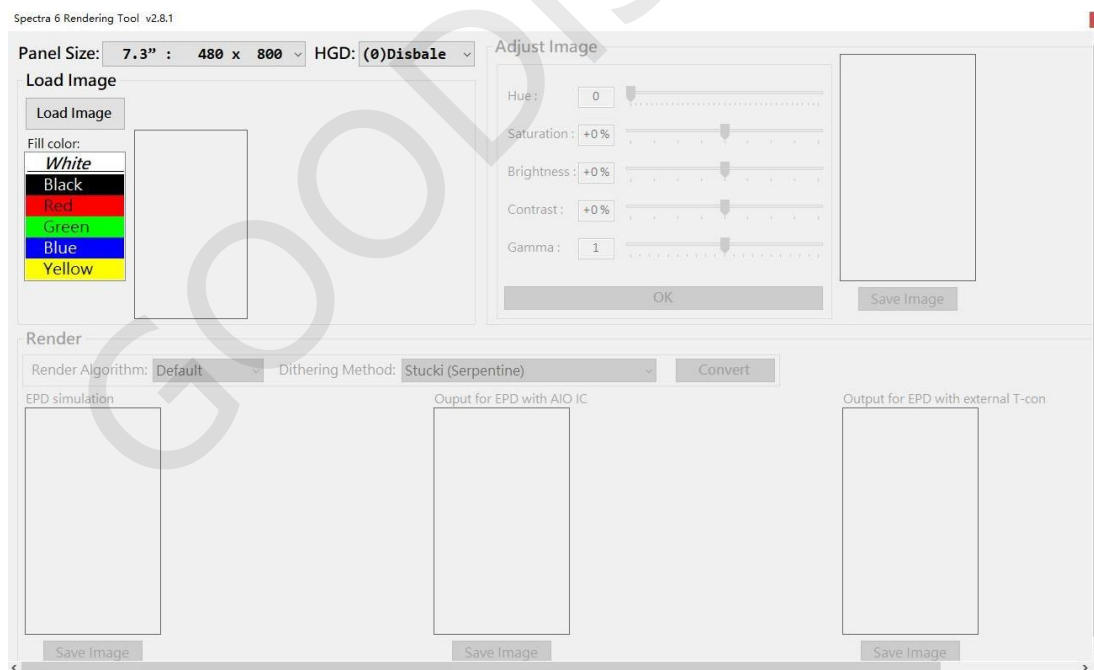


2. Next, click “ Choose File” to select the image to be processed. Make sure the selected image matches the resolution of the e-paper display. Then click “ Download Image” to complete the process.

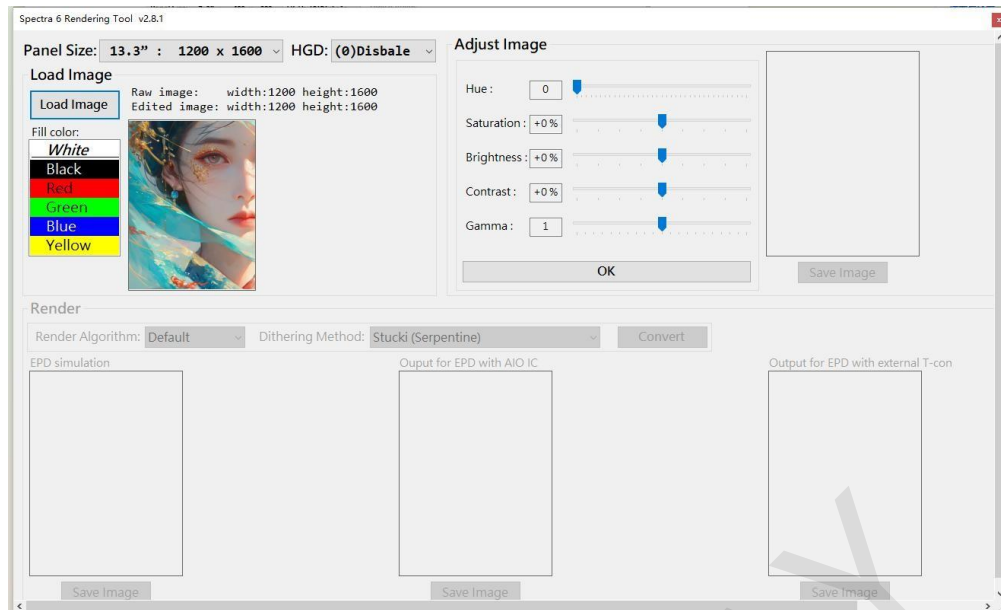


Method 2: PC-based application (to obtain the software, please contact our sales team)

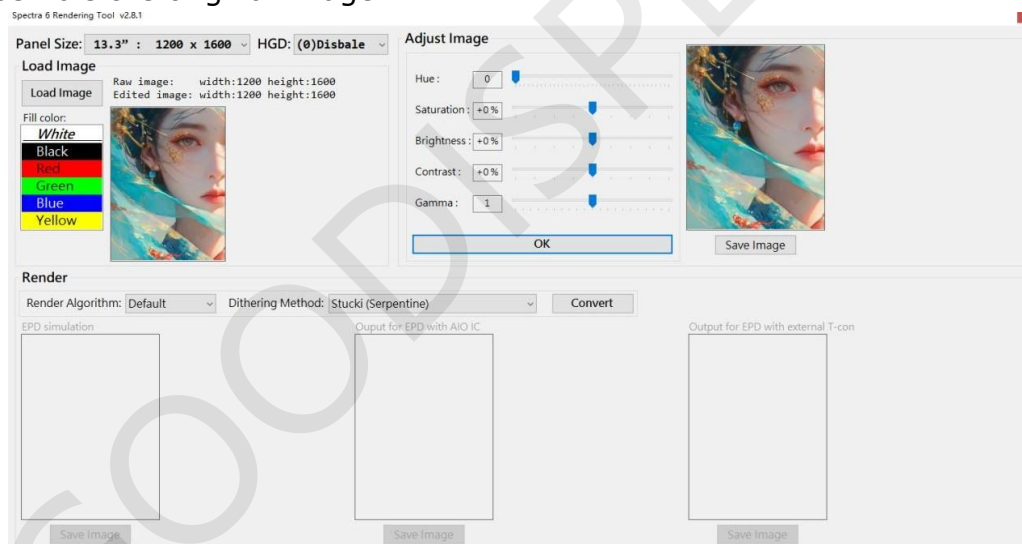
1. First, double-click " Spectra_6_Rendering_Tool_v2.8.1.exe" to open the PC software.



2. Next, select the resolution of the image to be processed by choosing the appropriate " panel size" . After selecting, click " Load Image" to add the image you want to process.



3. Click the "OK" button under the "Adjust Image" section. Each option under "Adjust Image" is important, as they are key to making the refreshed image on the e-paper closely resemble the original image.



4. Next, select the dithering algorithm under " Dithering Method" . After making your selection, click " Convert" to generate the dithered image.

" Output for EPD with AIO IC" is the processed dithered image for the e-paper display.

" EPD simulation" shows the software' s simulation of how the e-paper will display the image.

You can adjust the options under " Adjust Image" and repeat the above steps until the simulation result closely matches the original image.

5. Click " Save Image" under " Output for EPD with AIO IC" to save the processed dithered image.

6. Firmware Flashing Instructions

To support user secondary development, the ESP32-133C02-IL-

X also allows flashing new firmware. The following are the detailed steps:

1) Download and install the flashing tool

ESP-IDF 4.4 CMD

Download link: [<https://www.e-paper-display.com/esp-idf-tools-setup-offline-4.4.7.exe>]

After downloading and installing the official software, run `idf\_cmd\_init.bat`.

```
D:\Espressif\tools\xtensa-esp32s-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s-elf\bin
D:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch5-8.4.0\riscv32-esp-elf\bin
D:\Espressif\tools\esp32ulp-elf\2.35_20220830\esp32ulp-elf\bin
D:\Espressif\tools\cmake\3.23.1\bin
D:\Espressif\tools\openocd-esp32\v0.12.0-esp32-20230921\openocd-esp32\bin
D:\Espressif\tools\ninja\1.10.2\
D:\Espressif\tools\idf-exe\1.0.3\
D:\Espressif\tools\ccache\4.3\ccache-4.3-windows-64
D:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
D:\Espressif\frameworks\esp-idf-v4.4.7\tools

Checking if Python packages are up to date...
Python requirements from D:\Espressif\frameworks\esp-idf-v4.4.7\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7>
```

2) First, enter `cd /d` followed by the directory path of the program, then press Enter to navigate to the program location.

Please note that the path should not be too long and must not contain any Chinese characters, otherwise the compiler may report an error.

```
D:\Espressif\tools\xtensa-esp-elf-gdb\11.2_20220823\xtensa-esp-elf-gdb\bin
D:\Espressif\tools\riscv32-esp-elf-gdb\11.2_20220823\riscv32-esp-elf-gdb\bin
D:\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32-elf\bin
D:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s2-elf\bin
D:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch5-8.4.0\xtensa-esp32s3-elf\bin
D:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch5-8.4.0\riscv32-esp-elf\bin
D:\Espressif\tools\esp32ulp-elf\2.35_20220830\esp32ulp-elf\bin
D:\Espressif\tools\cmake\3.23.1\bin
D:\Espressif\tools\openocd-esp32\v0.12.0-esp32-20230921\openocd-esp32\bin
D:\Espressif\tools\ninja\1.10.2\
D:\Espressif\tools\idf-exe\1.0.3\
D:\Espressif\tools\ccache\4.3\ccache-4.3-windows-64
D:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
D:\Espressif\frameworks\esp-idf-v4.4.7\tools

Checking if Python packages are up to date...
Python requirements from D:\Espressif\frameworks\esp-idf-v4.4.7\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7>cd /dC:\Users\Administrator\Desktop\GDEP133C02
C:\Users\Administrator\Desktop\GDEP133C02>
```

3) Enter `idf.py build` and press Enter to compile the program.

```

ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62 - python.exe "D:\Espressif\idf_cmd_init.bat"
Go to the project directory and run:

idf.py build

D:\Espressif\frameworks\esp-idf-v4.4.7>cd /d C:\Users\Administrator\Desktop\GDEP133C02
C:\Users\Administrator\Desktop\GDEP133C02>idf.py build
Executing action: all (aliases: build)
Running cmake in directory C:\Users\Administrator\Desktop\GDEP133C02\build
Executing "cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 -DIDF_TARGET=esp32s3 -DOCACHE_ENABLE=1 C:\Users\Administrator\Desktop\GDEP133C02"...
-- Found Git: D:/Espressif/tools/idf-git/2.43.0/cmd/git.exe (found version "2.43.0.windows.1")
-- ccache will be used for faster recompilation
-- The C compiler identification is GNU 8.4.0
-- The CXX compiler identification is GNU 8.4.0
-- The ASM compiler identification is GNU
-- Found assembler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-gcc.exe
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: D:/Espressif/tools/xtensa-esp32s3-elf/esp-2021r2-patch5-8.4.0/xtensa-esp32s3-elf/bin/xtensa-esp32s3-elf-g++.exe - skipped
-- Detecting CXX compile features

```

4) Enter `idf.py flash monitor` to start flashing the program. Wait for the process to complete.

```

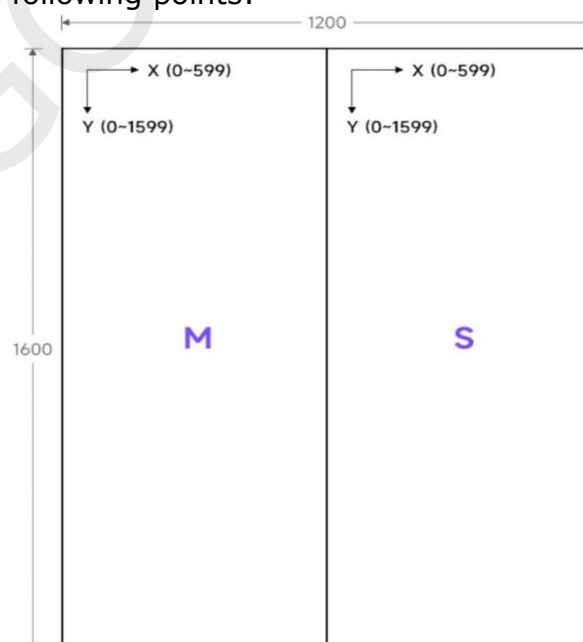
Project build complete. To flash, run this command:
D:\Espressif\python_env\idf4.4_py3.11_env\Scripts\python.exe D:\Espressif\frameworks\esp-idf-v4.4.7\components\esptool.py
esptool.py --port COM55 --baud 460800 --before default_reset --after hard_reset --chip esp32s3 write_flash --flash_mode
dio --flash_size detect --flash_freq 80m 0x0 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition_
table.bin 0x10000 build\133_ReferenceDesign_SampleCode.bin
or run 'idf.py -p (PORT) flash'

C:\Users\Administrator\Desktop\GDEP133C02>idf.py flash monitor
Executing action: flash
Serial port COM55
Connecting...
Detecting chip type... ESP32-S3
Running 'ninja flash' in directory C:\Users\Administrator\Desktop\GDEP133C02\build
Executing 'ninja flash'...
[1/5] cmd.exe /C "cd /D C:\Users\Administrator\Desktop\GDEP133C02\build\133_ReferenceDesign_SampleCode.bin"
133_ReferenceDesign_SampleCode.bin binary size 0x1234b0 bytes. Smallest app partition is 0x1f0000 bytes. 0xcdb50 bytes (
1M) free.
[2/5] Performing build step for 'bootloader'
[1/1] cmd.exe /C "cd /D C:\Users\Administrator\Desktop\GDEP133C02\build\bootloader\esp-idf\esptool.py && D:\Espressif\py
thon_env\idf4.4_py3.11_env\Scripts\python.exe D:\Espressif\frameworks\esp-idf-v4.4.7\components\partition_table\check_si
zes.py --offset 0x8000 bootloader 0x0 C:\Users\Administrator\Desktop\GDEP133C02\build\bootloader\bootloader.bin
bootloader binary size 0x52b0 bytes. 0x2d50 bytes (35%) free.
[2/3] cmd.exe /C "cd /D D:\Espressif\frameworks\esp-idf-v4.4.7\components\esptool.py/run_serial_tool.cmake"
esptool.py esp32s3 -p COM55 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_fre
q 80m --flash_size 16MB 0x0 bootloader\bootloader.bin 0x10000 133_ReferenceDesign_SampleCode.bin 0x8000 partition_table/
partition_table.bin
esptool.py v3.3.4-dev
Serial port COM55
Connecting...

```

5) Dual IC Programming Explanation

The GDEP133C02 is an e-paper display controlled by dual ICs. Please pay attention to the following points:

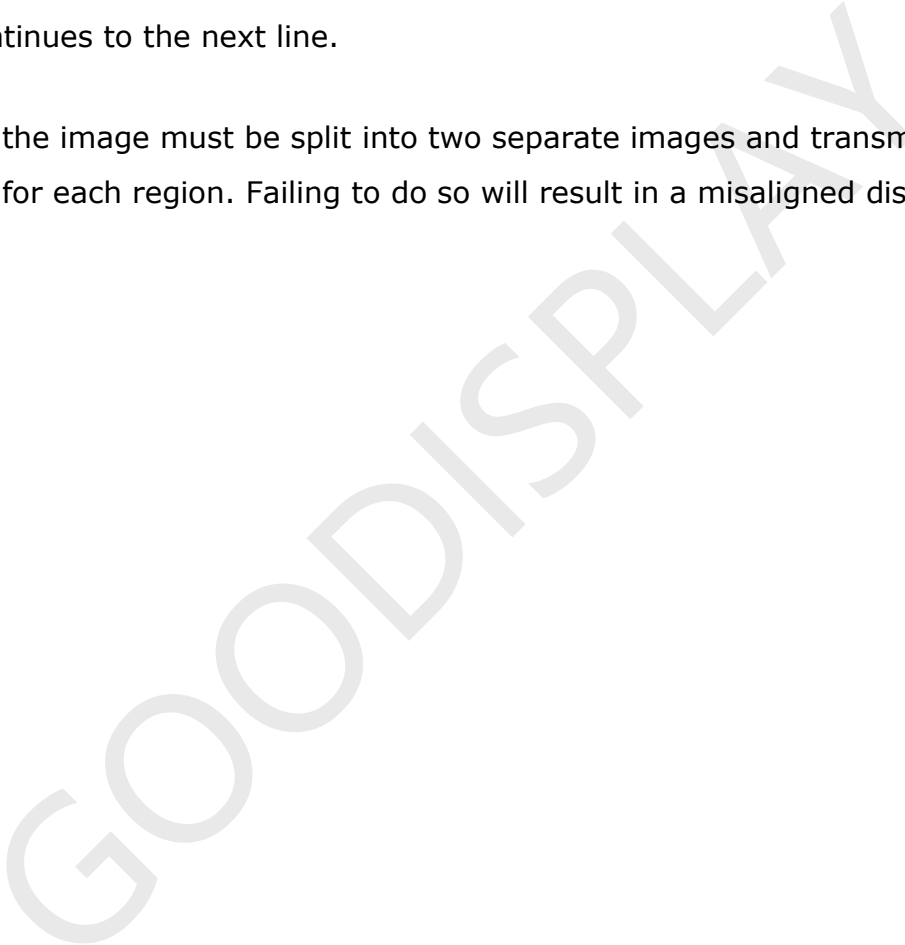


As shown in the figure, each IC controls half of the screen. Two CS (chip select) pins are used to control the corresponding areas: M (main) and S (sub). Except for the CS pins, all other control pins are shared.

In addition to shared commands (such as initialization and refresh commands used in example programs), only one CS pin is allowed to be pulled low at a time.

When transmitting image data, please note that the screen displays content in landscape orientation (along the X-axis). After reaching the end of one area, the display continues to the next line.

Therefore, the image must be split into two separate images and transmitted separately for each region. Failing to do so will result in a misaligned display.



7.Precautions

Do not keep the screen powered on for a long time. When the screen is not refreshing, set it to sleep mode or power it off. Otherwise, the screen will remain in a high-voltage state for a long time, which will damage the diaphragm and cannot be repaired.

When using the E-ink screen, it is recommended that the refresh interval be at least 180s, and the screen be refreshed at least once every 24 hours.

If the E-ink screen is not used for a long time, refresh it to a white screen for storage (refer to the data sheet for specific storage environment requirements).

After the screen enters sleep mode, it will ignore the sent image data, and only re-initialization can make it refresh normally.

If the created image data is displayed incorrectly on the screen, it is recommended to check whether the image size setting is correct, and try swapping the width and height settings.

The working voltage of the E-ink screen is 3.3V. If you purchase a bare screen and need to match a 5V working environment when designing the circuit, it is recommended to perform level conversion processing.

The FPC cable of the screen is relatively fragile. Please note: Do not bend the cable perpendicular to the screen to avoid tearing; do not bend the cable repeatedly and excessively to avoid breakage; do not bend the cable toward the front of the screen to avoid disconnection between the cable and the panel.

It is recommended to fix the cable during debugging and development.

The E-ink screen is fragile, so avoid dropping, collision and hard pressing as much as possible.

We recommend that customers test the screen with the provided sample programs and the corresponding development board after receiving it.

Friendly Reminder:

The official website only provides the most basic image refresh programs. Functions such as SD card, WiFi and USB communication need to be developed by yourself or contact our sales for relevant technical support.