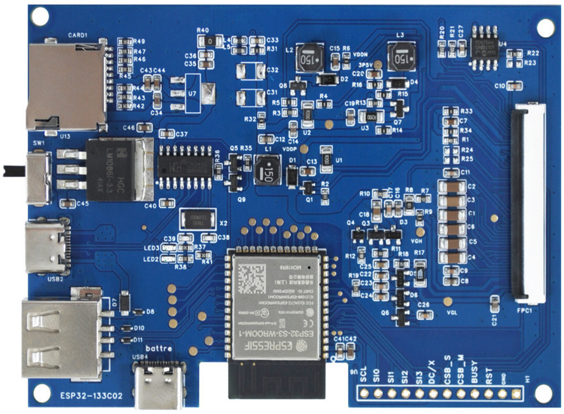
E-paper Display Driver Board

GooDisplay

ESP32-133C02-X

Dalian Good Display Co., Ltd.

Product Specifications



Customer	Standard
Description	E-paper display driver board
Model Name	ESP32-133C02-X
Date	2025/06/12
Revision	1.0

	Design Engineering		
	Approval	Check	Design
	刘玉宝印	修凤燕印	吴良之印

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

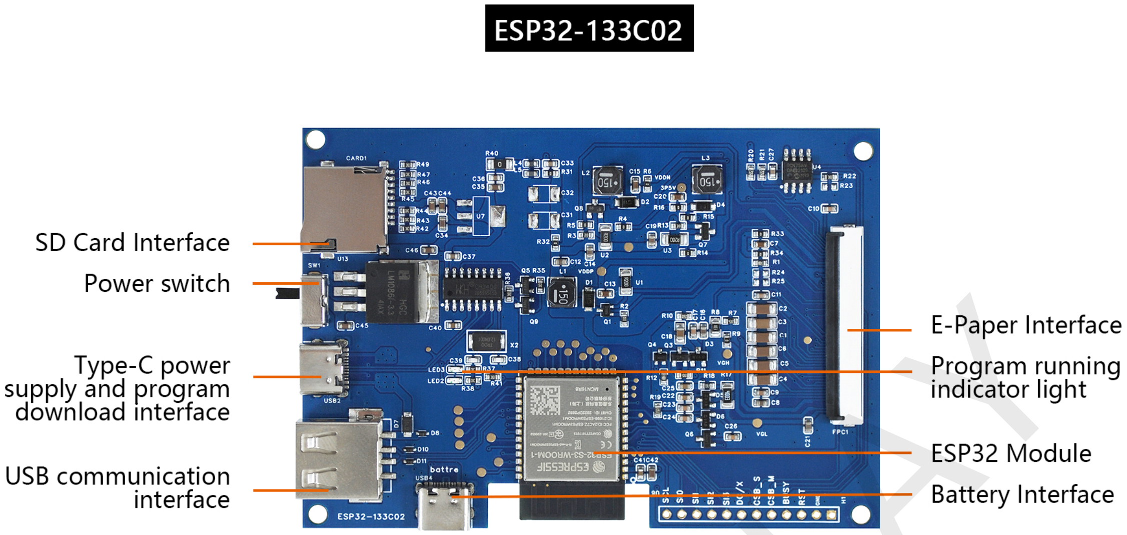
The ESP32-133C02-X development board is specifically designed to drive the 13.3-inch color E Ink Spectra 6 display, model GDEP133C02. It supports image demonstration, image updates via PC software, wireless image transmission through Wi-Fi, and various other functions.

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-X
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

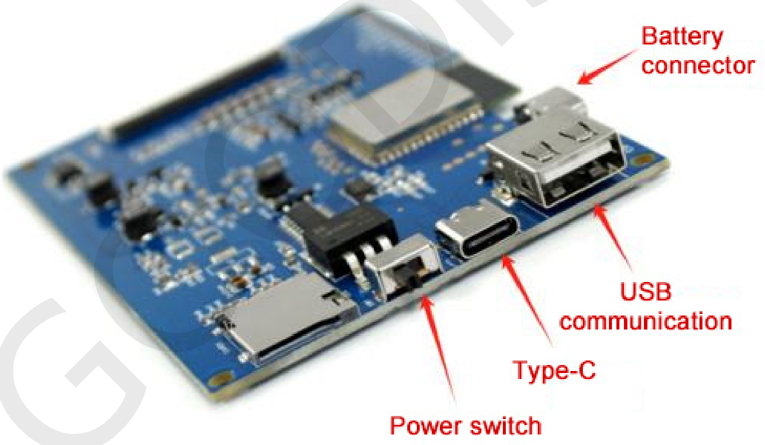

ESP32-133C02
SD Card Interface
Power switch
Type-C power supply and program download interface
USB communication interface
E-Paper Interface
Program running indicator light
ESP32 Module
Battery Interface

4. Program Download

For the ESP32-133C02-X development board, four functional versions have been developed:
1. ESP32-133C02-1: Updates images via USB communication between the PC and the development board.
2. ESP32-133C02-2: Updates images by parsing image files stored on an SD card.
3. ESP32-133C02-3: Updates images through a web-based application after connecting the mobile device to the development board via Wi-Fi.
4. ESP32-133C02-4: Cycles through and displays fixed image content for display testing and demonstration purposes.

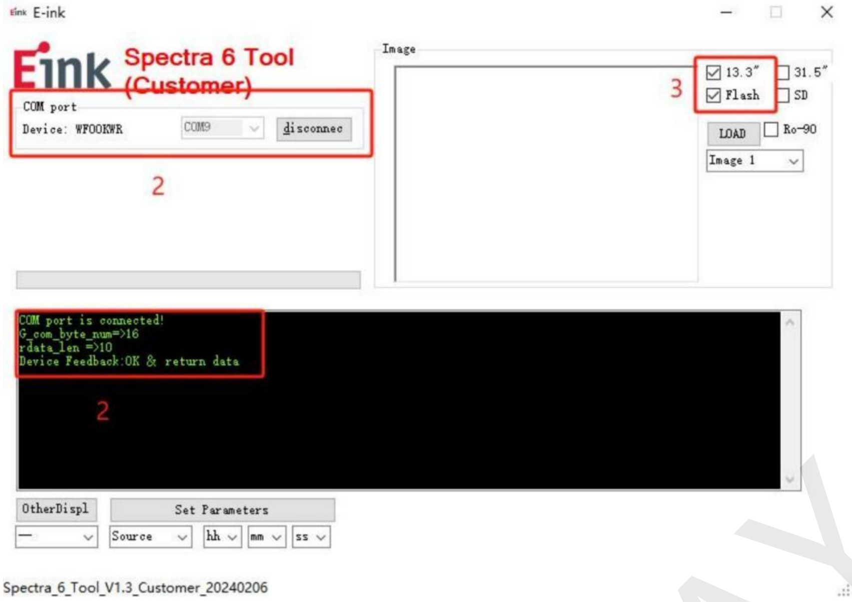
Note: Each version requires flashing a different firmware. Please select the correct model based on the required function when making a purchase. Each model supports only one function.

ESP32-133C02-1: Updating images via USB communication between PC and development board
1. First, connect the microcontroller development board to the EPD and power it via Type-C or battery. Then, use a dual male USB data cable to connect the development board to the PC.
Turn on the power switch.



2. Install and open Spectra 6_Image_Tool.exe (available for download from *our official website*). After successful installation, launch 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, please check the Device Manager to ensure the COM port is properly connected.
(3) Once connected, select "13.3" and click "Flash".



COM port
Device: WFOOKWR COM9 disconnec

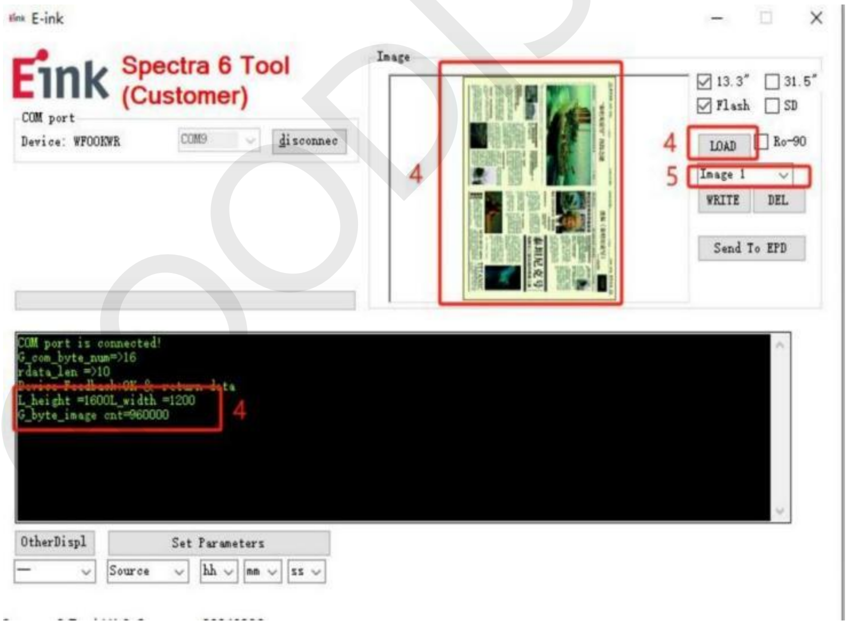
☑ 13.3" □ 31.5"
☑ Flash □ SD
□ Ro-90

```
COM port is connected!
G_com_byte_num=>16
rdata_len =>10
Device Feedback:OK & return data
```

Spectra_6_Tool_V1.3_Customer_20240206

(4) Click "LOAD" and select an image from the folder (resolution: 1200x1600, and the image should be pre-processed with dithering).

(5) Select the image storage location (you can choose from Image1 to Image15).



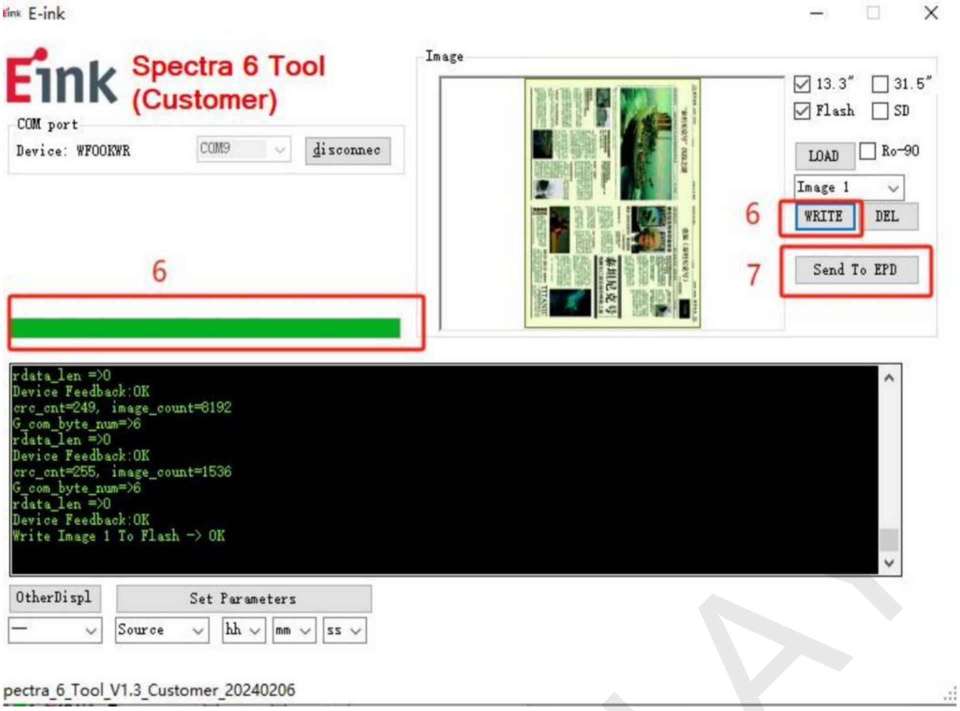
COM port
Device: WFOOKWR COM9 disconnec

☑ 13.3" □ 31.5"
☑ Flash □ SD
□ Ro-90

```
COM port is connected!
G_com_byte_num=>16
rdata_len =>10
Device Feedback:OK & return data
L_height =1600L_width =1200
G_byte_image cnt=960000
```

(6) Click "WRITE" to burn the image data into the microcontroller. Wait for the progress bar to complete.

(7) After the burning is completed, click "Send To EPD". The EPD will display the image stored at the selected Imagexx location, and the e-paper will begin refreshing.



Device: WFOOKWR COM9

☑ 13.3" □ 31.5"
☑ Flash □ SD
□ Ro-90
Image 1

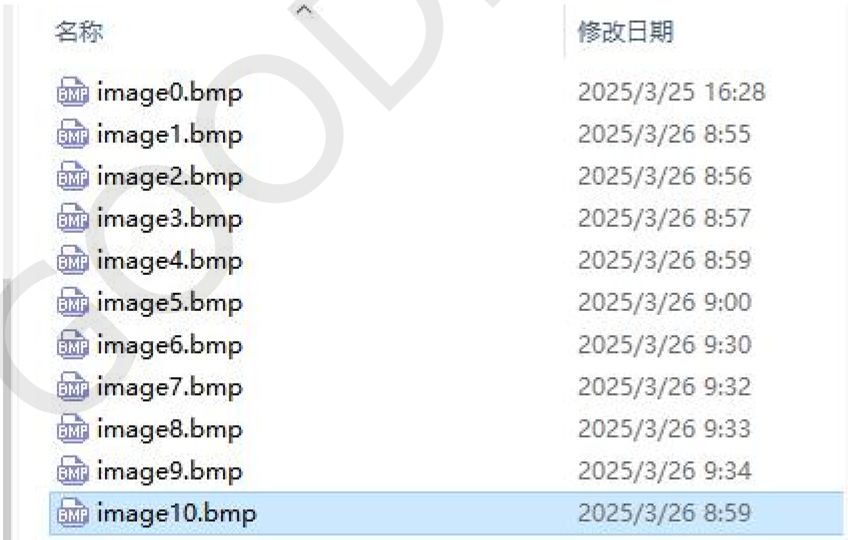
```
rdata_len =>0
Device Feedback:OK
crc_cnt=249, image_count=8192
G_com_byte_num=>6
rdata_len =>0
Device Feedback:OK
crc_cnt=255, image_count=1536
G_com_byte_num=>6
rdata_len =>0
Device Feedback:OK
Write Image 1 To Flash -> OK
```

— Source hh mm ss

pectra_6_Tool_V1.3_Customer_20240206

ESP32-133C02-2: Updates images by parsing the pictures stored on the SD card.

1. First, insert the SD card into a card reader and connect it to the computer.
2. Copy the pre-dithered images to the SD card using file names such as image0.bmp, image1.bmp, etc.



名称	修改日期
image0.bmp	2025/3/25 16:28
image1.bmp	2025/3/26 8:55
image2.bmp	2025/3/26 8:56
image3.bmp	2025/3/26 8:57
image4.bmp	2025/3/26 8:59
image5.bmp	2025/3/26 9:00
image6.bmp	2025/3/26 9:30
image7.bmp	2025/3/26 9:32
image8.bmp	2025/3/26 9:33
image9.bmp	2025/3/26 9:34
image10.bmp	2025/3/26 8:59

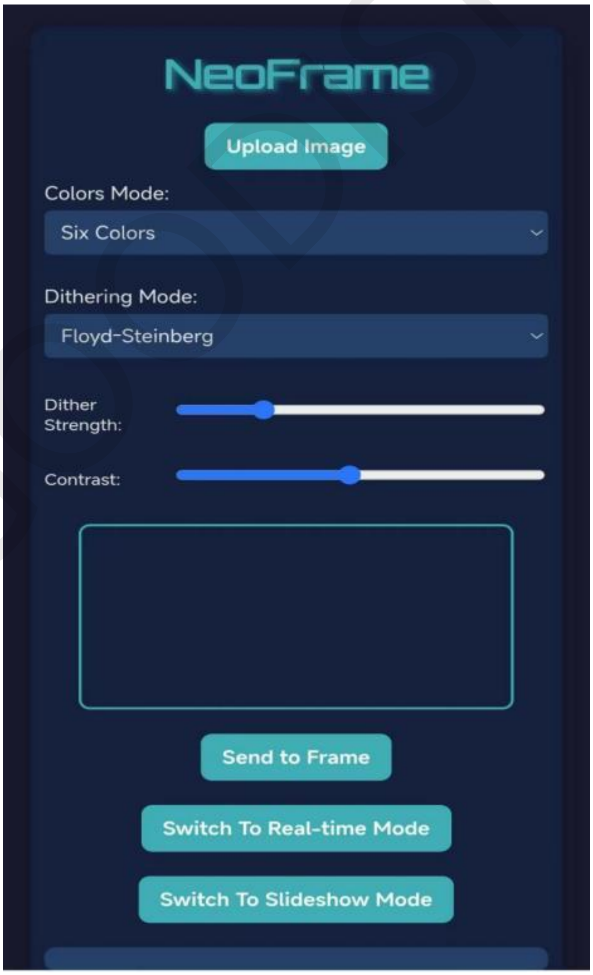
3. Insert the SD card into the card slot on the development board, turn on the power switch, and the display will update approximately once every 2 minutes.

Note: As users may have different capacity requirements, to avoid unnecessary costs, the SD card is not included and should be provided by the user.

ESP32-133C02-3: Update images via Wi-Fi connection between mobile device and development board using a web-based application

1. While the mobile device is connected to the internet, scan the QR code below using browser.

2. The mobile device will enter the NeoFrame operation page, as shown below.
The actual URL is [*http://www.einkapp.com/esp32-133c02.html*]. You can also bookmark this page in
your browser for quick access when needed.



Colors Mode: Six Colors

Dithering Mode: Floyd-Steinberg

3. While powered via USB cable or battery, the development board will create a Wi-Fi network named “NeoFrame”. On your mobile device, go to WLAN settings, find the “NeoFrame” network, and connect to it using the password “123456789”. Make sure your phone is only connected to this network.

Note: If your phone prompts, “This WLAN has no internet access. A different WLAN is available. Switch?”, please be sure to select “Stay connected”.

4. Uploading an image

(1) Tap “Upload Image” to choose a photo from your camera or your photo gallery.

(2) There are two function options:

① Colors Mode:

Six Colors / Four Colors / Black & White / Three Colors

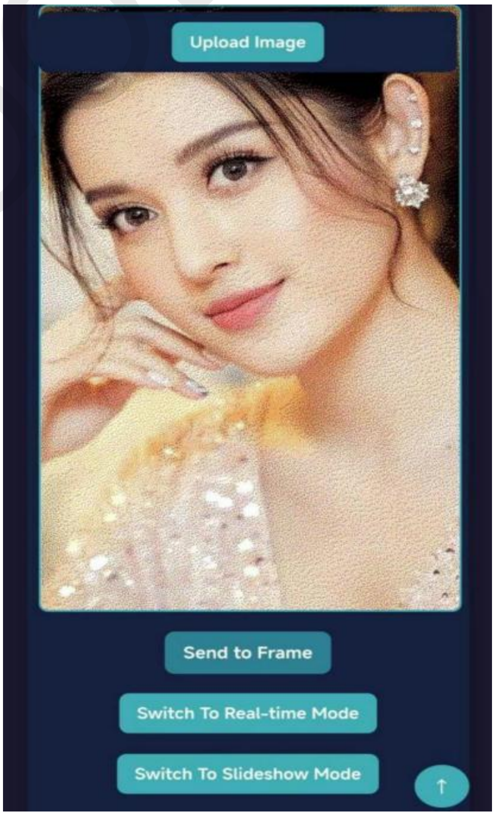
Since the display used is an E Ink Spectra 6 screen, select “Six Colors”.

② Dithering Mode:

Floyd-Steinberg / Atkinson / Stucki / Jarvis-Judice-Ninke

You can adjust the Dithering Strength and Contrast by sliding the controls on the screen.

(3) Once the image processing is complete, tap “Send to Frame”. Wait for the e-paper display to refresh.



ESP32-133C02-4: The display sequentially shows fixed image content, used for screen testing and demonstration.

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

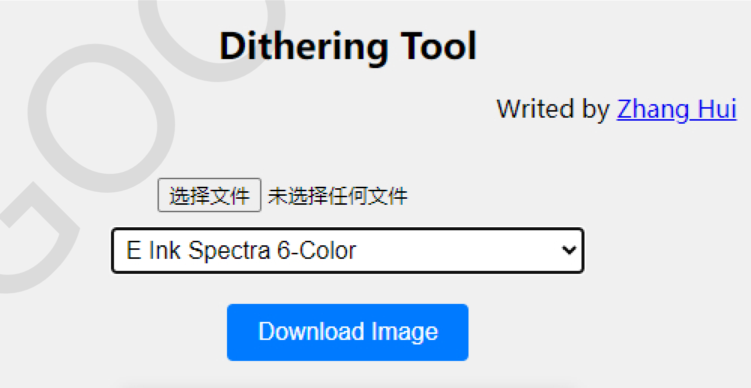
1. Black, white, red, yellow, blue, and green color bars
2. A full-color image
3. A checkerboard pattern in black, white, red, yellow, blue, and green
4. A white screen (blank display)

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*

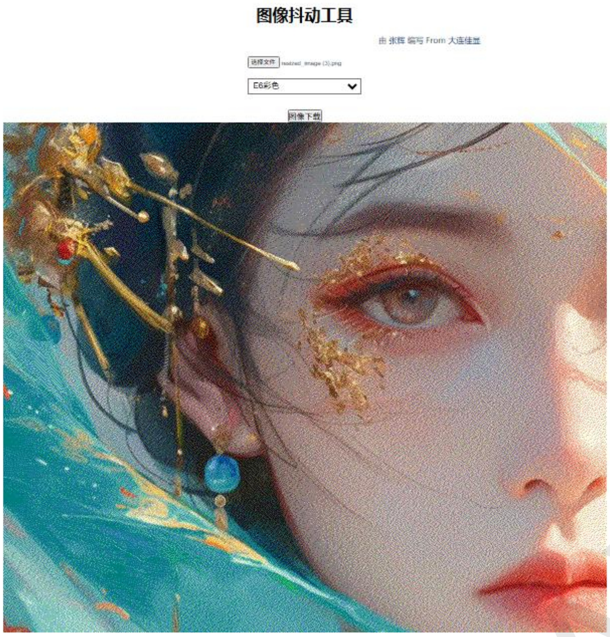
1) First, open the web application and select " E6 Color " .



Dithering Tool

Writed by Zhang Hui

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.

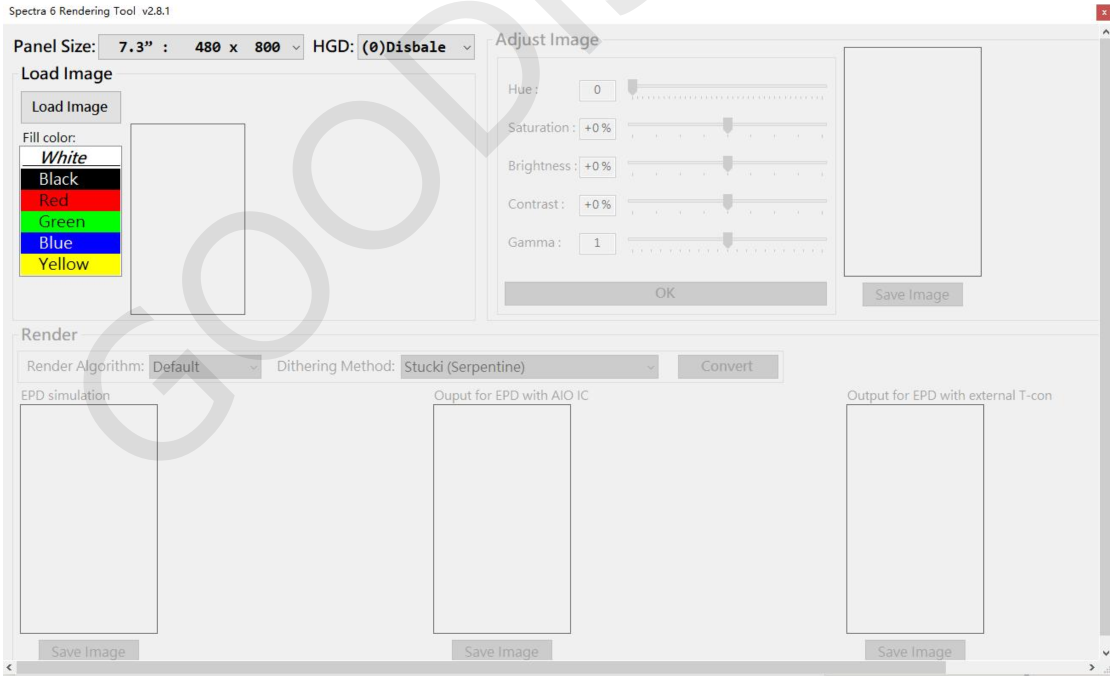


图像抖动工具

由 张辉 编写 From 大连佳显

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.



Spectra 6 Rendering Tool v2.8.1

Panel Size: 7.3” : 480 x 800

HGD: (0)Disbale

Hue : 0

Saturation : +0%

Brightness : +0%

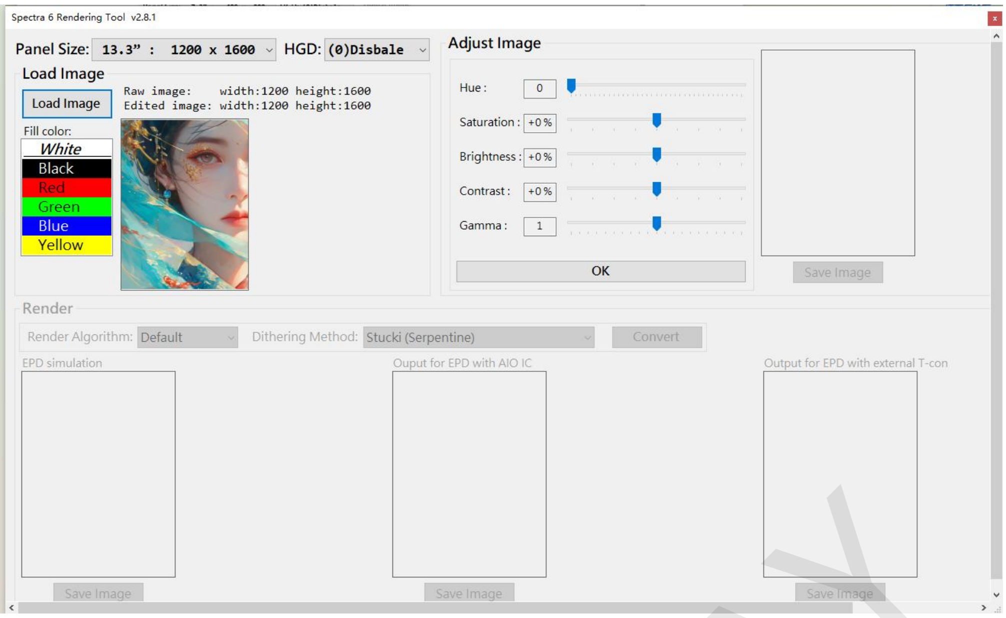
Contrast : +0%

Gamma : 1

Render Algorithm: Default

Dithering Method: Stucki (Serpentine)

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.



Spectra 6 Rendering Tool v2.8.1

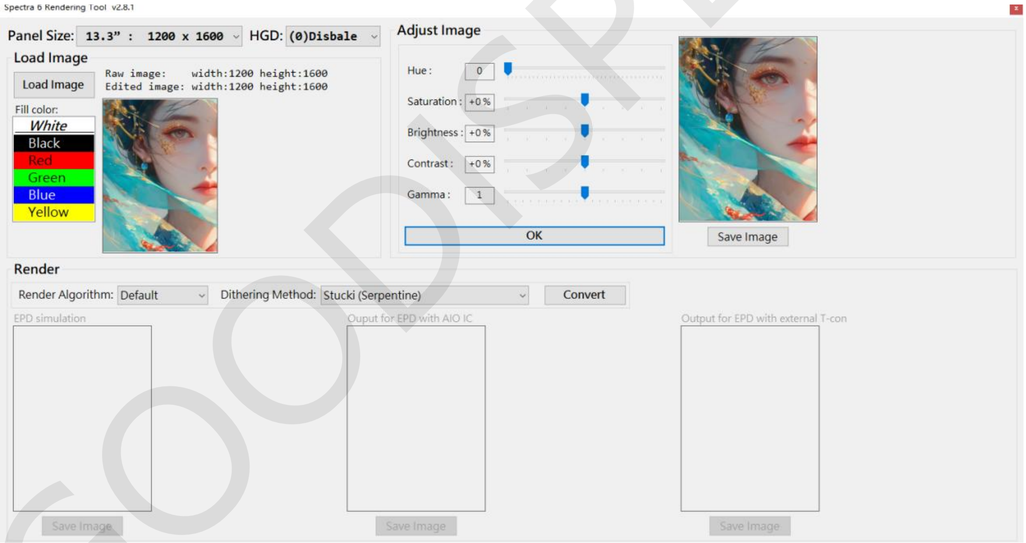
Panel Size: 13.3" : 1200 x 1600 HGD: (0)Disbale

```
Raw image:    width:1200 height:1600
Edited image: width:1200 height:1600
```

Hue: 0
Saturation: +0%
Brightness: +0%
Contrast: +0%
Gamma: 1

Render Algorithm: Default Dithering Method: Stucki (Serpentine)

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.



Spectra 6 Rendering Tool v2.8.1

Panel Size: 13.3" : 1200 x 1600 HGD: (0)Disbale

```
Raw image:    width:1200 height:1600
Edited image: width:1200 height:1600
```

Hue: 0
Saturation: +0%
Brightness: +0%
Contrast: +0%
Gamma: 1

Render Algorithm: Default Dithering Method: Stucki (Serpentine)

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

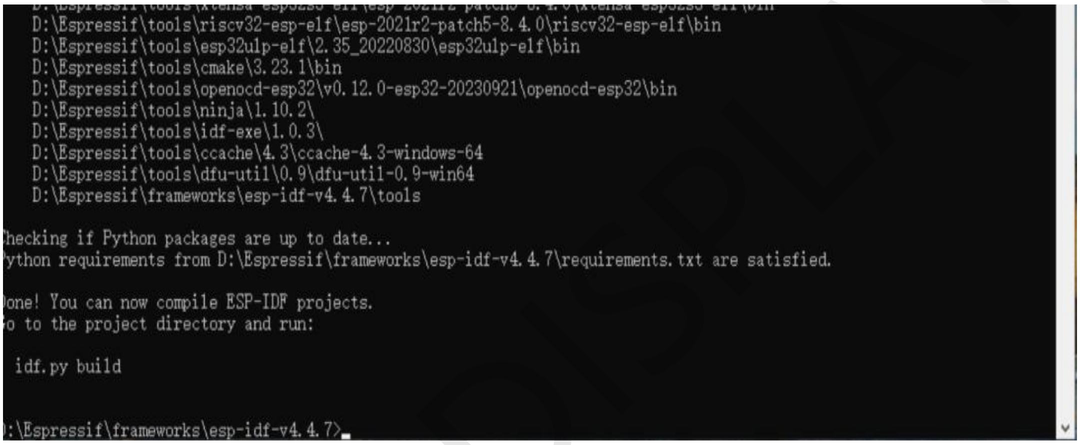
To support user secondary development, the ESP32-133C02-X also allows flashing new firmware. The following are the detailed steps:

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*]

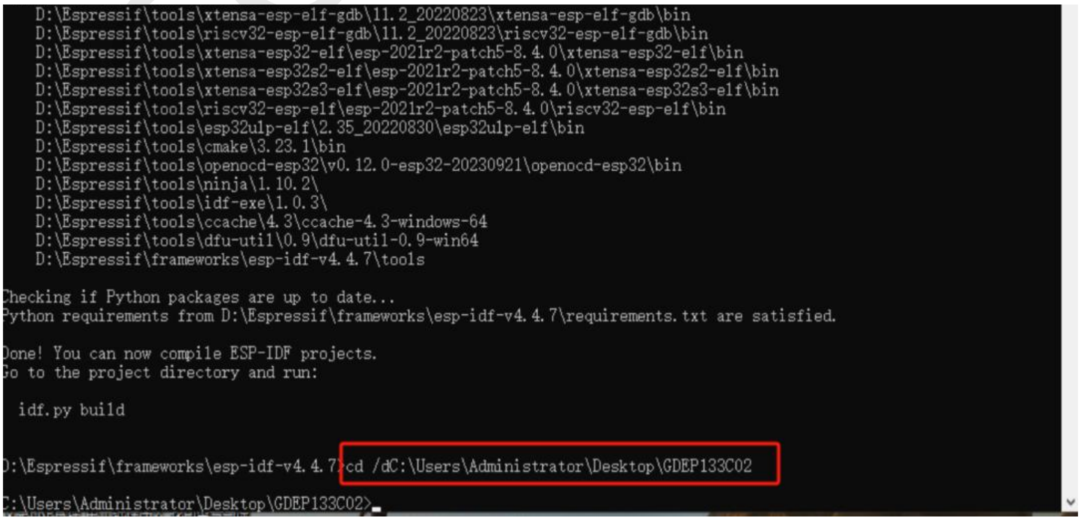
After downloading and installing the official software, run `idf_cmd_init.bat`.



```
   D:\Espressif\tools\xtensa-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

hecking if Python packages are up to date...
ython requirements from D:\Espressif\frameworks\esp-idf-v4. 4. 7\requirements. txt are satisfied.

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

 idf.py build

:\Espressif\frameworks\esp-idf-v4. 4. 7>_
```

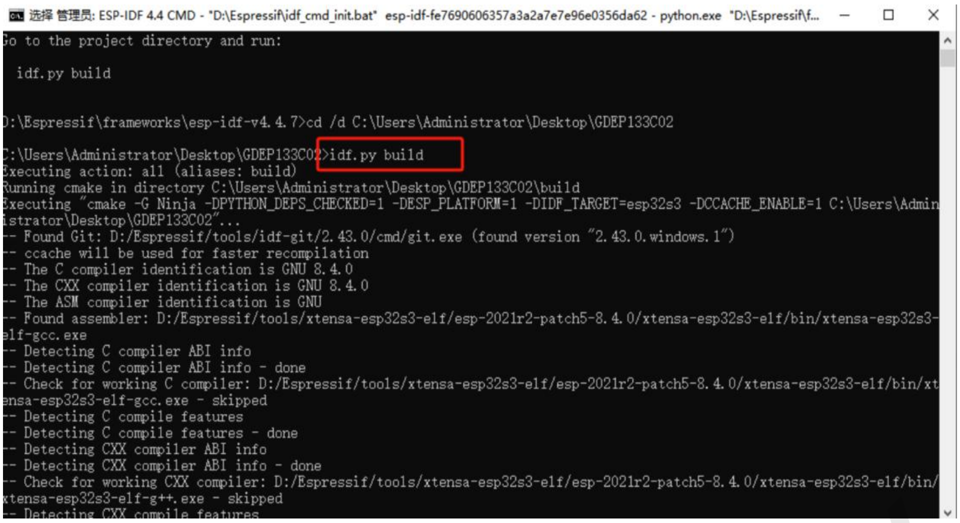
2) First, enter `cd /d` followed by the directory path of the program, then press Enter to navigate to the program location.

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

hecking if Python packages are up to date...
ython requirements from D:\Espressif\frameworks\esp-idf-v4. 4. 7\requirements. txt are satisfied.

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

 idf.py build

:\Espressif\frameworks\esp-idf-v4. 4. 7>cd /dC:\Users\Administrator\Desktop\GDEP133C02

:\Users\Administrator\Desktop\GDEP133C02>_
```

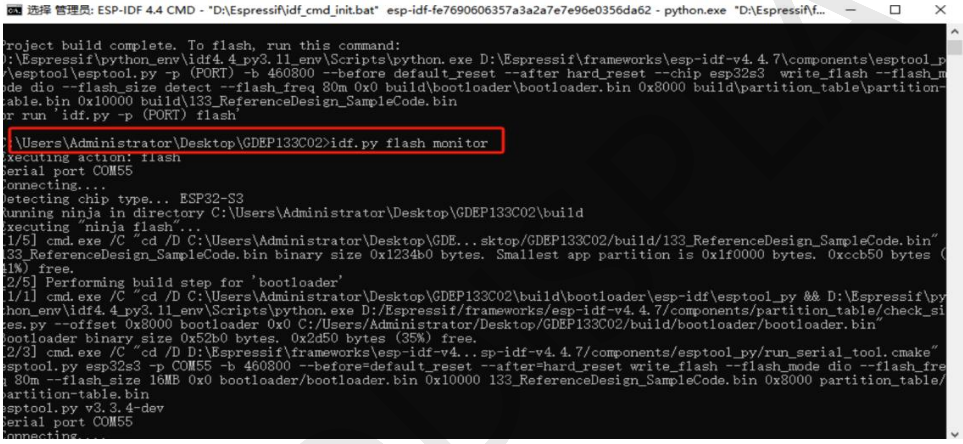
3) Enter `idf.py build` and press Enter to compile the program.



选择 管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62 - python.exe "D:\Espressif\f...

```
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 -DCCACHE_ENABLE=1 C:\Users\Admin
istrator\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/xt
ensa-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.

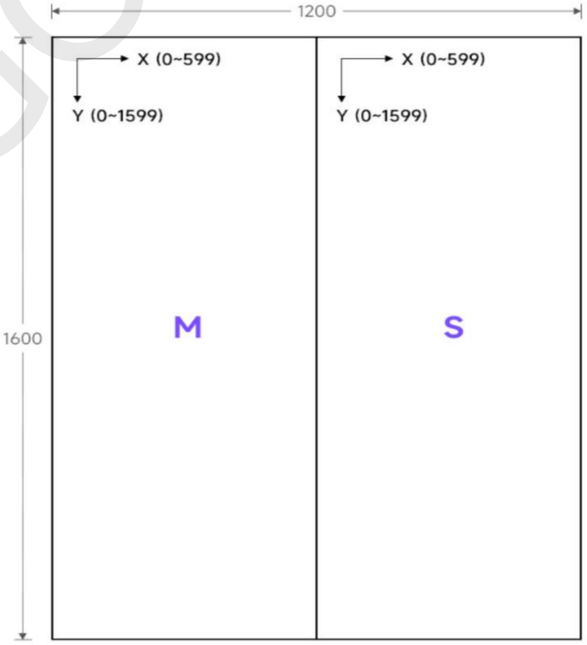


选择 管理员: ESP-IDF 4.4 CMD - "D:\Espressif\idf_cmd_init.bat" esp-idf-fe7690606357a3a2a7e7e96e0356da62 - python.exe "D:\Espressif\f...

```
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_p
y\esptool\esptool.py -p (PORT) -b 460800 --before default_reset --after hard_reset --chip esp32s3  write_flash --flash_m
ode 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 in directory C:\Users\Administrator\Desktop\GDEP133C02\build
Executing "ninja flash"...
[1/5] cmd.exe /C "cd /D C:\Users\Administrator\Desktop\GDE...sktop/GDEP133C02/build/133_ReferenceDesign_SampleCode.bin"
133_ReferenceDesign_SampleCode.bin binary size 0x1234b0 bytes. Smallest app partition is 0x1f0000 bytes. 0xccb50 bytes (
41%) 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...sp-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

1) Do not leave the screen powered on for a long time without refreshing. When the screen is not being refreshed, it should be set to sleep mode or powered off. Prolonged exposure to high voltage can damage the film layer of the screen and may cause irreversible failure.

2) When using the e-paper display, it is recommended to have a minimum refresh interval of 180 seconds, and to perform at least one full refresh every 24 hours.

3) If the e-paper display will not be used for an extended period, it should be refreshed to a white screen before storage. (For detailed storage requirements, refer to the datasheet.)

4) Once the screen enters sleep mode, it will ignore any incoming image data. Reinitialization is required to resume normal operation.

5) If the image data appears incorrectly on the screen, check whether the image resolution is set properly. Try switching the width and height settings and test again.

6) The operating voltage of the e-paper display is 3.3V. If you are using a bare panel, and your circuit is designed to work in a 5V environment, it is recommended to implement level shifting to ensure proper operation.

7) The FPC (flexible printed circuit) of the screen is fragile. Please note:

– Do not bend the FPC vertically relative to the screen to avoid tearing.

– Do not bend the FPC repeatedly or excessively to avoid breakage.

– Do not bend the FPC toward the front side of the screen to prevent disconnection between the cable and the panel.

8) During development and debugging, it is recommended to fix the FPC in place before use. The e-paper screen is delicate—avoid dropping, bumping, or applying excessive pressure to prevent damage.