# 2.66inch e-Paper Module (B) Manual

## Parameters

| | |
|---|---|
| Dimensions | 2.66inch |
| Driver board dimensions | 83.5mm × 37mm |
| Display dimensions | 60.088mm × 30.704mm |
| Outline dimensions (screen only) | 71.820mm × 36.304mm × 1.0mm |
| Operating voltage | 3.3V / 5V (5V is required for power and signal) |
| Communication interface | SPI |
| Dot pitch | 0.202mm × 0.203mm |
| Resolution | 296 × 152 |
| Display color | Black, White, Red |
| Frey scale | 2 |
| Refresh time | 15s |
| Refresh power | 26.4mW（typ.） |
| Standby current | < 0.01uA (almost 0) |

- Refresh time: The refresh time is the experimental test data, the actual refresh time will have errors, and the actual effect shall prevail. There will be a flickering effect during the global refresh process, this is a normal phenomenon.

- Power consumption: The power consumption data is the experimental test data. The actual power consumption will have a certain error due to the existence of the driver board and the actual use situation. The actual effect shall prevail.

## SPI Communication Timing
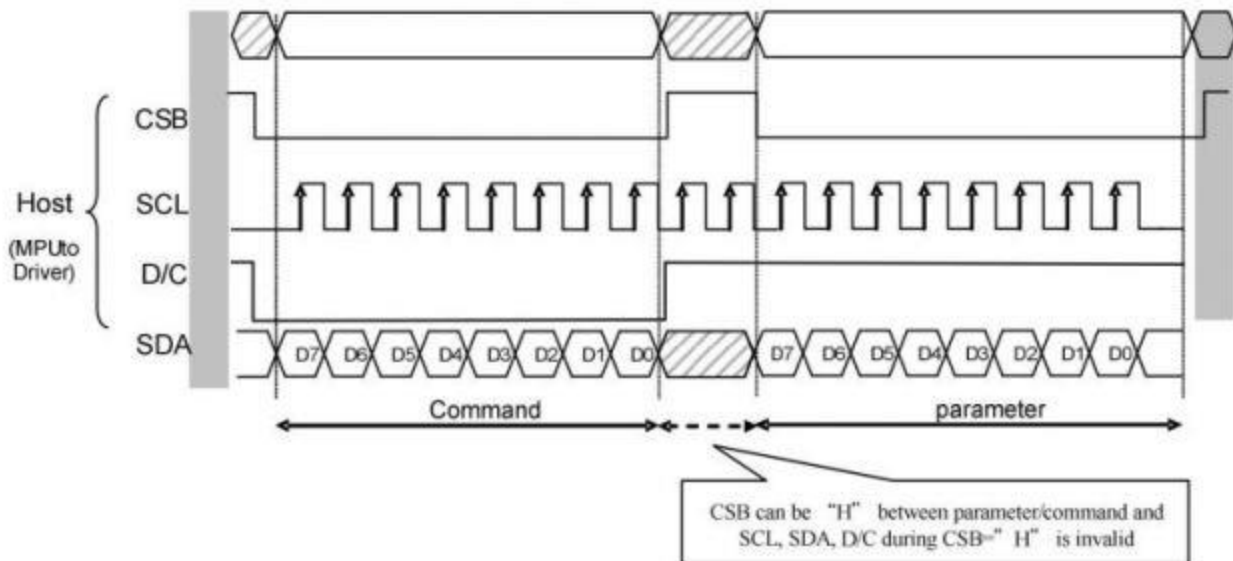
- CSB (CS): Slave chip select, when CS is low, the chip is enabled.
- SCL (SCK/SCLK): UART clock signal.
- D/C (DC): data/command control pin, writes commands in low level; writes data/parameter in high level.
- SDA (DIN): serial data signal.
- Timing: CPHL=0, CPOL=0 (SPI0)

【Remarks】For more information about SPI, you can search for information online.

## Working Principle

This product is an E-paper device adopting the image display technology of Microencapsulated Electrophoretic Display, MED. The initial approach is to create tiny spheres, in which the charged color pigments are suspended in the transparent oil and would move depending on the electronic charge. The E-paper screen display patterns by reflecting the ambient light, so it has no background light requirement. Under ambient light, the E-paper screen still has high visibility with a wide viewing angle of 180 degrees. It is the ideal choice for E-reading.

## Programming Principle

We define the pixels in a monochrome picture, 0 is black and 1 is white.
White：□: Bit 1
Black：■：Bit 0

- The dot in the figure is called a pixel. As we know, 1 and 0 are used to define the color, therefore we can use one bit to define the color of one pixel, and 1 byte = 8pixels
- For example, If we set the first 8 pixels to black and the last 8 pixels to white, we show it by codes, they will be 16-bit as below:

| Pixel | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Bit | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Color | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ |

For the computer, the data is saved in MSB format:

| Pixel | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Index | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Color | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | □ | □ | □ | □ | □ | □ | □ | □ |
| Byte | 0x00 | | | | | | | | 0xFF | | | | | | | |

So we can use two bytes for 16 pixels.

For e-paper B, the display colors are red, black, and white. We need to split the picture into 2 pictures, one is a black and white picture, and another is a red and white picture. When transmitting, because one register controls a black or white pixel, one controls a Red or white display. The black and white part of display uses 1 byte to control 8 pixels, and the red and white part use 1 byte to control 8 pixels.

For example, suppose there are 8 pixels, the first 4 are red, and the back 4 are black:
They need to be disassembled into a black-and-white picture and a red-and-white picture. Both pictures have 8 pixels, but the first four pixels of the black and white picture are white, the last 4 pixels are black, and the first 4 pixels of the red and white picture One pixel is red, and the last four pixels are white.

| Original picture | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
|---|---|---|---|---|---|---|---|---|
| Divided | | | | | | | | |
| Black/White | □ | □ | □ | □ | ■ | ■ | ■ | ■ |
| Red/White | ■ | ■ | ■ | ■ | □ | □ | □ | □ |

If you define that the data of white pixel is 1 and the black is 0, then we can get:

| Black/White | □ | □ | □ | □ | ■ | ■ | ■ | ■ |
|---|---|---|---|---|---|---|---|---|
| Data | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Red/White | ■ | ■ | ■ | ■ | □ | □ | □ | □ |
| Data | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

So that we can use 1 byte to control every eight pixels.

| Bit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Black/White | ☐ | ☐ | ☐ | ☐ | ■ | ■ | ■ | ■ |
| Data | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| Byte | 0xF0 | | | | | | | |

| Bit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Red/White | ■ | ■ | ■ | ■ | ☐ | ☐ | ☐ | ☐ |
| Data | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Byte | 0x0F | | | | | | | |

## Precautions

1. For E-paper displays that support partial refresh, please note that you cannot refresh them with the partial refresh mode all the time. After refreshing partially several times, you need to fully refresh EPD once. Otherwise, the display effect will be abnormal, which cannot be repaired!

2. It is a normal phenomenon that the three-color EPD will have a certain color difference in different batches. Hence, It is recommended to use the program to clear all the pictures on the EPD and store it facing up. Please clear the screen several times before powering on.

3. Note that the screen cannot be powered on for a long time. When the screen is not refreshed, please set the screen to sleep mode or power off it. Otherwise, the screen will remain in a high voltage state for a long time, which will damage the e-Paper and cannot be repaired!

4. When using the e-Paper display, it is recommended that the refresh interval be at least 180s, and refresh at least once every 24 hours. If the e-Paper is not used for a long time, you should use the program to clear the screen before storing it. (Refer to the datasheet for specific storage environment requirements.)

5. After the screen enters sleep mode, the sent image data will be ignored, and it can be refreshed normally only after initializing again.

6. Control the 0x3C or 0x50 (refer to the datasheet for details) register to adjust the border color. In the demo, you can adjust the Border Waveform Control register or VCOM AND DATA INTERVAL SETTING to set the border.

7. If you find that the created image data is displayed incorrectly on the screen, it is recommended to check whether the image size setting is correct, change the width and height settings of the image and try again.

8. The working voltage of the e-Paper display is 3.3V. If you buy the raw panel and you need to add a level convert circuit for compatibility with 5V voltage. The new version of the driver board (V2.1 and subsequent versions) has added a level processing circuit, which can support both 3.3V and 5V. The old version only supports a 3.3V working environment. You can confirm the version before using it. (The one with the 20-pin chip on the PCB is generally the new version.)

9. The FPC cable of the screen is relatively fragile, pay attention to bending the cable along the horizontal direction of the screen when using it, and do not bend the cable along the vertical direction of the screen.

10. The screen of e-Paper is relatively fragile, please try to avoid dropping, bumping, and pressing hard.

11. We recommend that customers use the sample program provided by us to test with the corresponding development board.

# Working With Raspberry Pi

## Hardware Connection

When connecting the Raspberry Pi, you can directly insert the board into the 40PIN pin header of the Raspberry Pi, and pay attention to the correct pins.
If you choose to connect with an 8PIN cable, please refer to the pin correspondence table below:

| Raspberry Pi Pinout | | |
|---|---|---|
| e-Paper | Raspberry Pi BCM2835 | Board |
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | MOSI | 19 |
| CLK | SCLK | 23 |
| CS | CE0 | 24 |
| DC | 25 | 22 |

| RST | 17 | 11 |
| BUSY | 24 | 18 |

## Enable SPI Interface

- Open the Raspberry Pi terminal and enter the following command in the config interface:

```
sudo raspi-config

Choose Interfacing Options -> SPI -> Yes Enable SPI interface
```

```
1 Change User Password  Change password for the current user
2 Network Options        Configure network settings
3 Boot Options           Configure options for start-up
4 Localisation Options   Set up language and regional settings to match your location
5 Interfacing Options    Configure connections to peripherals
6 Overclock              Configure overclocking for your Pi
7 Advanced Options       Configure advanced settings
8 Update                 Update this tool to the latest version
9 About raspi-config     Information about this configuration tool
```

```
P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins
```

```
Would you like the SPI interface to be enabled?




                        <Yes>              <No>
```

Then reboot your Raspberry Pi:

```
sudo reboot
```

- Check /boot/config.txt, and you can see 'dtparam=spi=on' was written in.

```
# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on
```

- To make sure SPI is not occupied, it is recommended to close other drivers' coverage. You can use ls /dev/spi* to check whether SPI is occupied. If the terminal outputs /dev/spidev0.1 and /dev/spidev0.1, SPI is not occupied.

```
pi@raspberrypi:~ $ ls /dev/spi*
/dev/spidev0.0  /dev/spidev0.1
```

## C

- Install BCM2835

```
#Open the Raspberry Pi terminal and run the following command

wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.71.tar.gz

tar zxvf bcm2835-1.71.tar.gz

cd bcm2835-1.71/

sudo ./configure && sudo make && sudo make check && sudo make install

# For more information, please refer to the official website: http://www.
airspayce.com/mikem/bcm2835/
```

- Install WiringPi (Optional)

```
#Open the Raspberry Pi terminal and run the following command:

sudo apt-get install wiringpi

#For Raspberry Pi systems after May 2019 (earlier than before, you may no
t need to execute), you may need to upgrade:

wget https://project-downloads.drogon.net/wiringpi-latest.deb

sudo dpkg -i wiringpi-latest.deb

gpio -v

#Run gpio -v and version 2.52 will appear. If it does not appear, the ins
tallation is wrong.


#Bullseye branch system use the following command:

git clone https://github.com/WiringPi/WiringPi

cd WiringPi

./build

gpio -v

# Run gpio -v and version 2.60 will appear. If it does not appear, it mea
ns that there is an installation error.
```

- Download the demo via GitHub (You can skip this step if you have downloaded it.)

```
git clone https://github.com/waveshare/e-Paper.git

cd e-Paper/RaspberryPi_JetsonNano/
```

- Download the demo (You can skip this step if you have downloaded it.)

```
sudo apt-get install p7zip-full

wget  https://www.waveshare.com/w/upload/3/39/E-Paper_code.7z

7z x E-Paper_code.7z -O./e-Paper

cd e-Paper/RaspberryPi_JetsonNano/
```

- Compile the demo (Note: -j4 is to compile with 4 threads, the numbers can be modified by yourself; EPD=epd2in66b is to specify a macro definition, and epd2in66b corresponds to the test demo in the main function).

```
# Now at e-Paper/RaspberryPi_JetsonNano

cd c

sudo make clean

sudo make -j4 EPD=epd2in66b
```

- Run the demo

```
sudo ./epd
```

## Python

- Install the function library

```
sudo apt-get update

sudo apt-get install python3-pip

sudo apt-get install python3-pil

sudo apt-get install python3-numpy

sudo pip3 install RPi.GPIO

sudo pip3 install spidev
```

- Install function library (python2)

```
sudo apt-get update
```

```
sudo apt-get install python-pip
sudo apt-get install python-pil
sudo apt-get install python-numpy
sudo pip install RPi.GPIO
sudo pip install spidev
```

- Download the demo via GitHub (You can skip this step if you have downloaded it.)

```
git clone https://github.com/waveshare/e-Paper.git
cd e-Paper/RaspberryPi_JetsonNano/
```

- Download the demo (You can skip this step if you have downloaded it.)

```
sudo apt-get install p7zip-full
wget  https://www.waveshare.com/w/upload/3/39/E-Paper_code.7z
7z x E-Paper_code.7z -O./e-Paper
cd e-Paper/RaspberryPi_JetsonNano/
```

- Run the demo

```
# Make sure it's in e-Paper/RaspberryPi_JetsonNano/
cd python/examples/
python3 epd_2in66b_test.py
```

# Working With Arduino

## Hardware Connection

Use an 8PIN cable to connect, please refer to the pin correspondence table below:

| Connect To Arduino | | |
| --- | --- | --- |
| e-Paper | Arduino UNO | Mega2560 |
| VCC | 5V | 5V |
| GND | GND | GND |
| DIN | D11 | D51 |

| CLK | D13 | D52 |
|---|---|---|
| CS | D10 | D10 |
| DC | D9 | D9 |
| RST | D8 | D8 |
| BUSY | D7 | D7 |

## Install IDE
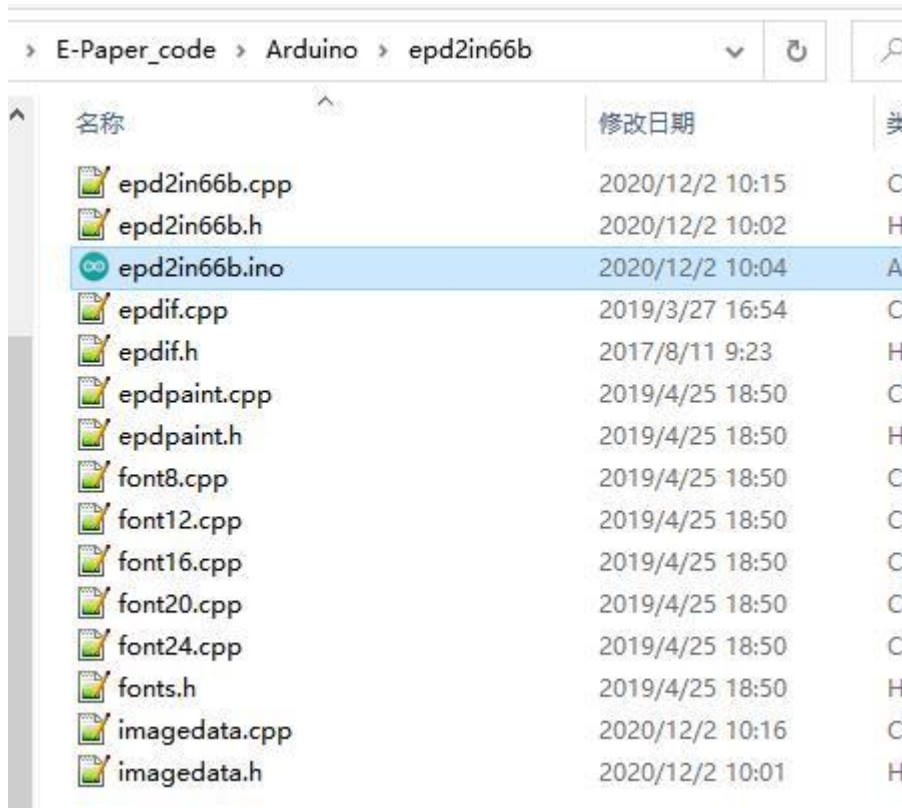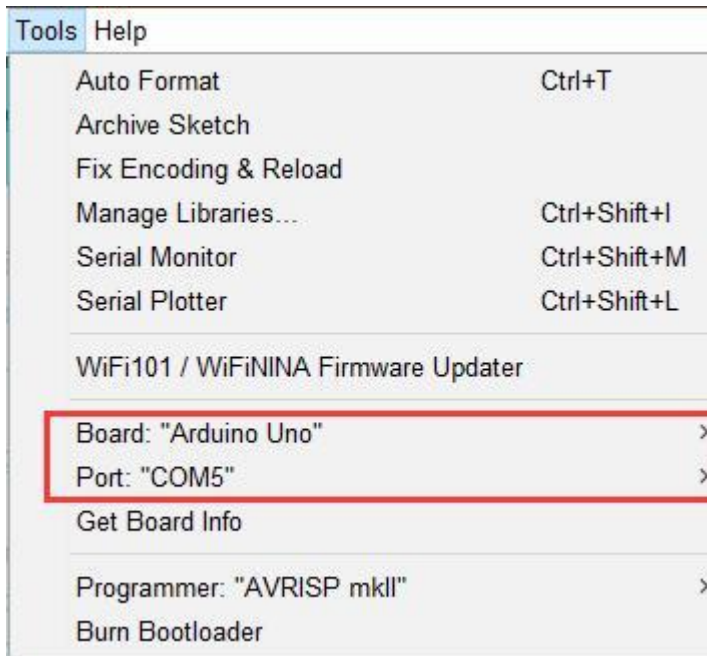
Arduino IDE Windows Install Guide

## Run The Demo

- Download the demo in Resource, unzip it to the "E-Paper_code" directory, and you can see the following content:



- Open the test demo: E-Paper_code\Arduino\epd2in66b\epd2in66b.ino

- Select the corresponding Board and Port in the Tools in the Arduino IDE.

```
Tools  Help
       Auto Format                      Ctrl+T
       Archive Sketch
       Fix Encoding & Reload
       Manage Libraries...              Ctrl+Shift+I
       Serial Monitor                   Ctrl+Shift+M
       Serial Plotter                   Ctrl+Shift+L

       WiFi101 / WiFiNINA Firmware Updater

       Board: "Arduino Uno"                        >
       Port: "COM5"                                >
       Get Board Info

       Programmer: "AVRISP mkII"                   >
       Burn Bootloader
```

- Finally, click upload, the upload is successful as follows (Arduino 1.8.13).

```
avrdude: AVR device initialized and ready to accept instructions

Reading | ############################################## | 100% 0.01s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file
avrdude: writing flash (6916 bytes):

Writing | ############################################## | 100% 2.64s

avrdude: 6916 bytes of flash written

avrdude done.  Thank you.
```

# Working With Jetson Nano

## Hardware Connection

The 40PIN pin of Jetson Nano is compatible with the 40PIN pin of Raspberry Pi and provides a Jetson.GPIO library with the same API as the RPI.GPIO library of Raspberry Pi, so the serial number connected here is the same as that of Raspberry Pi. The module can be directly inserted into the 40Pin headers of the Jetson Nano when using the 40PIN interface.

If you choose to connect with an 8PIN cable, please refer to the pin correspondence table below:

| e-Paper | Jetson Nano Developer Kit | |
| --- | --- | --- |
| | BCM2835 | Board |
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | 10 (SPI0_MOSI) | 19 |
| CLK | 11 (SPI0_SCK) | 23 |
| CS | 8 (SPI0_CS0) | 24 |
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

## C

- Download the demo via GitHub (you can skip this step if you have downloaded it.)

```
git clone https://github.com/waveshare/e-Paper.git

cd e-Paper/RaspberryPi_JetsonNano/
```

- Download the test demo: (you can skip this step if you have downloaded it.)

```
sudo apt-get install p7zip-full

wget  https://www.waveshare.com/w/upload/3/39/E-Paper_code.7z

7z x E-Paper_code.7z -O./e-Paper

cd e-Paper/RaspberryPi_JetsonNano/
```

- Compile the demo (Note: JETSON is the specified device, and RPI is not specified by default. -j4 is to compile by 4 threads, and the number can be changed by yourself. "EPD=epd2in66b" is to specify a macro definition, and "epd2in66b" corresponds to the test demo in the main function.)

```
# Now at e-Paper/RaspberryPi_JetsonNano

cd c
```

```
sudo make clean
sudo make JETSON -j4 EPD=epd2in66b
```

- Run the demo

```
sudo ./epd
```

## Python

- Install function library

```
sudo apt-get update
sudo apt-get install python3-numpy
sudo apt-get install python3-pip
sudo pip3 install Jetson.GPIO
```

- Download the demo via GitHub (you can skip this step if you have downloaded it.)

```
git clone https://github.com/waveshare/e-Paper.git
cd e-Paper/RaspberryPi_JetsonNano/
```

- Download the demo (you can skip this step if you have downloaded it.)

```
sudo apt-get install p7zip-full
wget  https://www.waveshare.com/w/upload/3/39/E-Paper_code.7z
7z x E-Paper_code.7z -O./e-Paper
cd e-Paper/RaspberryPi_JetsonNano/
```

- Run the demo

```
# Make sure it's in e-Paper/RaspberryPi_JetsonNano/
cd python/examples/
python3 epd_2in66b_test.py
```

# Working With Sunrise X3 Pi

# Hardware Connection

When connecting the Sunrise X3 Pi, you can directly insert the board into the 40PIN pin header of the Sunrise X3 Pi, and pay attention to the correct pins.

If you choose to connect with an 8PIN cable, please refer to the pin correspondence table below:

| Connect to Sunrise X3 Pi | | |
|---|---|---|
| e-Paper | Sunrise X3 Pi | |
| | BCM | Board |
| VCC | 3.3V | 3.3V |
| GND | GND | GND |
| DIN | MOSI | 19 |
| CLK | SCLK | 23 |
| CS | CE0 | 24 |
| DC | 25 | 22 |
| RST | 17 | 11 |
| BUSY | 24 | 18 |

# Enable SPI

- SPI is enabled by default. If you have disabled it, you can enable it by following the steps below.

- Enter the command: sudo srpi-config

Peripherals config

```
                    ┤ Choose an option ├
    spi2                                  disabled
    i2s1                                  okay
    i2s0                                  okay
    serial3                               okay
    i2c0                                  okay
```

Your

Are you
sure
want to
enable

<Yes>    <No>

```
├─────────┤ Choose an option ├─────────┤

    spi2                                  okay
    i2s1                                  okay
    i2s0                                  okay
    serial3                               okay
    i2c0                                  okay
```

## Python

- The corresponding library has been installed in the function. If you uninstall it accidentally, please use the following command to install it.

```
sudo apt-get update

sudo apt-get install python-pip

sudo apt-get install python-pil

sudo apt-get install python-numpy

sudo pip install Hobot.GPIO

sudo pip install spidev
```

- Download the demo via GitHub (skip this step if you have downloaded it).

```
git clone https://github.com/waveshare/e-Paper.git

cd e-Paper/RaspberryPi_JetsonNano/
```

- Download the demo (skip this step if you have downloaded it).

```
sudo apt-get install p7zip-full

wget  https://www.waveshare.com/w/upload/3/39/E-Paper_code.7z

7z x E-Paper_code.7z -O./e-Paper

cd e-Paper/RaspberryPi_JetsonNano/
```

- Run the demo

```
# Make sure you are in e-Paper/RaspberryPi_JetsonNano/

cd python/examples/

python3 epd_2in66b_test.py
```

# Working With STM32

## Hardware Connection

| Connect to STM32 | |
|---|---|
| e-Paper | STM32 |
| VCC | 3.3V |
| GND | GND |
| DIN | PA7 |

| CLK | PA5 |
|---|---|
| CS | PA4 |
| DC | PA2 |
| RST | PA1 |
| BUSY | PA3 |

## Run The Program

- [Click to download the demo](), and then unzip it into the E-Paper_code directory to see the following content.

E-Paper_code  ›

名称

- Arduino
- RaspberryPi_JetsonNano
- STM32
- Version_CN.txt
- Version_EN.txt
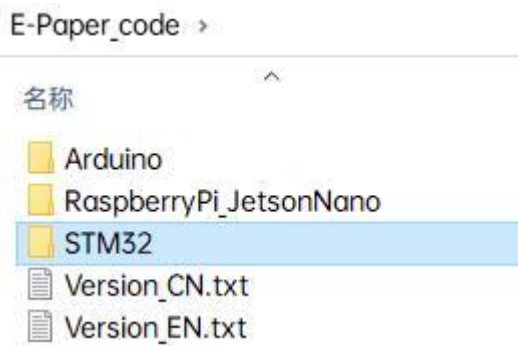
- Use Keil to open epd-demo.uvprojx in the E-Paper_code\STM32\STM32-F103ZET6\MDK-ARM directory

- Open Keil's compilation toolbar (usually already open).
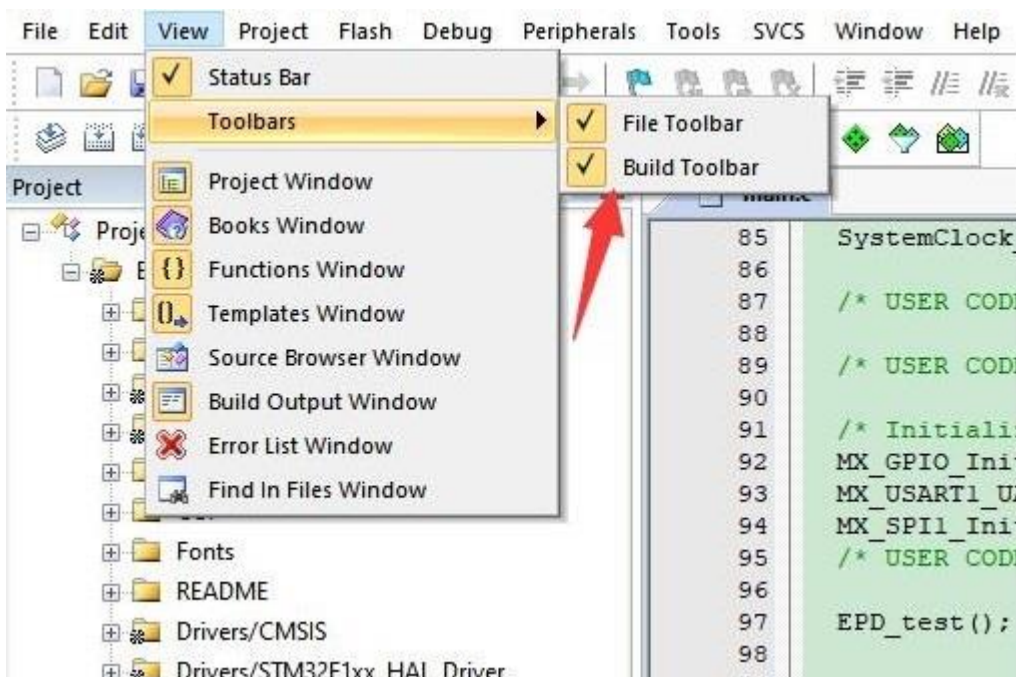


- Select the EPD_2in66b_test at the location shown in the picture.
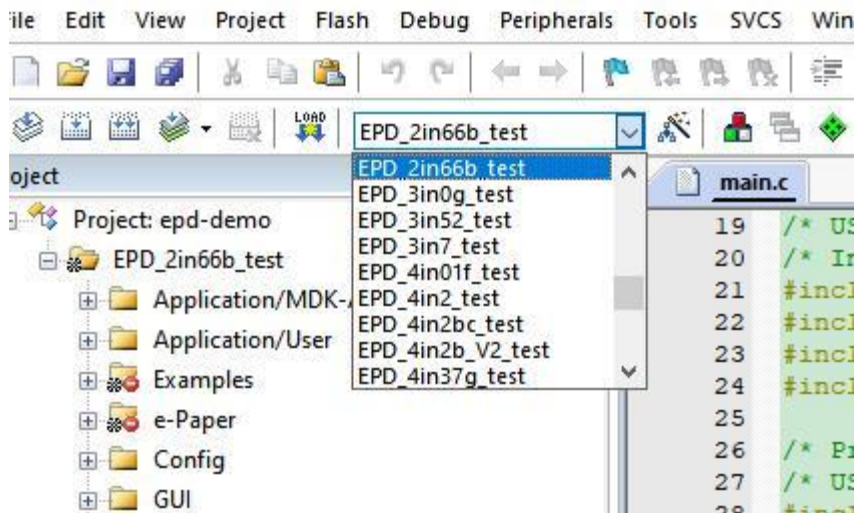
- Click to compile.

```
linking...
Program Size: Code=24732 RO-data=19288 RW-data=396 ZI-data=53428
FromELF: creating hex file...
"epd-demo\epd-demo.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:12
```

- Make sure the appropriate programmer is connected, then click LOAD to download the demo to the microcontroller.

# Resources

## Documentation

- 2.66inch e-Paper Shcematic
- 2.66inch e-Paper (B) Specification

## Demo code

- Example demo
- Github

## Development Resources

- E-Paper Floyd-Steinberg
- Image2Lcd.7z
- Image2Lcd Image Modulo
- Zimo221.7z
- E-Paper_API_Analysis

## Related Resources

- [Raspberry Pi Documentation](#)

# FAQ

## Question about Software

**Question:**The Raspberry Pi runs the python program and the following occurs?

```
pi@liguanghao:~/e-Paper/RaspberryPi_JetsonNano/python/examples $ python3 epd_7in
5_V2_test.py
INFO:root:epd7in5_V2 Demo
INFO:root:init and Clear
INFO:root:[Errno 2] No such file or directory
```

**Answer:**

- Enter the command: ls /dev/spi*

- The result may appear as shown in the figure

```
pi@raspberrypi:~ $ ls /dev/spidev*
/dev/spidev0.1   /dev/spidev1.1   /dev/spidev1.2
```

- This is because the SPI interface is occupied in the /boot/config.txt file.

```
dtparam=spi=on
dtoverlay=spi1-3cs
dtoverlay=mcp251xfd,spi0-0,interrupt=25
dtoverlay=mcp251xfd,spi1-0,interrupt=24
```

- Steps:

  - Delete the occupation of spi0-0 in the /boot/config.txt file.

  - Modify the location shown in the picture in the /e-Paper/RaspberryPi_JetsonNano/lib/waveshare_epd/epdconfig.py file and change it to 0,1.

```
 75
 76        # SPI device   bus = 0   device = 0
 77        self.SPI.open(0, 0)
 78        self.SPI.max_speed_hz = 4000000
 79        self.SPI.mode = 0b00
 80        return 0
```

**Answer:**

*Our demo uses stm32f103zet6. If the customer modifies other models in MDK, such as stm32f103rbt6, the ram space becomes smaller, and the stack size and heap size in the startup file need to be modified on the original basis.

**Answer:**

When transmitting B/W data, use Data Start Transmission 1; when transmitting RED data, use Data Start Transmission 2.

**Answer:**

In this case, the customer needs to reduce the position of the round brush and clear the screen after 5 rounds of refreshing (increasing the voltage of VCOM can improve the color, but it will increase the afterimage).

**Answer:**

The border display color can be set through the Border Waveform Control register or the VCOM AND DATA INTERVAL SETTING register.

**Answer:**

In this case, the customer needs to reduce the position of the game and clear the screen after 5 times of the game.

**Answer:**

The process of re-awakening the e-ink screen is actually the process of re-powering. Therefore, when the EPD wakes up, the screen must be cleared first, so as to avoid the afterimage phenomenon to the greatest extent.

**Answer:**

*It may be caused by the unsuccessful spi driver.

1. First check whether the wiring is correct.
2. Check whether the spi is turned on and whether the

parameters are configured correctly (spi baud rate, spi mode and other parameters).

**Question:**Why can't the image be displayed after full refreshing?

**Answer:**

The full refresh initialization function needs to be added when the ink screen is switched from partial refresh to full refresh.

**Question:**Why is the printing information normal when running a python program, but the ink screen does not respond?

**Answer:**

It may be a demo based on the BCM2835 library that has run the C language before. At this time, you need to restart the Raspberry Pi and then run the python demo.

**Question:**ImportError: No module named Image?

**Answer:**

*Install the imaging library using the command sudo apt-get install python-imaging

## Question about Hardware

**Question:**Can Arduino 5V drive the ink screen?

**Answer:**

Yes, now there is a level conversion chip onboard, supporting a 5V drive.

**Question:**What should be paid attention to when designing the driver board?

**Answer:**

- The rated input voltage of the ink screen is 2.3~3.6V. If it is a 5V system, level conversion is required. In addition, the voltage should not be lower than 2.5V, so as not to affect the display effect of the ink screen.

- Device selection can use the model in the schematic diagram we provide or choose according to the data sheet.

**Question:**Can I use analog SPI?

**Answer:**

Yes, pay attention to the correct timing.

**Question:**Why is the BUSY pin always busy?

**Answer:**

- Check if SPI communication is normal.

- Confirm whether the BUSY pin is normally initialized to input mode.

- It may be that there is no normal reset, try to shorten the duration of the low level during reset (because the power-off switch is added to the drive circuit, the reset low level is too long, which will cause the drive board to power off and cause the reset to fail).

- If the busy function sends the 0x71 command, you can try to comment it out.

**Answer:**

- 1.64inch, 2.36inch, 3inch, 0.5mm pitch, 26Pin.
- 1.02inch, 0.5mm pitch, 30Pin.
- 4.37inch, 7.3inch, 0.5mm pitch, 50Pin.
- The rest (non-parallel ports) are 0.5mm pitch, 24Pin.

**Answer:**

Cable socket 0.5-24pin rear-flip 2.0H (FPC connector).

## Question about Screen

**Answer:**

- 【Working conditions】Temperature range: 0~50°C; Humidity range: 35%~65%RH.
- 【Storage conditions】: Temperature range: below 30°C; Humidity range: below 55%RH; Maximum storage time: 6 months.
- 【Transportation conditions】: Temperature range: -25~70°C; Maximum transportation time: 10 days.

- 【After unpacking】: Temperature range: 20°C±5°C; Humidity range: 50±5%RH; Maximum storage time: Assemble within 72 hours.

**Answer:**

- refresh mode

  - Full refresh: The electronic ink screen will flicker several times during the refresh process (the number of flickers depends on the refresh time), and the flicker is to remove the afterimage to achieve the best display effect.

  - Partial refresh: The electronic ink screen has no flickering effect during the refresh process. Users who use the partial brushing function note that after refreshing several times, a full refresh operation should be performed to remove the residual image, otherwise the residual image problem will become more and more serious, or even damage the screen (currently only some black and white e-ink screens support partial refreshing, please refer to product page description).

  - refresh rate

  - During use, it is recommended that customers set the refresh interval of the e-ink screen to at least

180 seconds (except for products that support the local brush function).

- During the standby process (that is, after the refresh operation), it is recommended that the customer set the e-ink screen to sleep mode, or power off (the power supply part of the ink screen can be disconnected with an analog switch) to reduce power consumption and prolong the life of the e-ink screen. (If some e-ink screens are powered on for a long time, the screen will be damaged beyond repair.)

- During the use of the three-color e-ink screen, it is recommended that customers update the display screen at least once every 24 hours (if the screen remains the same screen for a long time, the screen burn will be difficult to repair).

- Application

  - The e-ink screen is recommended for indoor use. If it is used outdoors, it is necessary to avoid direct sunlight on the e-ink screen, and at the same time, take UV protection measures, because charged particles will dry out under strong light for a long time, resulting in loss of activity and failure to refresh. This situation is irreversible. When designing e-ink screen products, customers should pay attention to determining whether the use

environment meets the requirements of an e-ink screen.

**Question:**What is the refresh rate/lifetime of the e-ink screen?

 **Answer:**

Ideally, with normal use, it can be refreshed 1,000,000 times (1 million times).

**Question:**After using for a period of time, the screen refresh (full refresh) has a serious afterimage problem that cannot be repaired?

 **Answer:**

Power on the development board for a long time, after each refresh operation, it is recommended to set the screen to sleep mode or directly power off processing, otherwise, the screen may burn out when the screen is in a high voltage state for a long time.

**Question:**After the ink screen enters deep sleep mode, can it be refreshed again?

 **Answer:**

Yes, but you need to re-initialize the electronic paper with software.

**Question:**Why is the image displayed offset?

 **Answer:**

Maybe the SPI rate is too high, resulting in data loss, try to reduce the SPI rate.

Insufficient or unstable power supply leads to data loss.

The data cable is too long to cause data loss, the extension cable should not exceed 20cm.

**Answer:**

The display gray scale of electrophoretic electronic paper is determined by the spatial position of the particles in the Microcapsule or Microcup. The electrophoresis phenomenon occurs between black particles and white particles under the action of voltage. This voltage sequence that promotes the electrophoretic movement of the particles is the driving force of the electronic paper. waveform. The driving waveform is the core part of the electronic paper display, and the optimization of the driving waveform will directly affect the display effect of the display. The driving waveform file is used to describe the parameters formed by the voltage sequence that promotes the electrophoretic movement of the particles, and it needs to be called regularly when the electronic paper is refreshed.

Different batches of e-paper diaphragms and electrophoretic matrices require different voltage values when driving the display due to materials, manufacturing processes, etc. The waveform of the e-ink screen is reflected in the relationship between grayscale, voltage, and temperature. Generally speaking, after each batch of electrophoresis matrix is generated, there will be a corresponding waveform file in the form of a .wbf file. The film manufacturer will provide the waveform file and electrophoresis matrix to the

manufacturer of the electronic paper screen, and then the manufacturer of the electronic paper screen integrates the protection board, substrate, and driver and then provides it to customers; if the waveform file does not correspond to the screen, it is likely that the display cannot be displayed or the display effect is unsatisfactory. Generally, the waveform file has OTP built into the driver IC of the ink screen when leaving the factory, and some programs we provide also call external waveform files to drive the e-ink screen.

**Answer:**

LUT is the abbreviation of LOOK UP TABLE, and OTP is the abbreviation of ONE TIME PROGRAM. The original intention of LUT is to load waveform files, and the waveform files are divided into OTP and REGISTER. Among them, OTP is the built-in waveform storage method, and REGISTER is the external waveform storage method.

**Answer:**

*There are mainly two types of ink screens.

- 
  - One is to refresh the background image first.
  - The other is to alternately refresh old data and new data.

**Question:** How do I play in different positions at the same time?

**Answer:**

Simultaneous brushing in different locations needs to be operated in the program design, that is, first brushing the data of different locations into the electronic paper IC, and finally doing the Update/TurnOnDisplay uniformly.

**Question:** Does the three-color e-Paper have a red/yellow color difference?

**Answer:**

Yes, when e-Paper is batched, there will be some color difference, which is a normal phenomenon. Store the e-paper faces up to reduce the reddish/yellowishness to a certain extent.

**Question:** Are bare screens shipped with a film?

**Answer:**
with film.

**Question:** Does e-Paper have a built-in temperature sensor?

**Answer:**

At present, all screens have built-in temperature sensors, and you can also use an external LM75 temperature sensor with IIC pins.