

# 13.3inch e-Paper HAT

From Waveshare Wiki

Jump to: navigation, search

## Overview

### Introduction

- 13.3-inch EPD (Electronic Paper Display) HAT for Raspberry Pi, 1600 × 1200 resolution. It equips with IT8951 as the controller, can be controlled by USB/SPI/I2C interface, and supports partial refresh.
- It has the advantages of low power consumption, wide viewing angle, and clear display under direct sunlight, and is often used in display applications such as e-book readers, ink screen displays and industrial instruments.
- **13.3inch e-Paper is a big size screen, the glass panel, and FPC is fragile, please be careful when using it for development. we recommend you reinforce the FPC with scotch tape when developing. Please connect all the cables before powering the device, the device cannot support a hot plug.**

More (<http://www.waveshare.com/13.3inch-e-paper-hat.htm>)

### Features

- No backlight, keeps displaying the last content for a long time even when power is off.
- Low power consumption, basically power is only required for refreshing.
- Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards.
- USB/SPI/I2C interface, for connecting with host boards like Raspberry Pi/Nucleo, etc.

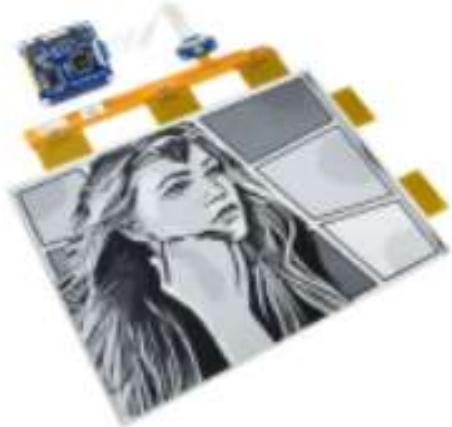
**13.3inch e-Paper HAT**



(<https://www.waveshare.com/13.3inch-e-paper-hat.htm>)

1600 x 1200, 13.3inch e-Paper HAT without driver board.

**13.3inch e-Paper Module**



(<https://www.waveshare.com/13.3inch-hdmi-e-paper.htm>)

1600 x 1200, 13.3inch e-Paper module 1600 x 1200

- Comes with online development resources and manual (examples for Raspberry Pi/STM32).

## Specifications

- Operating voltage: 5V
- Interface: USB/SPI/I2C
- Outline dimension: 285.80mm × 213.65mm × 0.78mm
- Display size: 270.40 × 202.80mm
- Dot pitch: 0.169 × 0.169mm
- Resolution: 1600 × 1200 pixels
- Display color: black, white
- Grayscale: 2-16 (1-4 bit)
- Full refresh time: <1s
- Total refresh power: 1.2W (typ.)
- Total standby power: 0.1W (typ.)
- Viewing angle: >170°
- Operation temperature: 0 ~ 50 °C
- Storage temperature: -25 ~ 70 °C

### Note:

- Refresh Time: The refresh time is based on experimental test data, and actual refresh time may vary. Actual performance should be considered. During the global refresh process, there may be flickering, which is a normal phenomenon.
- Refresh Power Consumption: Power consumption data is based on experimental test data. Actual power consumption may vary due to the presence of driver boards and different usage scenarios. Actual performance should be considered.
- Refreshing in low-temperature environments may result in color deviation. It is recommended to stabilize at 25°C for 6 hours before refreshing.

## Working principle

This product is an E-paper device adopting the image display technology of Microencapsulated Electrophoretic Display, MED. The basic principle is that charged nanoparticles suspended in a liquid migrate under the action of an electric field. The E-paper screen displays 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. (**Note that the e-Paper cannot support updating directly under sunlight.**)

## How to use

### Working with Windows PC (USB)

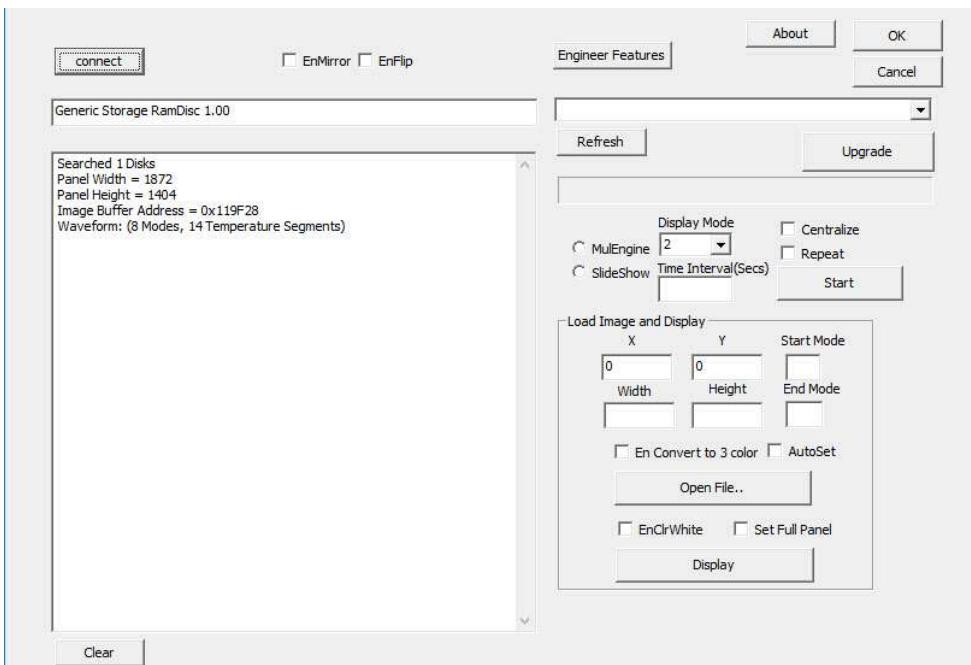
- Connect the display to IT8951 driver board as below:



(/wiki/File:10.3inch\_e-

Paper\_3.jpg)

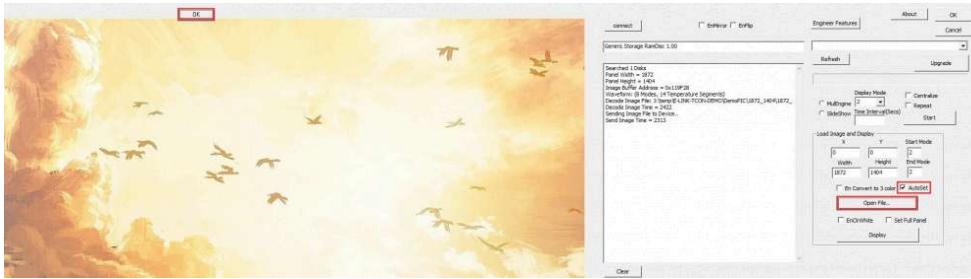
- Connect the USB interface of the IT8951 driver board to the PC.
- Download and open E-LINK-TCON-DEMO software.
- Click connect as below.



(/wiki/File:10.3inch-e-

Paper-HAT-D-Manual-02.jpg

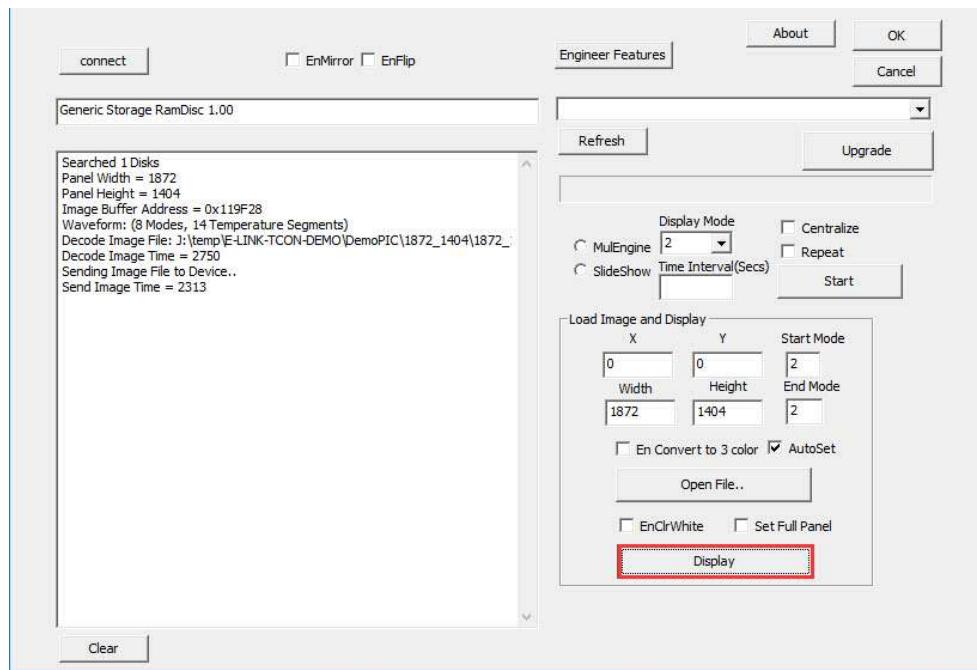
- Check option "AutoSet", then click "Open File" to open one picture for display. The browse diagram will be opened, and you should click "OK".



(/wiki/File:10.3inch-e-

Paper-HAT-D-Manual-03.jpg

- Click "display" to refresh the picture.



(/wiki/File:10.3inch-e-

### Paper-HAT-D-Manual-04.jpg

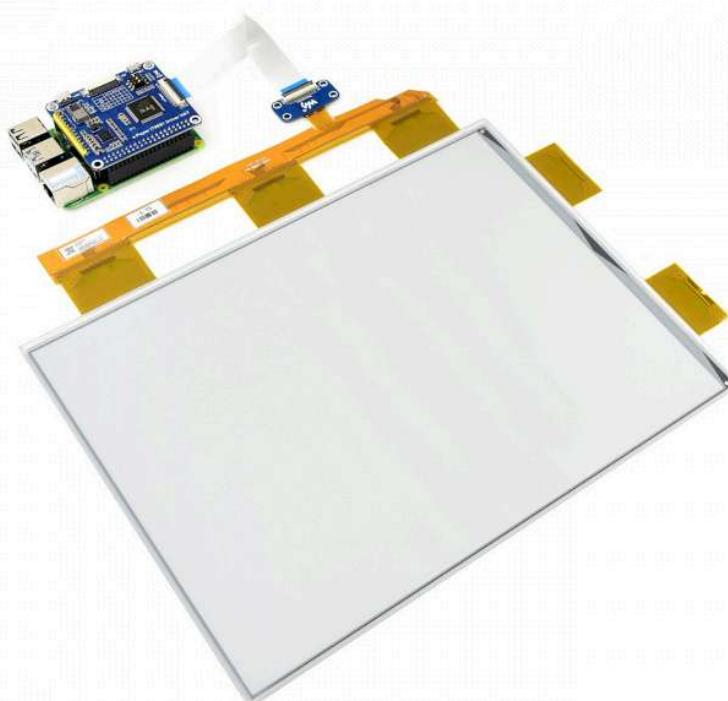
Note: If you want to use the USB interface to develop the e-Paper on Windows, you need to contact the Waveshare team and sign the NDA before you can get the E-LINK-TCON-DEMO source code.

Note: Currently not available for individual users.

## Working with Raspberry Pi (SPI)

- Step 1: Hardware connection:

Insert IT8951 driver board to GPIO of Raspberry Pi, using SPI interface to communicate with Raspberry PI.



(/wiki/File:10.3inch\_e-

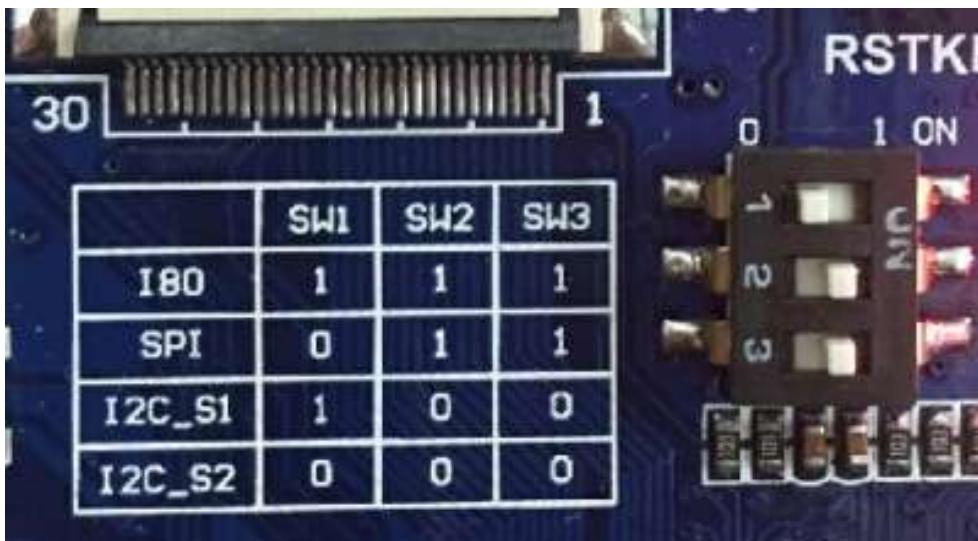
Paper3.jpg)

You can also connect by wires, as shown in the following table:

#### Connect to Raspberry Pi via SPI

<b>IT8951 Driver HAT</b>	<b>Raspberry Pi (BCM)</b>	<b>Description</b>
5V	5V	5V power input
GND	GND	Ground
MISO	P9	MISO Pin of SPI
MOSI	P10	MOSI Pin of SPI
SCK	P11	SCK Pin of SPI
CS	P8	Chip selection of SPI (Low active)
RST	P17	Reset pin (Low active)
HRDY	P24	Busy status pin (Low when busy)

- Step 2: Make sure you have switched the dip switch to SPI mode.



(/wiki/File:9.7inch-e-Paper-

HAT-Manual-14.jpg)

- Step 3: Install the C function library, open the Raspberry Pi terminal, and execute the following codes:

#### 1. Install bcm2835 libraries (Recommended for Pi4 (faster)).

```

wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
tar zxvf bcm2835-1.60.tar.gz
cd bcm2835-1.60
./configure
make
sudo make check
sudo make install
#For more details, you can refer to http://www.airspayce.com/mikem/bcm2835/

```

#### 2. Install lg library (Recommended for Pi5, BCM library currently does not support Pi5).

```

#Open the Raspberry Pi terminal and run the following commands:
wget https://github.com/joan2937/lg/archive/master.zip
unzip master.zip
cd lg-master
make
sudo make install
# For more details, you can refer to: https://github.com/gpiozero/lg

```

#### 3. Install gpiod library (optional).

```

#Open the Raspberry Pi terminal and run the following commands:
sudo apt-get update
sudo apt install gpiod libgpiod-dev

```

- Step 4: Enable SPI interface.

The Raspberry PI uses SPI to drive the IT8951 and then the e-Paper, hence, you need to open the SPI first:

Open the terminal command line and input:

```
sudo raspi-config
```

Choose Interfacing Options -> P4 -> SPI -> Yes.

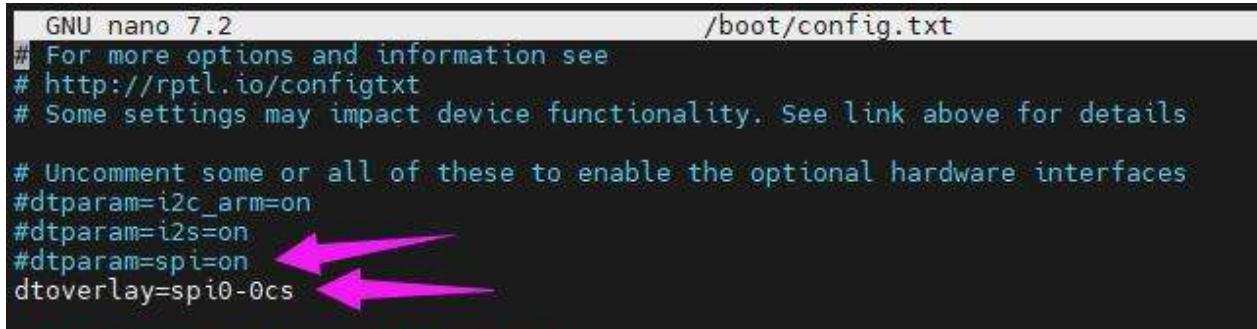


(/wiki/File:7.8inch\_e-

Paper\_HAT\_RPI01.png)

If you use Ig or gpiod library, you need to execute the following commands:

```
# Input the commands:  
sudo nano /boot/config.txt  
# Command out dtparam=spi=on and add  
dtoverlay=spi0-0cs
```



(/wiki/File:E-Paper\_Ig.jpg)

Input " sudo reboot " to reboot the Raspberry Pi to activate the configuration.

- Step 5: Download demo codes and compile them.

Method 1: Download from our website. (Recommended)

```

sudo apt-get install p7zip-full
sudo wget https://files.waveshare.com/upload/8/80/IT8951_20200319_Release.7z
7z x IT8951_20200319_Release.7z -O/IT8951
cd IT8951/
sudo make clean
sudo make -j4
# The BCM library is used by default, if you want to use other libraries, please change them in the Makefile or use the following commands:
make -j4 LIB=LGPI0      # for using lg library
make -j4 LIB=GPIOD      # for using gpiod library

```

## Method 2: Download from GitHub.

```

git clone https://github.com/waveshare/IT8951-ePaper.git
cd IT8951-ePaper/Raspberry
sudo make clean
sudo make -j4
# The BCM library is used by default, if you want to use other libraries, please change them in the Makefile or use the following commands:
make -j4 LIB=LGPI0      # for using lg library
make -j4 LIB=GPIOD      # for using gpiod library

```

- Check the VCOM value on the FPC.



(/wiki/File:6inch-HD-e-Paper-

HAT-Manual-06.png)

Add the VCOM value as a parameter and take the display mode as a input parameter, this model is mode 0.

```
sudo ./epd -1.50 0
```

- Demo execution phenomenon:

- 1) First, the e-paper will refresh the 16 gray bars partially in sequence.
- 2) Then demonstrate the demo of drawing lines, circles, rectangles, characters, and numbers.
- 3) Then respectively demonstrate the demo of displaying BMP pictures in 1bp, 2bp, and 4bp modes.
- 4) Next, demonstrate the effect of fast refresh in A2 mode.
- 5) Then display a gif animation.
- 6) Then demonstrate a Demo that counts the frame rate.
- 7) Finally, the e-paper is whitened in Init mode for long-term storage.

## Working with STM32 (SPI or I80)

### Driver interface selection

- Adopt SPI mode drive (recommended), the advantage is that it occupies less pins, the drive mode is simple, and the communication speed can meet the needs of most scenarios.
- Adopt 180 mode drive (I80 interface is only considered when the SPI interface communication speed does not meet the application requirements), the advantage is that the drive mode is relatively simple, the communication speed is very fast, but it occupies a lot of pins.
- Adopt I2C driver (not recommended, the communication speed is very slow).

### Driver board selection

- Since IT8951 needs to occupy a relatively large RAM space, the general STM32 single-chip microcomputer can not directly satisfy it, and additional chips such as SDRAM are required for RAM expansion. For the test here, we use Open429I ([https://www.waveshare.com/open\\_429i-c-standard.htm](https://www.waveshare.com/open_429i-c-standard.htm)) as the test board, and its onboard IS42S16400J (64-MBIT) SDRAM can fully meet the requirements of driving 10.3inch e-paper HAT.

## SPI

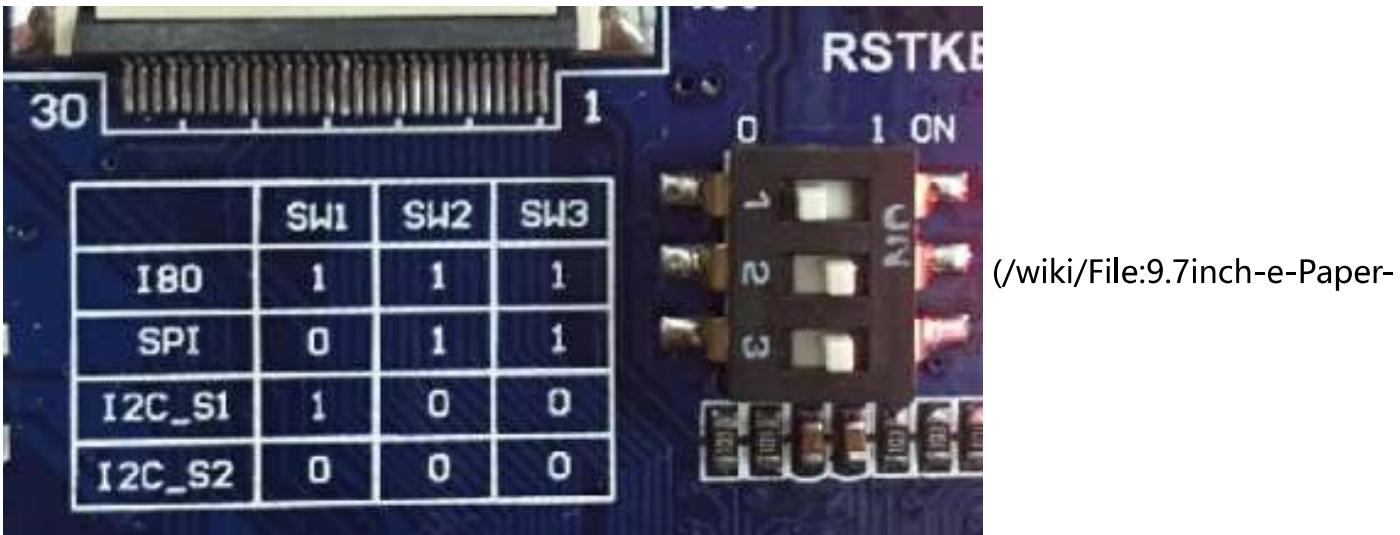
---

- 1) Follow the table below for hardware connection:

IT8951 Driver HAT	STM32	Description
5V	5V	5V power input
GND	GND	Ground
MISO	PE13	MISO Pin of SPI
MOSI	PE14	MOSI Pin of SPI
SCK	PE12	SCK Pin of SPI
CS	PE11	Chip selection of SPI (Low active)
RST	PC5	Reset pin (Low active)

HRDY	PA7	Busy status pin (Low when busy)
------	-----	---------------------------------

2) Set the DIP switch to SPI mode



HAT-Manual-14.jpg)

3) Refresh the picture with the DEMO.

You can download the Demo code (<https://files.waveshare.com/upload/d/da/Open429I-C-6-IT8951-Demo.7z>).

Open the project with keil: Open429I-C-6-IT8951-Demo\Project\6-IT8951\MDK-ARM\Project.uvproj.

Compile it, then open IT8951.h, and check if IT8951\_Interface\_SPI is defined. Compile it again and download to your board. After a few seconds, you should be able to see the e-paper start displaying images in a loop.

Line 14, '#define IT8951\_Interface\_SPI', is highlighted with a red rectangle.
"/>

```

1  /**
2  * ***** *****
3  * @author Waveshare Team
4  * *****
5  */
6
7 #ifndef _IT8951_H
8 #define _IT8951_H
9
10 #include "stm32f4xx.h"
11 #include "usart.h"
12
13 // #define IT8951_Interface_I2O
14 #define IT8951_Interface_SPI
15 // #define IT8951_Interface_I2C
16

```

(/wiki/File:9.7inch-e-Paper-HAT-Manual-15.jpg)

4) The information will be printed as below (115200, 8N1):

**SYSCLK: 180M****HCLK: 180M****PCLK1: 45M****PCLK2: 90M****IT8951 Example****Panel(W, H) = (1872, 1404)****Image Buffer Address = 119F28****FW Version = SWv\_0.1.**[\*\*LUT Version = M841\\_TFA5210\*\*](/wiki/File:10.3inch-e-</a></div>
<div data-bbox=)**IT8951DisplayExample 01****IT8951HostAreaPackedPixelWrite01****IT8951HostAreaPackedPixelWrite02****IT8951DisplayExample 02****IT8951DisplayExample 03****IT8951HostAreaPackedPixelWrite01**

Paper-HAT-D-Manual-09.jpg

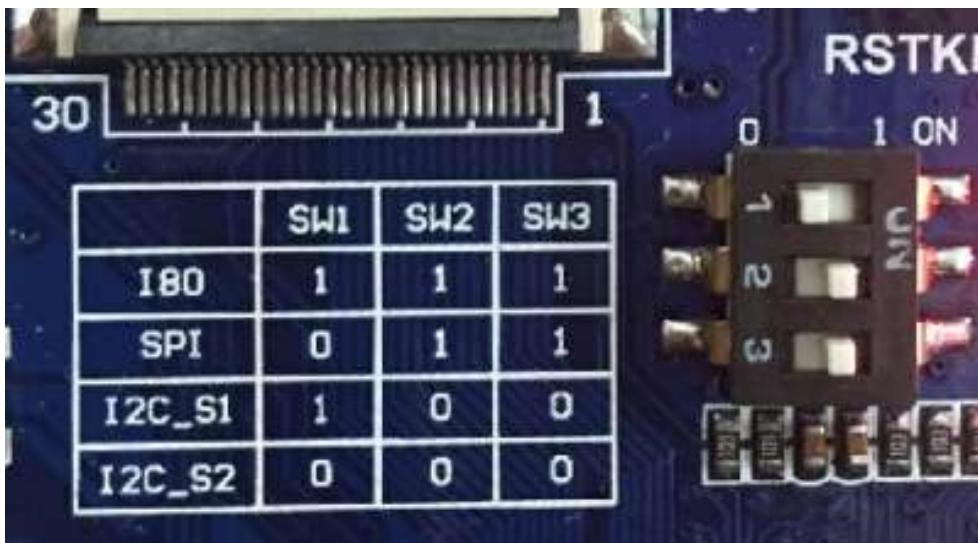
**I80**

## 1) Hardware connection:

Connect to STM32F429IGT6 (I80)

IT8951	STM32	Description
VCC	5V	5V Power input
GND	GND	Ground
DBUS0~DBUS15	PB0~PB15	Data pins
HWE	PC1	Write enable (Low active)
D/C	PC7	Data/Command (Low for command)
CSEL	PC6	Chip select (Low active)
HRD	PC3	Read enable (Low for active)
RST	PC0	Reset (Low for reset)
BUSY	PA7	Busy state output (Low for busy)

## 2) Set the DIP switch to I80 mode.



HAT-Manual-14.jpg)

3) Refresh the picture with DEMO.

You can download the Demo code (<https://files.waveshare.com/upload/d/da/Open429I-C-6-IT8951-Demo.7z>).

Open the project with keil: Open429I-C-6-IT8951-Demo\Project\6-IT8951\MDK-ARM\Project.uvproj.

Compile it, then open IT8951.h, and check if IT8951\_Interface\_SPI is defined. Compile it again and download to your board. After a few seconds, you should be able to see the e-paper start displaying images in a loop.

```

Project
STM32F429_439xx
  CMSIS
  MDK-ARM
  STM32F4xx_StdPeriph_Driver
  User
    main.c
    ctm32f4xx_it.c

main.c
IT8951.h
IT8951.c

9
10 #include "stm32f4xx.h"
11 #include "usart.h"
12
13 #define IT8951_Interface_I80
14 // #define IT8951_Interface_SPI
15 // #define IT8951_Interface_I2C
16

```

Paper-HAT-Manual-18.jpg)

(/wiki/File:9.7inch-e-

4) Information is printed to the serial port as below (115200, 8N1):

SYSCLK: 180M

HCLK: 180M

PCLK1: 45M

PCLK2: 90M

IT8951 Example

Panel(W, H) = (1872, 1404)

Image Buffer Address = 119F28

FW Version = SWv\_0.1.

(/wiki/File:10.3inch-e-

LUT Version = M841\_TFA5210

IT8951DisplayExample 01

IT8951HostAreaPackedPixelWrite01

IT8951HostAreaPackedPixelWrite02

IT8951DisplayExample 02

IT8951DisplayExample 03

IT8951HostAreaPackedPixelWrite01

'

Paper-HAT-D-Manual-10.jpg

## Display pictures

---

For easy porting of our demo code, we display pictures with a data matrix instead of the file system.

We should first convert the BMP picture to a data matrix (arrays), and use it in the demo code.

- 1) Prepare a BMP image, here an image with a resolution of 800 x 600 is used as an example.
- 2) Open BMP convert software (<https://files.waveshare.com/upload/b/b3/BmpCvt.zip>), Click File -> Open.. -> to open the picture as below:

Bitmap Converter for emWin V5.22 - J:\temp\zoo\_800\_600.bmp

File Edit View Image Options Help

Resolution: 800 \* 600

Colors: RGB

Zoom: 1.0 \*

Transparent —



(/wiki/File:10.3inch-e-

Loaded 24 bpp bitmap

Paper-HAT-D-Manual-11.jpg)

3) Click Image -> Convert to -> Gray256(8 BPP).

Bitmap Converter for emWin V5.22 - J:\temp\zoo\_800\_600.bmp

File Edit View Image Options Help

Resolution: 800 \* 600

Colors: 256

Zoom: 1.0 \*

Transparent —



(/wiki/File:10.3inch-e-

Conversion done

Paper-HAT-D-Manual-12.jpg)

4) Click File -> Save As... -> Choose "C" bitmap file (\*.c) -> input file name and click Save.

5) Choose 8 bits per pixel, and click OK. A C file will be saved to your PC.

### Format specification



Select format:

8 bit per pixel

Compressed\_RLE8

12bpp (444\_12),

12bpp (M444\_12),

12bpp (444\_12\_1),

12bpp (M444\_12\_1),

12bpp (444\_16),

12bpp (M444\_16),

(/wiki/File:9.7inch-e-

Without palette

OK

Cancel

Paper-HAT-Manual-21.jpg)

6) Add the C file to the keil project and delete unusable information.

```
*****
*          SEGGER Microcontroller GmbH & Co. KG
*          Solutions for real time microcontroller applications
*          www.segger.com
*****
*
* C-file generated by
*
*      Bitmap Converter for emWin V5.22.
*      Compiled Jul 4 2013, 12:18:24
*      (c) 1998 - 2013 Segger Microcontroller GmbH && Co. KG
*
*****
* Source file: zoo_800_600
* Dimensions: 800 * 600
* NumColors: 256
*
*****
*/
#include <stdlib.h>

#include "GUI.h"

#ifndef GUI_CONST_STORAGE
#define GUI_CONST_STORAGE const
#endif

extern GUI_CONST_STORAGE GUI_BITMAP bmzoo_800_600;

/*
*      Palette
*
*      Description
*      The following are the entries of the palette table.
*      The entries are stored as a 32-bit values of which 24 bits are
*      actually used according to the following bit mask: 0xBBGGRR
*
*      The Lower 8 bits represent the Red component.
*      The middle 8 bits represent the Green component.
*      The highest 8 bits represent the Blue component.
*/
static GUI_CONST_STORAGE GUI_COLOR _Colorszoo_800_600[] = {
    0x000000, 0x010101, 0x020202, 0x030303,
    0x040404, 0x050505, 0x060606, 0x070707,
```

0x080808, 0x090909, 0x0A0A0A, 0x0B0B0B,  
0x0C0C0C, 0x0D0D0D, 0x0E0E0E, 0x0F0F0F,  
0x101010, 0x111111, 0x121212, 0x131313,  
0x141414, 0x151515, 0x161616, 0x171717,  
0x181818, 0x191919, 0x1A1A1A, 0x1B1B1B,  
0x1C1C1C, 0x1D1D1D, 0x1E1E1E, 0x1F1F1F,  
0x202020, 0x212121, 0x222222, 0x232323,  
0x242424, 0x252525, 0x262626, 0x272727,  
0x282828, 0x292929, 0x2A2A2A, 0x2B2B2B,  
0x2C2C2C, 0x2D2D2D, 0x2E2E2E, 0x2F2F2F,  
0x303030, 0x313131, 0x323232, 0x333333,  
0x343434, 0x353535, 0x363636, 0x373737,  
0x383838, 0x393939, 0x3A3A3A, 0x3B3B3B,  
0x3C3C3C, 0x3D3D3D, 0x3E3E3E, 0x3F3F3F,  
0x404040, 0x414141, 0x424242, 0x434343,  
0x444444, 0x454545, 0x464646, 0x474747,  
0x484848, 0x494949, 0x4A4A4A, 0x4B4B4B,  
0x4C4C4C, 0x4D4D4D, 0x4E4E4E, 0x4F4F4F,  
0x505050, 0x515151, 0x525252, 0x535353,  
0x545454, 0x555555, 0x565656, 0x575757,  
0x585858, 0x595959, 0x5A5A5A, 0x5B5B5B,  
0x5C5C5C, 0x5D5D5D, 0x5E5E5E, 0x5F5F5F,  
0x606060, 0x616161, 0x626262, 0x636363,  
0x646464, 0x656565, 0x666666, 0x676767,  
0x686868, 0x696969, 0x6A6A6A, 0x6B6B6B,  
0x6C6C6C, 0x6D6D6D, 0x6E6E6E, 0x6F6F6F,  
0x707070, 0x717171, 0x727272, 0x737373,  
0x747474, 0x757575, 0x767676, 0x777777,  
0x787878, 0x797979, 0x7A7A7A, 0x7B7B7B,  
0x7C7C7C, 0x7D7D7D, 0x7E7E7E, 0x7F7F7F,  
0x808080, 0x818181, 0x828282, 0x838383,  
0x848484, 0x858585, 0x868686, 0x878787,  
0x888888, 0x898989, 0x8A8A8A, 0x8B8B8B,  
0x8C8C8C, 0x8D8D8D, 0x8E8E8E, 0x8F8F8F,  
0x909090, 0x919191, 0x929292, 0x939393,  
0x949494, 0x959595, 0x969696, 0x979797,  
0x989898, 0x999999, 0x9A9A9A, 0x9B9B9B,  
0x9C9C9C, 0x9D9D9D, 0x9E9E9E, 0x9F9F9F,  
0xA0A0A0, 0xA1A1A1, 0xA2A2A2, 0xA3A3A3,  
0xA4A4A4, 0xA5A5A5, 0xA6A6A6, 0xA7A7A7,  
0xA8A8A8, 0xA9A9A9, 0xAAAAAA, 0xABABAB,  
0xACACAC, 0xADADAD, 0xAEAEAE, 0xAFAFAF,  
0xB0B0B0, 0xB1B1B1, 0xB2B2B2, 0xB3B3B3,  
0xB4B4B4, 0xB5B5B5, 0xB6B6B6, 0xB7B7B7,  
0xB8B8B8, 0xB9B9B9, 0xBABABA, 0xBBB BBBB,  
0xBCBCBC, 0xBDDBDB, 0xBEBEBE, 0xBF BF BF,  
0xC0C0C0, 0xC1C1C1, 0xC2C2C2, 0xC3C3C3,

```

0xC4C4C4, 0xC5C5C5, 0xC6C6C6, 0xC7C7C7,
0xC8C8C8, 0xC9C9C9, 0xCACACA, 0xCBCBCB,
0xCCCCCC, 0xCD CDCD, 0xCECECE, 0xCFCFCF,
0xD0D0D0, 0xD1D1D1, 0xD2D2D2, 0xD3D3D3,
0xD4D4D4, 0xD5D5D5, 0xD6D6D6, 0xD7D7D7,
0xD8D8D8, 0xD9D9D9, 0xDADADA, 0xDBDBDB,
0xDCDCDC, 0xDDDDDD, 0xDEDEDE, 0xDFDFDF,
0xE0E0E0, 0xE1E1E1, 0xE2E2E2, 0xE3E3E3,
0xE4E4E4, 0xE5E5E5, 0xE6E6E6, 0xE7E7E7,
0xE8E8E8, 0xE9E9E9, 0xEAEEAA, 0xEBEBEB,
0xEC ECEC, 0xED EDED, 0xEEEEEE, 0xEFEEEF,
0xF0F0F0, 0xF1F1F1, 0xF2F2F2, 0xF3F3F3,
0xF4F4F4, 0xF5F5F5, 0xF6F6F6, 0xF7F7F7,
0xF8F8F8, 0xF9F9F9, 0xFAFAFA, 0xFBFBFB,
0xFCFCFC, 0xFDFDFD, 0xFEFEFE, 0xFFFFFFF
};

static GUI_CONST_STORAGE GUI_LOGPALETTE _Palzoo_800_600 = {
    256, // Number of entries
    0, // No transparency
    &_Colorszoo_800_600[0]
};

GUI_CONST_STORAGE GUI_BITMAP bmzoo_800_600 = {
    800, // xSize
    600, // ySize
    800, // BytesPerLine
    8, // BitsPerPixel
    _aczoo_800_600, // Pointer to picture data (indices)
    &_Palzoo_800_600 // Pointer to palette
};

```

## 7) Modify the codes

```
static GUI_CONST_STORAGE unsigned char _aczoo_800_600[] = {
```

to this one. (You can change the name of the array as you like.)

```
const unsigned char zoo_800_600[] = {
```

## 8) Modify related codes in IT8951.C as below:

```

extern const unsigned char zoo_800_600[];
void IT8951DisplayExample3()
{
    IT8951LdImgInfo stLdImgInfo;
    IT8951AreaImgInfo stAreaImgInfo;
    TWord width = gstI80DevInfo.usPanelW;
    TWord high = gstI80DevInfo.usPanelH;
    TDWord i;

    for (i = 0; i < width*high; i++)
    {
        gpFrameBuf[i] = zoo_800_600[i];
    }

    IT8951WaitForDisplayReady();

    //Setting Load image information
    stLdImgInfo.ulStartFBAddr      = (TDWord)gpFrameBuf;
    stLdImgInfo.usEndianType       = IT8951_LDIMG_L_ENDIAN;
    stLdImgInfo.usPixelFormat      = IT8951_8BPP;
    stLdImgInfo.usRotate          = IT8951_ROTATE_0;
    stLdImgInfo.ulImgBufBaseAddr = gulImgBufAddr;
    //Set Load Area
    stAreaImgInfo.usX      = 0;
    stAreaImgInfo.usY      = 0;
    stAreaImgInfo.usWidth   = width;
    stAreaImgInfo.usHeight  = high;

    IT8951HostAreaPackedPixelWrite(&stLdImgInfo, &stAreaImgInfo); //Display function 2
    IT8951DisplayArea(0, 0, gstI80DevInfo.usPanelW, gstI80DevInfo.usPanelH, 2);
}

```

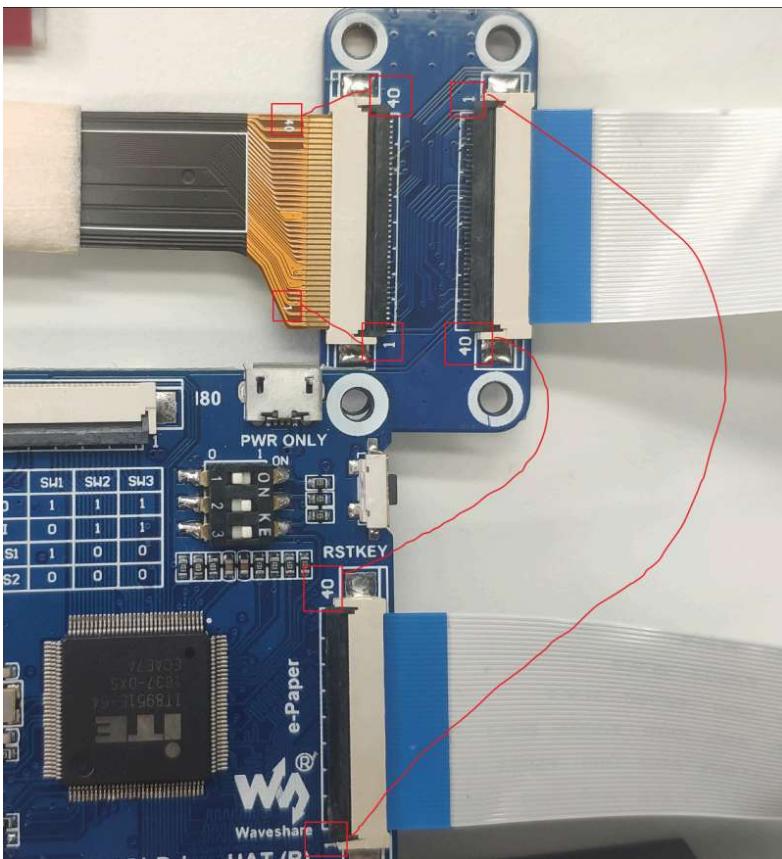
**【Note】** Guides provided here is all about how to use the 13.3inch e-Paper HAT. If you have any questions about how to modify and develop your own codes, please refer to resources of [IT8951 #Resources](#).

## Precaution

### Hardware connection

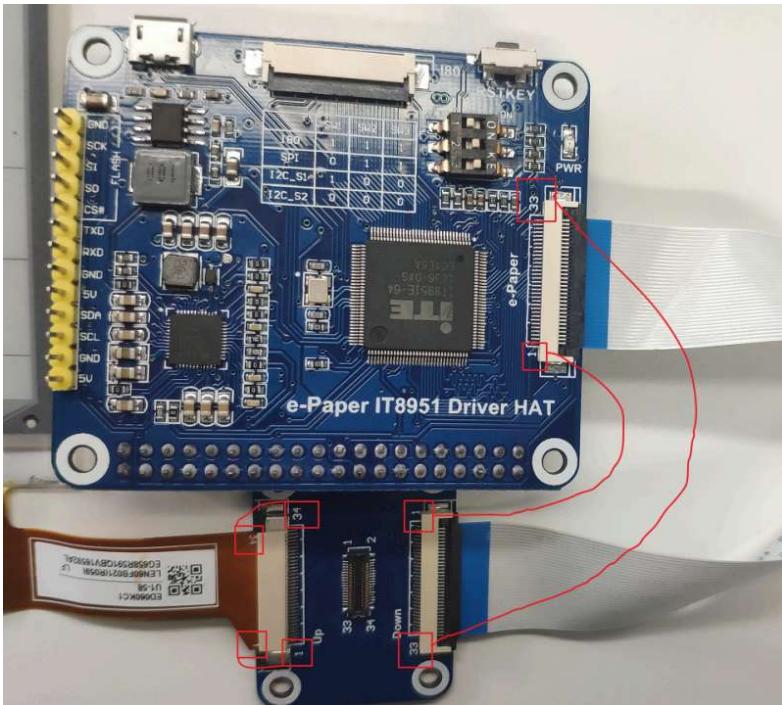
For different e-paper screen models, due to the product upgrades and improvements, there may be some differences in hardware connection compared with the picture. So the specific FPC line pin label shall prevail:

- Check example 1:



(/wiki/File:20191213143317.png)

- Check example 2:

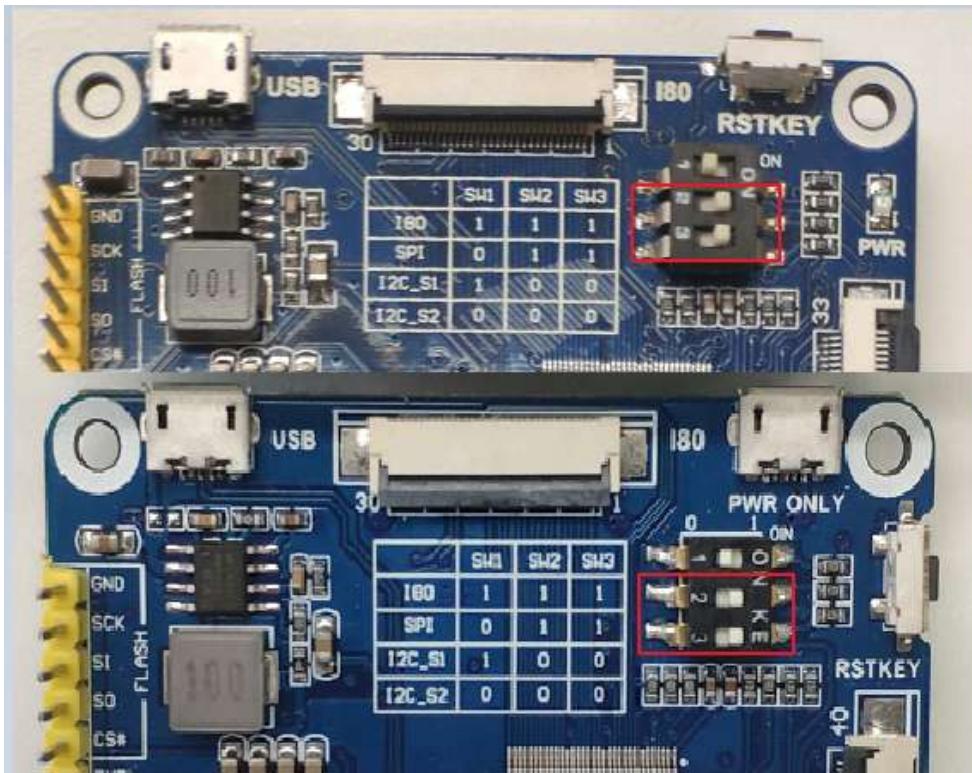


(/wiki/File:20191213141630.png)

## DIP switch

- We currently don't recommend using the I2C interface for the driver, it is recommended to use a USB, I80, or SPI interface.

- Whether you are using a USB interface driver, I80 interface driver, or SPI interface driver, you need to turn the DIP switch (marked in the red frame in the picture below) to the ON position on the right, as shown in the picture below:

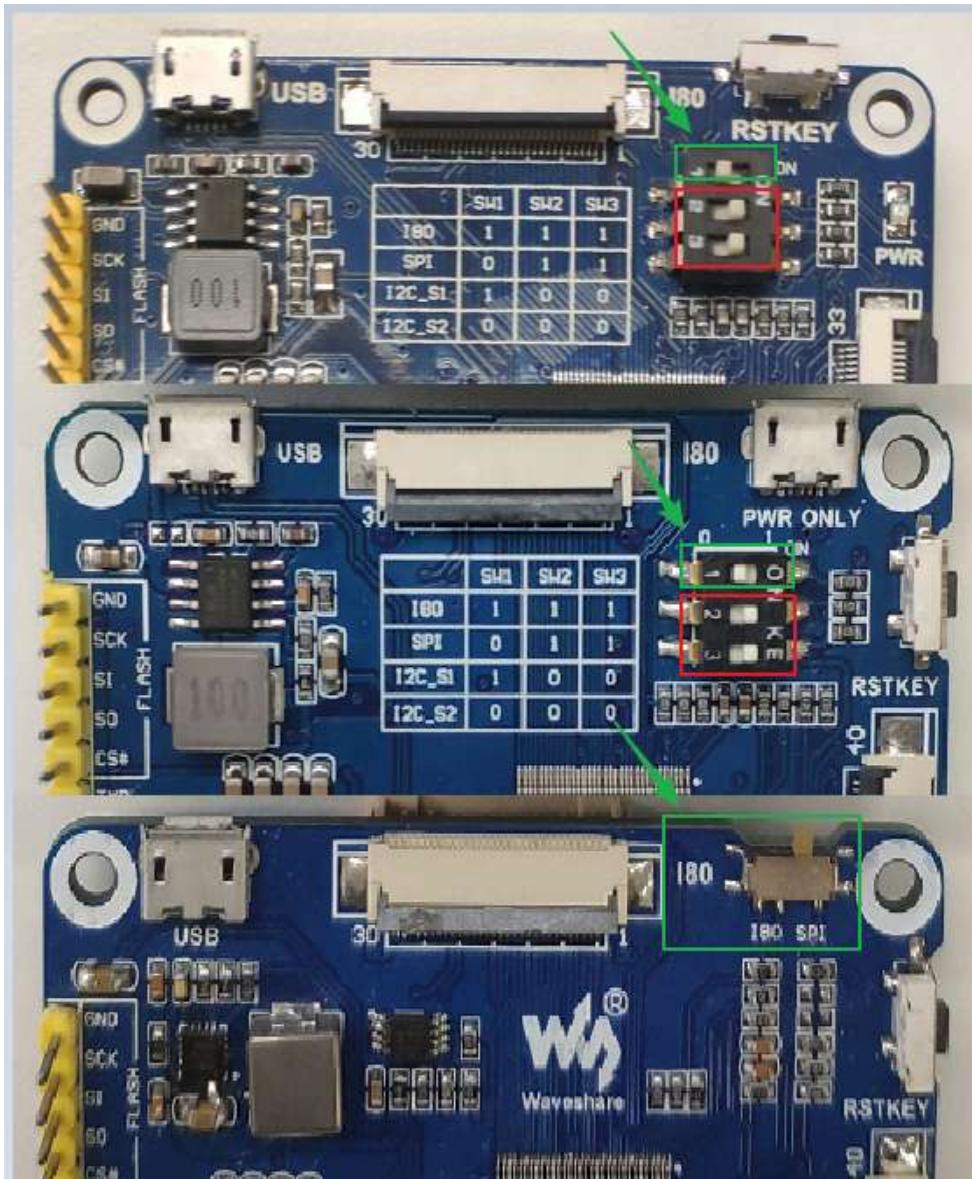


(/wiki/File:20191213160132.png)

- If you use USB to connect the driver board to the PC and the E-LINK-TCON-DEMO cannot recognize IT8951, please check whether the two DIP switches (marked in the red frame) are connected. Only the two DIP switches marked are both on, and the driver board can be recognized by the PC.

## SPI or I80 driver

- If you use a USB driver, you don't need to pay attention to the DIP switch in the green frame as shown in the picture below.
- If you want to use the SPI driver or the I80 driver, you need to pay attention to the dial switch in the green frame as shown in the picture below. If it is dialed to the I80 side, it means using the I80 driver, if it is dialed to the SPI side, it means using the SPI driver:



(/wiki/File:20191213161905.png)

## Codes Description

### New Features

1. Update 16 grayscale images in 4bpp mode to reduce the data size while transmitting via SPI, avoiding the waste of RAM and transmission time caused by the previous version using 8bpp to refresh the 16-level grayscale image.
2. Double SPI transmitting speed. Note that this feature can only be valid in Raspberry Pi 3, it is invalid in Pi 4 because of the improvement of CPU frequency.
3. The time interval for refreshing the picture is reduced to 1/4 of the original. The last version of Demo codes cost 10s to update a picture on 10.3inch e-Paper (D) in GC16 Mode, with the new version, it is reduced to 3s.

4. Fix the bug that the memory leaks problem occurs after opening a BMP file.
5. Add 1bpp, 2bpp, 3bpp, and 4bpp modes support.
6. Add A2 Mode (only work for 1bpp mode).
7. Add GUI functions (Draw point, line, circle, rectangle, and string display). Supports changing gray.
8. Add 1bpp, 2bpp, 4bpp, and 8bpp refresh support for BMP pictures, if you choose 1bpp, it also supports A2 mode refresh.
9. Add a demo for displaying GIF pictures, which can write multiple pictures into the IT8951 cache, and directly swipe the data of different addresses in the cache into the ink screen for display, eliminating the data transmission process between RPi and IT8951, refresh frame rate up to 7fps.
10. Add a frame rate test demo, which is convenient for testing and refreshing areas of different sizes, 1bpp, 2bpp, 4bpp, 8bpp, and A2 mode, GC16 mode, and refreshed frame rate.
11. Enhance the driver, to fix the bug of blurred display in the partial area.
12. Support 4-byte alignment for 6inch ePaper HAT and 6inch HD ePaper HAT in 1bpp mode.
13. Add VCOM setting and clear ePaper function (running Demo codes directly) to protect ePaper from damage.
14. Improving the structure of driver codes for higher code readability.

## Examples Description

### Display 16 bars in grayscale order

- Function: Display *ColorPalette* Example
- This function is used to display 16 bars in different grayscale. It uses 4bpp in GC16 mode.

### Draw dots, lines, circles, rectangles and write characters

- Function: Display *CharacterPattern* Example
- This function is used to draw points, lines, circles, and rectangles as well as write characters. It supports 1bpp, 2bpp, 4bpp and 8bpp mode. If you use 1bpp, it can also support the A2 mode.

### Display BMP image

- Function: Display *BMP* Example

- This function is used to display a BMP image. It supports 1bpp, 2bpp, 4bpp and 8bpp mode. If you use 1bpp, it can also support the A2 mode.

## Refresh in A2 mode

---

- Function: Dynamic *Refresh* Example
- This function is used to refresh in A2 Mode. In this demo, the refresh area will be refreshed several times, and the area of the refresh area will gradually become larger. After refreshing several times, an afterimage will appear. The afterimage will be cleared using INIT mode. Through this demo, you can intuitively feel the refresh speed of the A2 mode.

## Display GIF image

---

- Function: Dynamic *GIF* Example
- This Demo demonstrates how to refresh a GIF image. In this demo, a GIF image is divided into 7 bmp images in advance, and the 7bmp images are written to the cache of a continuous address of IT8951 first. During display, the image data will be read from the cache of the corresponding address of IT8951 in turn and refreshed on the e-paper screen. Because there is no process of data transmission between RPi and IT8951, the refresh speed will be very fast. This Demo also demonstrates the limit speed of the IT8951 refreshing the ink screen (about 7fps).

## FPS Testing

---

- Function: Check *FrameRate* Example
- This function is used to test the refresh frame rate when refreshing areas of different sizes, 1bpp, 2bpp, 4bpp, 8bpp, as well as A2 mode, GC16 mode. In this demo, the time to refresh 10 frames of images will be automatically counted, and frames per second (fps) is automatically calculated.

## Related Description

### Mode

---

IT8951 has flashed different firmware for different resolution screens. Different firmware has different refresh modes. See the Mode Description (<https://files.waveshare.com/upload/c/c4/E-paper-mode-declaration.pdf>) for details. The modes used in the demo are INIT mode, GC16 mode, and A2 mode.

Below is a brief description of the relevant modes:

Mode	Description	6inch/6inch HD	7.8inch/9.7inch/10.3inch
INIT	This mode is used for clearing the display. If you use A2 mode for refreshing, we recommend you use the INIT mode to clear display after refreshing several times.	Mode0	Mode0
GC16	This mode is used for refreshing ePaper with 16 grayscale. GC16 mode can provide best dispaly effect.	Mode2	Mode2
A2	A2 mode can only support 2 grayscale, however, the refresh speed is the fastest.	Mode4	Mode6

```
//basic mode definition
UBYTE INIT_Mode = 0;
UBYTE GC16_Mode = 2;
//A2_Mode's value is not fixed, is decided by the firmware's LUT
UBYTE A2_Mode = 6;
```

```
if( strcmp(LUT_Version, "M641") == 0 ){
    //6inch e-Paper HAT(800,600), 6inch HD e-Paper HAT(1448,1072), 6inch HD touch e-Paper H
AT(1448,1072)
    A2_Mode = 4;
    Four_Byte_Align = true;
}else if( strcmp(LUT_Version, "M841") == 0 ){
    //9.7inch e-Paper HAT(1200,825)
    A2_Mode = 6;
}else if( strcmp(LUT_Version, "M841_TFA2812") == 0 ){
    //7.8inch e-Paper HAT(1872,1404)
    A2_Mode = 6;
}else if( strcmp(LUT_Version, "M841_TFA5210") == 0 ){
    //10.3inch e-Paper HAT(1872,1404)
    A2_Mode = 6;
}else{
    //default set to 6 as A2 Mode
    A2_Mode = 6;
}
```

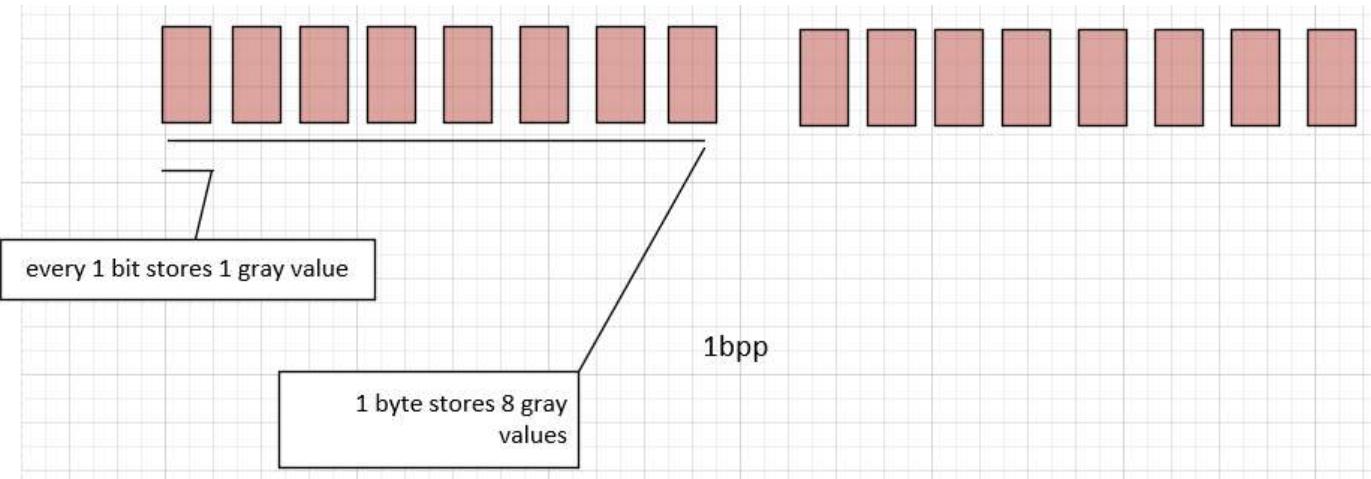
## bpp

bpp(Bits Per Pixel), stands for the bits costed by every pixel. The current versions of ePaper support 1bpp, 2bpp, 4bpp, and 8bpp modes.

- **1bpp**

- Evey pixel use 1 bit.
- It supports  $2^{1=2}$  gray, it is used for A2 mode.
- Evey byte contains 8 pixels.

- Save in RAM in big-endian format:

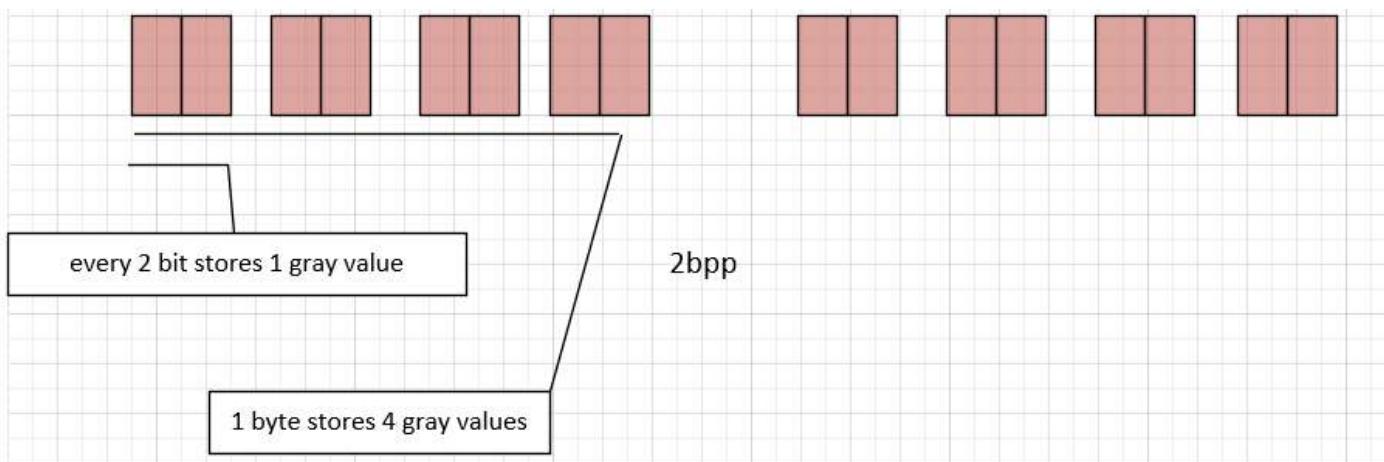


(/wiki/File:20191219165020.png)

- In IT8951, the little-endian format is used by default, and it is necessary to convert the big-endian format to the little-endian format.

#### ■ 2bpp

- Every pixel uses 2 bits.
- Support  $4(2^2=4)$  grays.
- Every byte contains 4 pixels.
- Save in RAM in big-endian format:

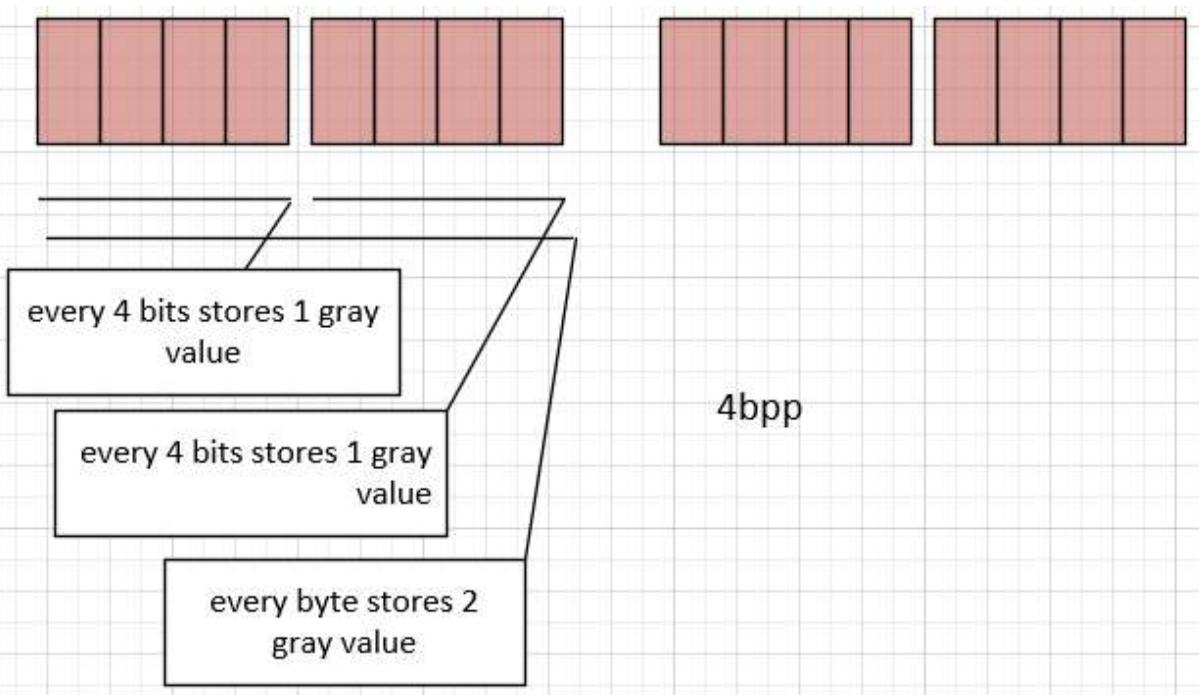


(/wiki/File:20191219164855.png)

- In IT8951, the little-endian format is used by default, and it is necessary to convert the big-endian format to the little-endian format.

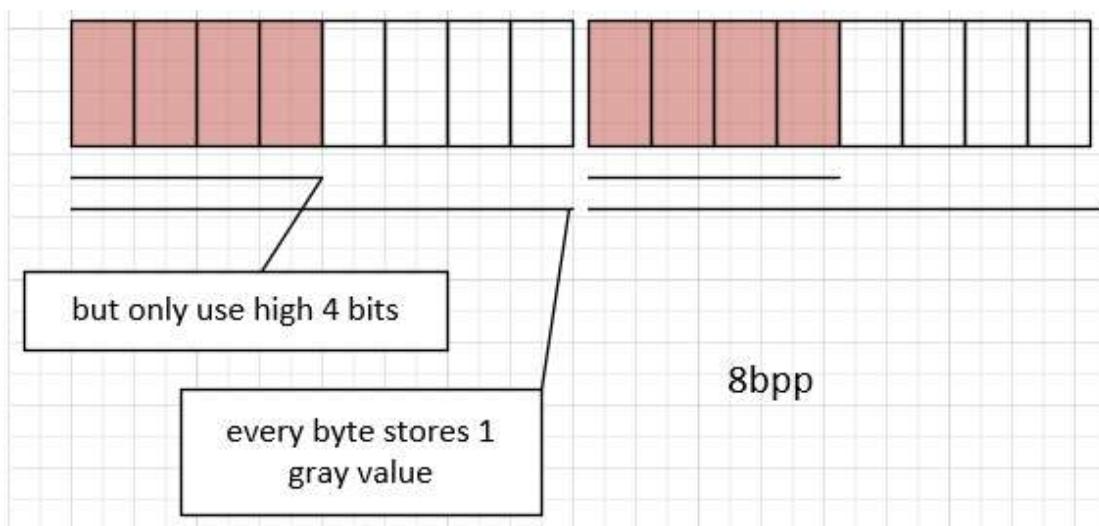
#### ■ 4bpp

- Every pixel uses 4 bits.
- Support  $16(2^4=16)$  grays.
- Every byte contains 2 pixels.
- Save in RAM in big-endian format:



(/wiki/File:20191219164916.png)

- In IT8951, the little-endian format is used by default, and it is necessary to convert the big-endian format to the little-endian format.
- **It is recommended to use 4bpp for a refreshing, which can display 16 levels of grayscale, and compared to 8bpp, the amount of transmitted data is reduced by half, the transmission speed is twice as fast, and there is no difference in display effect.**
- **8bpp**
  - Every pixel uses 8 bits.
  - Support 256( $2^8=256$ ) grays, however, IT8951 only uses the high four bits, only supports 16 grays.
  - Every byte contains 1 pixel.
  - Save in RAM in big-endian format:



(/wiki/File:20191219164949.png)

- In IT8951, the little-endian format is used by default, and it is necessary to convert the big-endian format to the little-endian format.
- To get the corresponding grayscale image from the original image. The specific operation in the program is: whether drawing points, drawing lines, or getting the image, each pixel point obtained is 1 byte (8 bits), if you want to get the corresponding grayscale, you only need to get the corresponding high bit of the byte. For example, if you want to get 2bpp pixel points, you only need to get the high 2 bits from the pixels of 8bpp (8 bits). The specific operation is shown in the following program, in which each byte in RAM is also converted from big-endian to little-endian.

```
UDOUBLE Addr = X * (Paint.BitsPerPixel) / 8 + Y * Paint.WidthByte;
switch( Paint.BitsPerPixel ){
    case 8:{  
        Paint.Image[Addr] = Color & 0xF0;  
        break;  
    }  
    case 4:{  
        Paint.Image[Addr] &= ~( (0xF0) >> (7 - (X*4+3)%8) );  
        Paint.Image[Addr] |= (Color & 0xF0) >> (7 - (X*4+3)%8);  
        break;  
    }  
    case 2:{  
        Paint.Image[Addr] &= ~( (0xC0) >> (7 - (X*2+1)%8) );  
        Paint.Image[Addr] |= (Color & 0xC0) >> (7 - (X*2+1)%8);  
        break;  
    }  
    case 1:{  
        Paint.Image[Addr] &= ~( (0x80) >> (7 - X%8) );  
        Paint.Image[Addr] |= (Color & 0x80) >> (7 - X%8);  
        break;  
    }  
}
```

## 4-byte alignment description

In the actual test, we found that for those 3 products: 6inch e-Paper HAT, 6inch HD e-Paper HAT, and 6inch HD touch e-Paper HAT, when refreshing in 1bpp mode, we need to make the starting point of the refresh area X and the refresh width W to perform 4-byte (32bit) alignment, otherwise, the image in the refresh area will display abnormally. The specific operation is shown in the following demo:

```

if( strcmp(LUT_Version, "M641") == 0 ){
    //6inch e-Paper HAT(800,600), 6inch HD e-Paper HAT(1448,1072), 6inch HD touch e-Paper H
AT(1448,1072)
    A2_Mode = 4;
    Four_Byte_Align = true;
}else if( strcmp(LUT_Version, "M841") == 0 ){
...
}
if(Four_Byte_Align == true){
    In_4bp_Refresh_Area_Width = Panel_Width - (Panel_Width % 32);
}else{
    In_4bp_Refresh_Area_Width = Panel_Width;
}
X_Start = Min_X < 32 ? 0 : Min_X - (Min_X % 32);
Debug("X_Start:%d\r\n",X_Start);
X_End = ( Max_X + (32 - (Max_X % 32)) ) > Touch_Pannel_Area_Width ? ( Max_X - (Max_X % 32)
) : ( Max_X + (32 - (Max_X % 32)) );
Debug("X_End:%d\r\n",X_End);
Y_Start = Min_Y;
Debug("Y_Start:%d\r\n",Y_Start);
Y_End = Max_Y;
Debug("Y_Start:%d\r\n",Y_End);
Width = X_End - X_Start;
if(Width<=0){
    Width = 32;
}
Debug("Width:%d\r\n",Width);
Height = Y_End-Y_Start;
if(Height<=0){
    Height = 32;
}
Debug("Height:%d\r\n",Height);

```

## SPI transmission speed description

Due to the difference in CPU frequency between Raspberry Pi 3 and Raspberry Pi 4:

- Raspberry Pi 3 can still transmit normally when the frequency is divided by 16, but the frequency by 16 is the fastest speed for Pi3.
- When the frequency of the Raspberry Pi 4B uses a divide by 16, the SPI rate is too high and transmission errors will occur. Therefore, the SPI of the Raspberry Pi 4B can only use a divide by 32 at the fastest.
- In the BCM2835 library manual, for different Raspberry Pi versions and different clock divisions, the corresponding frequency description is shown in the following figure:

Enumerator	
BCM2835_SPI_CLOCK_DIVIDER_65536	65536 = 3.814697260kHz on Rpi2, 6.1035156kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_32768	32768 = 7.629394531kHz on Rpi2, 12.20703125kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_16384	16384 = 15.25878906kHz on Rpi2, 24.4140625kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_8192	8192 = 30.51757813kHz on Rpi2, 48.828125kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_4096	4096 = 61.03515625kHz on Rpi2, 97.65625kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_2048	2048 = 122.0703125kHz on Rpi2, 195.3125kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_1024	1024 = 244.140625kHz on Rpi2, 390.625kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_512	512 = 488.28125kHz on Rpi2, 781.25kHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_256	256 = 976.5625kHz on Rpi2, 1.5625MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_128	128 = 1.953125MHz on Rpi2, 3.125MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_64	64 = 3.90625MHz on Rpi2, 6.250MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_32	32 = 7.8125MHz on Rpi2, 12.5MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_16	16 = 15.625MHz on Rpi2, 25MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_8	8 = 31.25MHz on Rpi2, 50MHz on RPI3
BCM2835_SPI_CLOCK_DIVIDER_4	4 = 62.5MHz on Rpi2, 100MHz on RPI3. Dont expect this speed to work reliably.
BCM2835_SPI_CLOCK_DIVIDER_2	2 = 125MHz on Rpi2, 200MHz on RPI3, fastest you can get. Dont expect this speed to work reliably.
BCM2835_SPI_CLOCK_DIVIDER_1	1 = 3.814697260kHz on Rpi2, 6.1035156kHz on RPI3, same as 0/65536

(/wiki/File:20191219173204.png)

- If you need to obtain the most suitable SPI transmission speed, you need to select a different SPI clock divider according to your Raspberry Pi version, as shown in the following program and its notes:

```

bcm2835_spi_begin(); //Start spi interface, set spi pin for the reuse function
bcm2835_spi_setBitOrder(BCM2835_SPI_BIT_ORDER_MSBFIRST); //High first transmission
bcm2835_spi_setDataMode(BCM2835_SPI_MODE0); //spi mode 0

//bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_16); //For RPi 3/3B/3B+
bcm2835_spi_setClockDivider(BCM2835_SPI_CLOCK_DIVIDER_32); //For RPi 4B

/* SPI clock reference Link: */
/*http://www.airspayce.com/mikem/bcm2835/group_constants.html#gaf2e0ca069b8caef24602a02e8a00884e*/

```

## Enhance driving capability

In some cases, due to the long FPC cable and other reasons, the ink screen display will be partially blurred. In this case, you can try to increase the driving capability to effectively solve the problem of the screen display blur.

You can check the demo below:

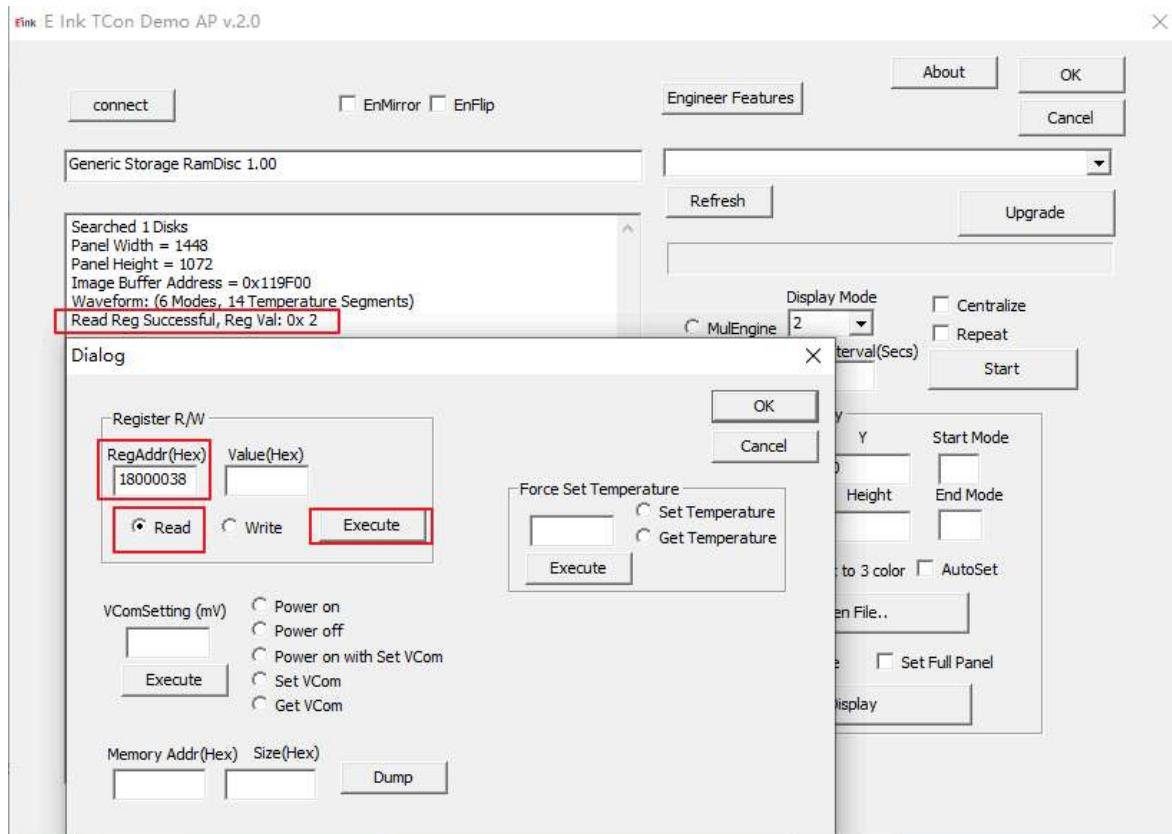
```

#ifndef(Enhance)
    Debug("Attention! Enhanced driving ability, only used when the screen is blurred\r\n");
    Enhance_Driving_Capability();
#endif
//*****************************************************************************
function: Enhanced driving capability
parameter: Enhanced driving capability for IT8951, in case the blurred display effect
//*****************************************************************************
void Enhance_Driving_Capability(void)
{
    UWORD RegValue = EPD_IT8951_ReadReg(0x0038);
    Debug("The reg value before writing is %x\r\n", RegValue);
    EPD_IT8951_WriteReg(0x0038, 0x0602);
    RegValue = EPD_IT8951_ReadReg(0x0038);
    Debug("The reg value after writing is %x\r\n", RegValue);
}

```

If you use the E-LINK-TCON-DEMO software on a PC (Windows) to refresh the ink screen through the USB interface, you can modify the register value in the following ways to enhance the drive capability:

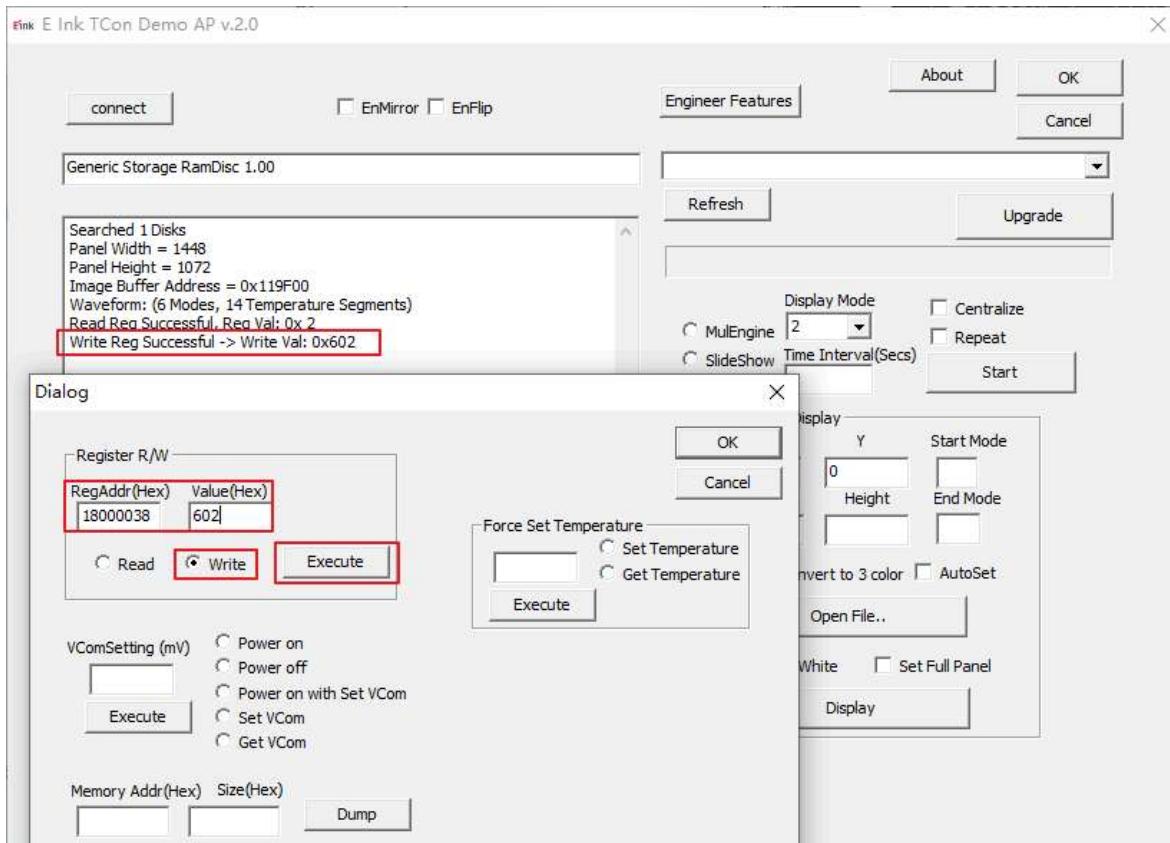
- Step1: Read the data of register address 0 x 18000038.



(/wiki/File:13.3inch\_epaper\_01.png)

If the read data of register address 0x18000038 is 0x02, it means that the drive capability has not been enhanced yet.

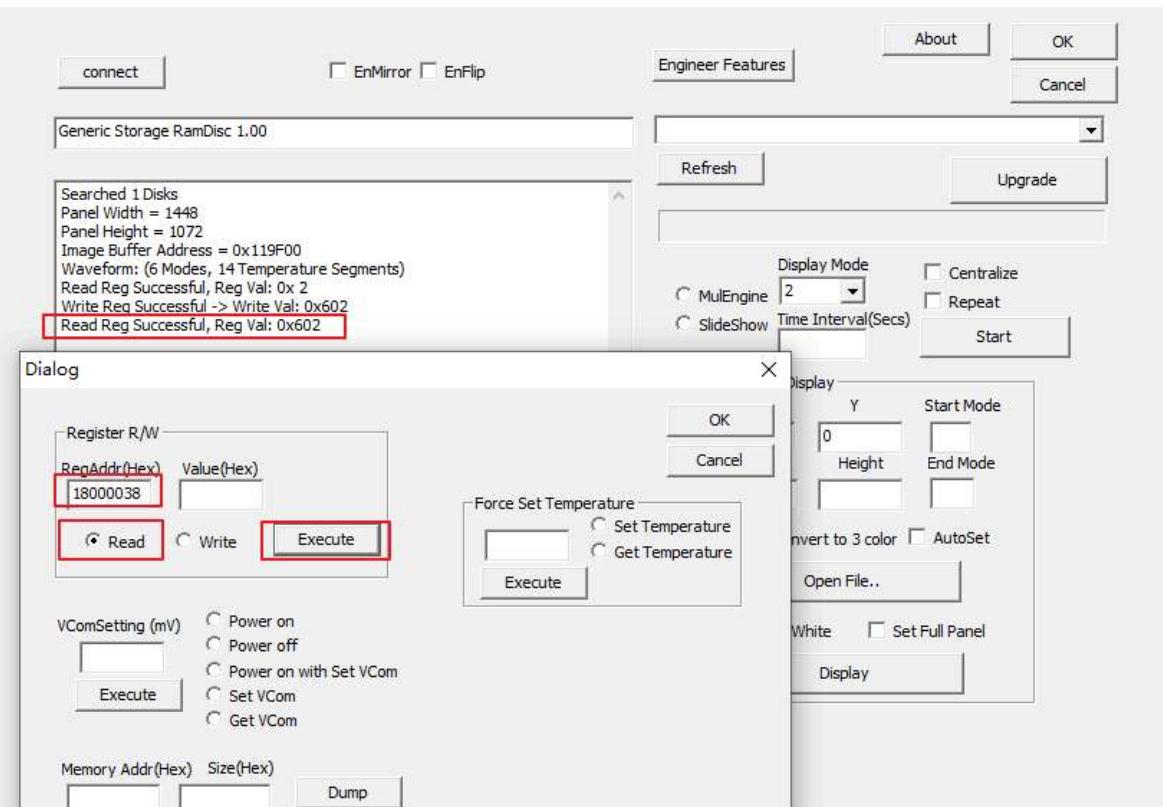
- Step2: Modify the data of register address 0x18000038.



(/wiki/File:13.3inch\_epaper\_02.png)

Modify the data of register address 0x18000038 to 602 to enhance the driving capability.

- Step3: Check the data of register address 0x18000038.



(/wiki/File:13.3inch\_epaper\_03.png)

Check whether the data of the registered address 0 x 18000038 is modified successfully. If the data of this address is 0 x 602, it means that the drive capacity has been enhanced and the screen can be refreshed in this state to avoid the blurred display in some cases, such as when the FPC line is too long and the production batch is different.

## Use the correct VCOM value

---

The VCOM values of each piece of ink screen are different to some extent and the VCOM value of each model of the e-paper is marked on the FPC cable. Make sure that you have used the correct VCOM value when you running the demo. Otherwise, the e-paper will deteriorate by working for a long time under the wrong VCOM value.

## Development Notes

---

The above content only introduces how to use 13.3inch e-Paper HAT. For details on how to modify and perform secondary development, users need to consult the relevant code by themselves.

Please refer to IT8951-related Resources.

# Resources

## Documentation

- Schematic of IT8951 Driver Board (<https://files.waveshare.com/upload/2/27/E-Paper-IT8951-Driver-HAT-B-Schematic.pdf.pdf>)
- Schematic of Adapter board ([https://files.waveshare.com/upload/3/39/13.3inch\\_e-Paper\\_Adapter-sch.pdf](https://files.waveshare.com/upload/3/39/13.3inch_e-Paper_Adapter-sch.pdf))
- E-Paper Floyd-Steinberg ([https://www.waveshare.com/wiki/E-Paper\\_Floyd-Steinberg](https://www.waveshare.com/wiki/E-Paper_Floyd-Steinberg))

## Demo code

- Demo code for Raspberry Pi (New) (<https://github.com/waveshare/IT8951-ePaper>)
- Demo code for STM32 (Open429I) (<https://files.waveshare.com/upload/d/da/Open429I-IT8951-Demo.7z>)
- Github (<https://github.com/waveshare/IT8951-ePaper>)

## Datasheet

- 13.3inch e-Paper Specification (<https://files.waveshare.com/upload/5/5d/13.3inch-e-paper-specification.pdf>)

- IT8951 Specifications ([https://files.waveshare.com/upload/1/18/IT8951\\_D\\_V0.2.4.3\\_20170728.pdf](https://files.waveshare.com/upload/1/18/IT8951_D_V0.2.4.3_20170728.pdf))

## Other documents

- IT8951 I<sub>2</sub>O/SPI/I<sub>2</sub>C Programming Guide ([https://files.waveshare.com/upload/b/b0/IT8951\\_I2O%2BProgrammingGuide\\_16bits\\_20170904\\_v2.7\\_common\\_CXDX.zip](https://files.waveshare.com/upload/b/b0/IT8951_I2O%2BProgrammingGuide_16bits_20170904_v2.7_common_CXDX.zip))
- IT851 USB Programming Guide ([https://files.waveshare.com/upload/c/c9/IT8951\\_USB\\_ProgrammingGuide\\_v.0.4\\_20161114.pdf](https://files.waveshare.com/upload/c/c9/IT8951_USB_ProgrammingGuide_v.0.4_20161114.pdf))
- Mode description (<https://files.waveshare.com/upload/c/c4/E-paper-mode-declaration.pdf>)

## Software

- E-LINK-TCON-DEMO (<https://files.waveshare.com/upload/2/2d/E-LINK-TCON-DEMO.zip>)
- Image2Lcd (<https://files.waveshare.com/upload/b/b3/BmpCvt.zip>)
- CharacterMatrix (<https://files.waveshare.com/upload/c/c6/Zimo221.7z>)

## FAQ

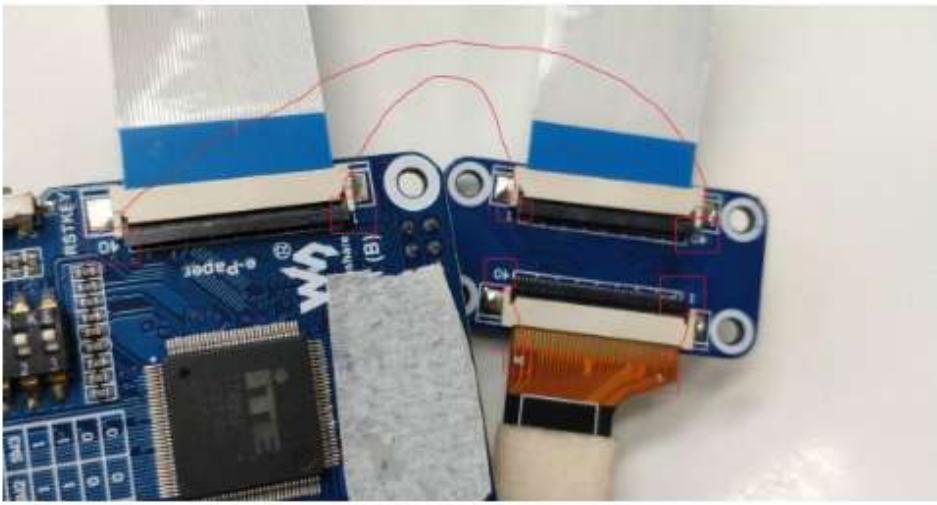
### Questions About Software & Hardware

**Question: Why can't I refresh the picture with E-LINK-TCON-DEMO? How can I check if the wiring is correct?**

**Answer:**

The most likely reason for not being able to print the picture is that the wires are not connected correctly. Please check the wiring. Due to the different batches of products, the wiring method may not be exactly the same as the picture. Please refer to the cable and 1

and 40 on the silk screen. Connect as shown in the red line mark in the figure below:



(/wiki/File:Faqe-

link.png)

**Question:Connect the PC, open the host computer, click connect, and it will prompt Open IT8951 Device Fail. What is the problem?**

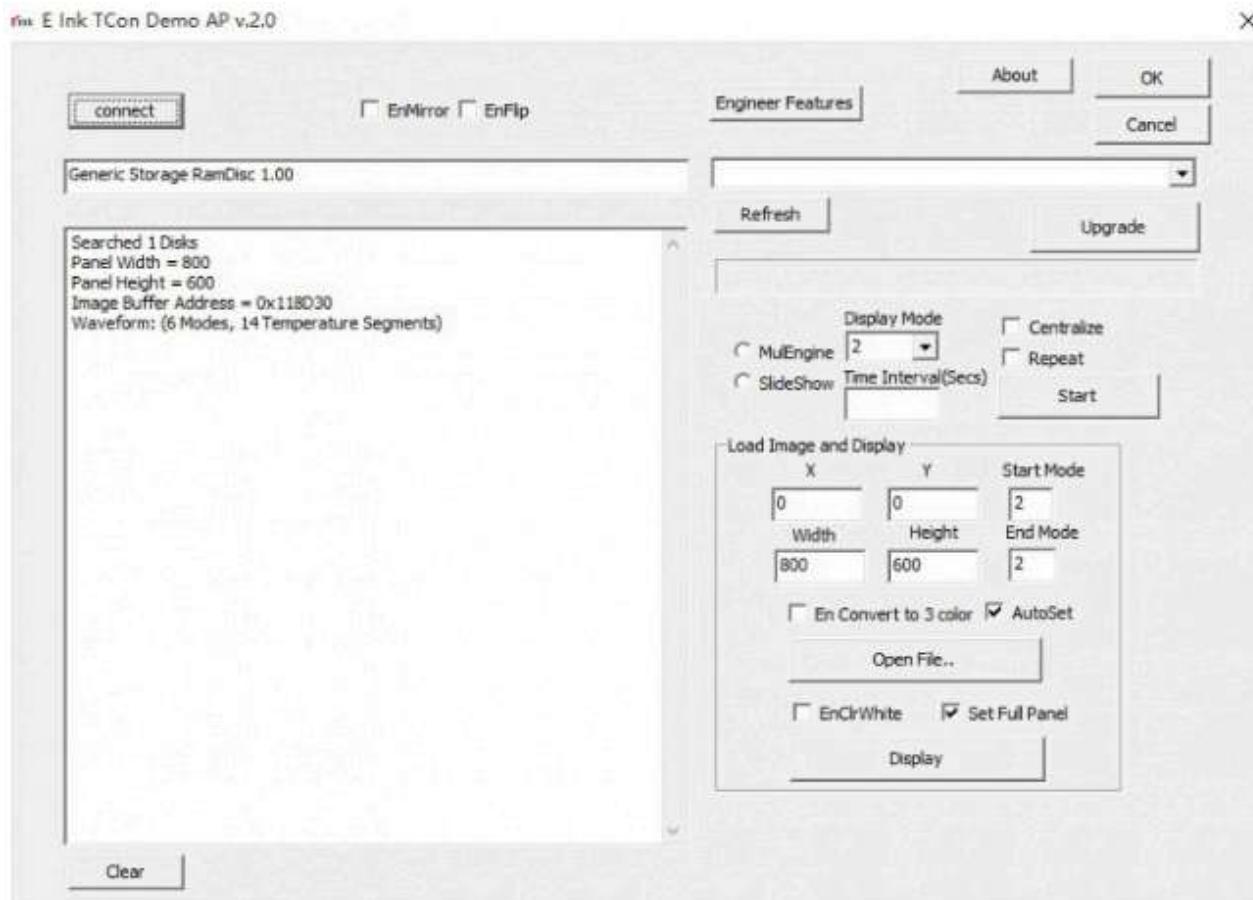
**Answer:**

1. Replace the Micro USB cable with good quality. There are many USB cables on the market that have too much voltage drop, which will cause the USB interface to fail to enumerate normally.
2. Try changing the USB interface. It is recommended to use the USB interface on the back of the PC. Relatively speaking, the power supply current will be larger.
3. Turn the DIP switch to the end. During shipping, the DIP switches may be loosened, and the floating configuration pins cannot put the IT8951 into USB mode.

**Question:Click connect, and the host computer prompts that it is normal. But point refresh doesn't respond?**

**Answer:**

Click connect. The following information will be displayed normally (different models will display different information):



(/wiki/File:Connect.png)

If you fail to refresh, check whether the hard link is normal, use the USB port on the back of the PC (most of the USB ports on the front of the PC have weak power supply capability), and replace the e-paper to test.

Note: Do not plug and unplug the e-paper with power on, otherwise the driver board and e-paper may be damaged.

**Question:** Can two e-Paper screens be connected to a single controller simultaneously?

**Answer:**

Yes, but the CS, RST, and HRDY pins of the second screen need to be connected to different pins on the controller for individual control.

## Questions About Screen

### Question: What is the usage environment of the e-ink screen?

#### Answer:

- **【Operating condition】** Temperature range: 0~50°C; Humidity range: 35%~65%RH
- **【Storage condition】** Temperature range: below 30°C; Humidity range: below 55%RH; Maximum storage time: 6 months
- **【Transportation condition】** Temperature range: -25~70°C; Maximum shipping time: 10 days
- **【After unpacking】** Temperature range: 20°C±5°C; Humidity range: 50±5%RH; Maximum storage time: Assembled within 72 hours

### Question: e-paper screen refresh precautions

#### Answer:

- Refresh mode:
  - Full refresh: The e-paper screen will flicker several times during refreshing (the flicking times depend on the refresh time), and the flicker is to remove the ghosting to achieve the best display effect.
  - Partial refresh: The e-paper screen has no flickering effect during refreshing. Users who use partial flashing should pay attention to performing a full flashing operation to remove the ghosting after refreshing several times, otherwise, the ghosting problem will become more and more serious, or even damage the screen. (At present, only some black and white e-paper screens support partial refreshing, please refer to the product page for details).
- Refresh frequency:
  - 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 partial refreshing function)
  - During the standby process (that is, after the refresh operation), it is recommended that the customer set the e-paper screen to sleep mode, or power off (the power supply part of the e-paper screen can be disconnected with an analog switch) to reduce power consumption and prolong the life of the e-paper screen. (**If some e-paper screens are powered on for a long time, the screen will be damaged beyond repair.**)
  - During the use of the e-paper screen, it is recommended that customers update the display screen at least every 24 hours. (If the screen keeps the same picture for a

long time, there will be a burn-in situation that is difficult to repair).

▪ Application scenarios:

- The e-paper screen is recommended for indoor use. If it is used outdoors, it is necessary to prevent the e-paper screen from being exposed to direct sunlight, and at the same time, it is necessary to take ultraviolet 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 electronic ink screen products, customers should pay attention to determine whether the use environment meets the requirements of electronic ink screens.

### **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 a period of use, 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, otherwise, the screen may burn out when the screen is in a high voltage state for a long time.

### **Question:After the e-paper screen enters deep sleep mode, can it be refreshed again?**

#### **Answer:**

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

### **Question:Is bare screen shipped with a surface coating?**

**Answer:**

Yes.

# Support

## Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket. Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 PM GMT+8  
(Monday to Friday)

[Submit Now \(https://service.waveshare.com/\)](https://service.waveshare.com/)

*Retrieved from "[https://www.waveshare.com/w/index.php?title=13.3inch\\_e-Paper\\_HAT&oldid=83958](https://www.waveshare.com/w/index.php?title=13.3inch_e-Paper_HAT&oldid=83958)" ([https://www.waveshare.com/w/index.php?title=13.3inch\\_e-Paper\\_HAT&oldid=83958](https://www.waveshare.com/w/index.php?title=13.3inch_e-Paper_HAT&oldid=83958))"*