2.13 inch E- Paper Display HAT for Raspberry Pi, SPI interface

250x122, 2.13inch E-Ink display HAT for Raspberry Pi

This is an E-Ink display HAT for Raspberry Pi, 2.13inch, 250x122 resolution, with embedded controller, communicating via SPI interface, supports partial refresh.

Due to the advantages like ultra low power consumption, wide viewing angle, great effect under sunlight, it is an ideal choice for applications such as shelf label, industrial instrument, and so on.

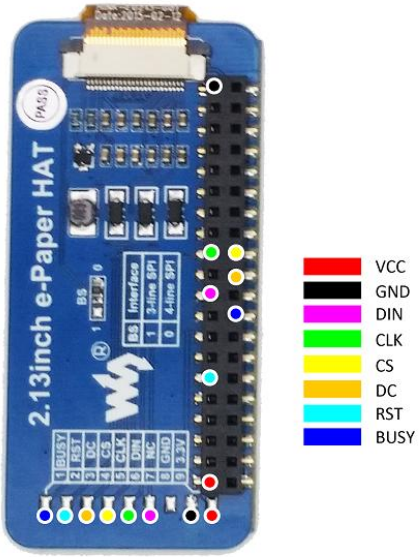
Features

- No backlight, keeps displaying last content for a long time even when power down
- Ultra low power consumption, basically power is only required for refreshing
- Compatible with Raspberry Pi 2B/3B/Zero/Zero W
- Preserved solder pads for SPI interface, for connecting with other controller boards like Arduino/Nucleo, etc.
- Comes with development resources and manual (examples for Raspberry Pi/Arduino/STM32)

Specifications

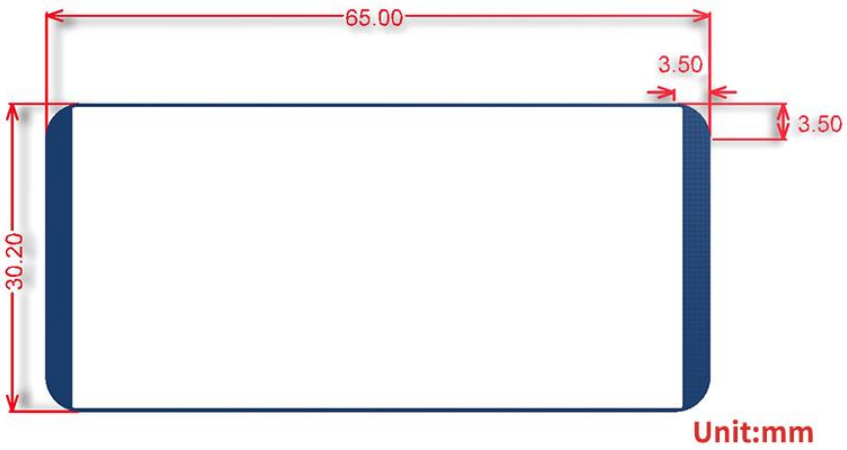
- Operating voltage: 3.3V
- Interface: SPI
- Outline dimension: 65mm × 30.2mm
- Display size: 23.71mm × 48.55mm
- Dot pitch: 0.194 × 0.194
- Resolution: 250 × 122
- Display color: black, white
- Grey level: 2
- Refresh power: 26.4mW(typ.)
- Standby power: <0.017mW
- Viewing angle: >170°

Interface



SYMBOL	DESCRIPTION
VCC	3.3V
GND	Ground
DIN	SPI MOSI pin
CLK	SPI SCK pin
CS	SPI chip selection, low active
DC	Data/Command selection (high for data, low for command)
RST	External reset, low active
BUSY	Busy status output, low active

Dimensions

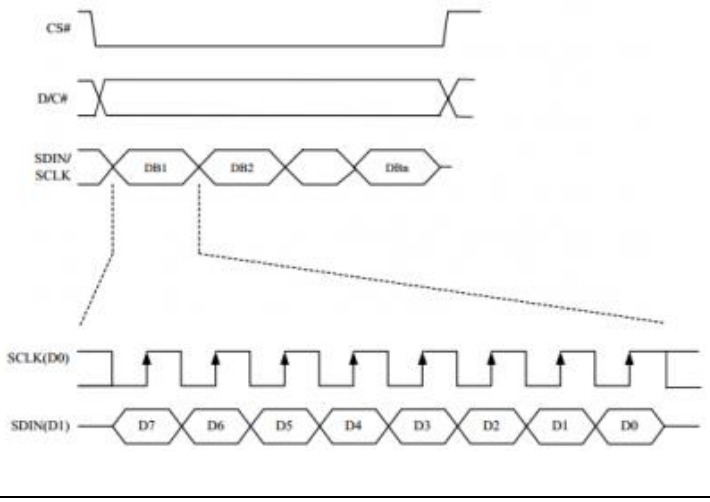


Working principle

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 suspending 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 sunshine, the E-paper screen still has high visibility with a wide viewing angle of 180 degree. It is the ideal choice for E-reading.

Communication protocol

Note: Different from the traditional SPI protocol, the data line from the slave to the master is hidden since the device only has display requirement.



SPI timing

- CS is slave chip select, when CS is low, the chip is enabled.
- DC is data/command control pin, when DC = 0, write command, when DC = 1, write data.
- SCLK is the SPI communication clock.
- SDIN is the data line from the master to the slave in SPI communication.

SPI communication has data transfer timing, which is combined by CPHA and CPOL.

1. CPOL determines the level of the serial synchronous clock at idle state. When CPOL = 0, the level is Low. However, CPOL has little effect to the transmission.
2. CPHA determines whether data is collected at the first clock edge or at the second clock edge of serial synchronous clock; when CPHL = 0, data is collected at the first clock edge.

- There are 4 SPI communication modes. SPI0 is commonly used, in which CPHL = 0, CPOL = 0.

As you can see from the figure above, data transmission starts at the first falling edge of SCLK, and 8 bits of data are transferred in one clock cycle. In here, SPI0 is in used, and data is transferred by bits, MSB first.

How to use

Working with Raspberry Pi

Installing libraries required

If you want to connect your E-paper screen to Raspberry Pi, you should install some necessary libraries, or else the Demo code (click to download) below may work improperly. For more information about how to install the Raspberry Pi libraries, please visit the website: Libraries Installation for RPi.

You can find the detailed presentation about the installations of libraries wiringPi, bcm2835 and python.

Hardware connection

Here is the connection between Raspberry Pi 3B and E-paper.

e-Paper	Raspberry Pi 3B
3.3V	3.3V
GND	GND
DIN	MOSI
CLK	SCLK
CS	CE0
DC	25 (BCM)
RST	17 (BCM)
BUSY	24 (BCM)

Compiling project

After installed the corresponding libraries, you can copy the relative programs into your Raspberry Pi, and then enter the corresponding file.

- BCM2835: Execute the command: make, to compile the code and generate a file main. Execute the command: sudo ./main, the program will run.
- WringPi: Execute the command: make, to compile the code and generate a file main. Execute the command: sudo ./main, the program will run.
- Python: Execute the command: sudo python main.py

Expected result

1. Refresh the whole screen
2. Refresh part of the screen
3. Display the Logo of Waveshare
4. Draw cycles and lines, display two different sizes of characters
5. Display progress bar and time. This function can demonstrate the partial refreshing capability.

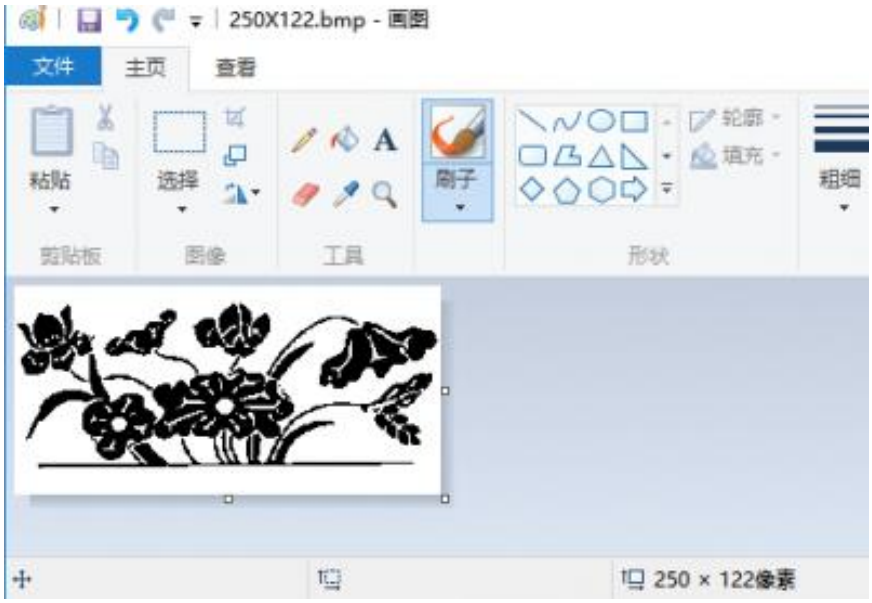
Image making

This module only supports the images with the gray level of two (Black and white). For the image with many gray levels, it cannot display all the colors.

1) Find an image with the gray level of two, and you can make one with a drawing tool on your PC.

2) Modify the image resolution with the drawing tool in your system.

Besides the drawing tool in your system, other graphing tools, such as Photoshop, can be used to modify the image resolution.



3. Use the software Image2Lcd.exe to generate an array for the image (a file with .c extension), as the figure shows below:

Open the image in this software, and set the parameters:

- Output data type: C language array
- Scanning mode: vertical scanning
- Output gray: single color (gray level of two)
- Maximum width and height: the size of pixel
- Inverse color: check

Save the modification, you will get an array for the image in a file with .c extension. Copy the corresponding values into your project, then, you can display the image by calling this array.