**2.9” e-Paper Module User Manual**

**For Raspberry Pi**

**1. Install Function Library**

You need to install necessary function library before the E-Ink display module works for Raspberry Pi, otherwise the following sample program might not work properly.

You can click: <http://www.waveshare.net/wiki/Pioneer600_Datasheets> for the detailed installation of Raspberry Pi Libraries, including wiringPi, bcm2835 and python function library.

1. **Hardware Connection**

Here we use Raspberry Pi 3B as an example.

|  |  |
| --- | --- |
| E-Paper | Raspberry Pi 3B |
| 3.3V | 3.3V |
| GND | GND |
| DIN | MOSI |
| CLK | SCLK |
| CS | CE0 |
| DC | 25 |
| RST | 17 |
| BUSY | 24 |

**3. Expected Results**

1) Install the library, copy the corresponding program into Raspberry Pi and enter the corresponding directory:

* **bcm2835:** Execute the command: make. It will compile the code and create an executable file called epd. Execute the command: sudo. / epd, the program will run.



* **wiringpi:** Execute the command: make. It will compile the code and create an executable file called epd. Execute the command: sudo. / epd, the program will run.



* **python:** Execute the command: sudo python main. py

2) The screen is all refreshed, displaying text and graphics.

3) The screen is partially refreshed, displaying graphics and time. The continuous change of time string shows the function of partial refreshing.

**For Arduino**

**1. Hardware Connection**

Connect E-Paper to UNO:

|  |  |
| --- | --- |
| E-Paper | Arduino |
| 3.3V | 3V3 |
| GND | GND |
| DIN | D11 |
| CLK | D13 |
| CS | D10 |
| DC | D9 |
| RST | D8 |
| BUSY | D7 |

**2. Expected Results**

1) Copy the files of the sample program package in the arduino/libraries directory to documents\ arduino\libraries, which can be specified by Arduino IDE → File → Preferences → Sketchbook location.

2) Click Upload to upload the program.

3) The screen is all refreshed, displaying text and graphics.

4) The screen is partially refreshed, displaying graphics and time. The continuous change of time string shows the function of partial refreshing.

**For STM32 development board**

* The MCU of development board used in this sample program is STM32F103ZE.
* This sample program is based on HAL library, so you can use STM32CubeMX to port the sample program to other STM chip.
* This sample program is compiled and passed in Keil v5 environment.

**1. Hardware Connection**

Connect E-Paper to NUCLEO-F103RB:

|  |  |
| --- | --- |
| E-Paper | NUCLEO-F103RB |
| 3.3V | 3V3 |
| GND | GND |
| NC | -- |
| DIN | PA7 |
| CLK | PA5 |
| CS | PB6 |
| DC | PC7 |
| RST | PA9 |
| BUSY | PA8 |

**2. Expected Results**

1) Open the Keil program (epd-demo.uvprojx) under the MDK-ARM directory.

2) Click Build to compile the program.

3) Click Download to write the program to the chip.

4) The screen is all refreshed, displaying text and graphics.

5) The screen is partially refreshed, displaying graphics and time. The continuous change of time string shows the function of partial refreshing.