Creating a Prototype **E-paper** Design Using **PaPiRus Zero** and **Raspberry Pi Zero W**

PERVASIVE DISPLAYS

Creating a prototype e-paper design using PaPiRus Zero and Raspberry Pi Zero W

By Scott Soong, CEO, Pervasive Displays

The use of e-paper displays (EPDs) is growing at a significant rate. Offering zero-power credentials once the data to be displayed has been transferred, they are a developer's dream come true. With the continued quest for lowering the overall power consumption profile of a design and thereby extending battery life, they are being incorporated into a myriad of applications.

An EPD uses a reflective display technology composed of millions of black and white particle-filled capsules held between two electrodes. A charge applied between the electrodes results in the capsules moving either towards the front electrode or the rear one, depending on the polarity of the charge. Once the charge has been applied, the particles maintain their visibility and display the resulting image. No power source is required to maintain or refresh the charge applied to the electrodes, resulting in a major power saving compared to an LCD display. In addition, the use of a reflective technology means that no backlighting is required, this being a significant power drain for many LCD-based designs. Even when an e-paper display is being updated it draws less power than an LCD display. For example, an EPD might draw just 2 mA compared to 30 mA for an LCD display. It is no wonder that engineers are keen to incorporate an e-paper display into their design.

Like many other trends that drive the global electronics industry, once engineers see one technology becoming popular, many are keen to develop new products, tools and resources that speed up the pace of adoption even further.

A good example of this is the PaPiRus Zero[1] e-paper display module from Pi Supply. Available with either a 1.44 or a 2 inch e-paper display manufactured by Pervasive Displays,[2] the PaPiRus Zero has been designed to mate with the Raspberry Pi Zero Wireless,[3] the 1 GHz single-core compact version of the popular Raspberry Pi single-board computer. Together the compact Raspberry Pi Zero Wireless and PaPiRus Zero's combined dimensions make them ideal for Internet of Things (IoT), wearable and ultra-low power embedded applications.

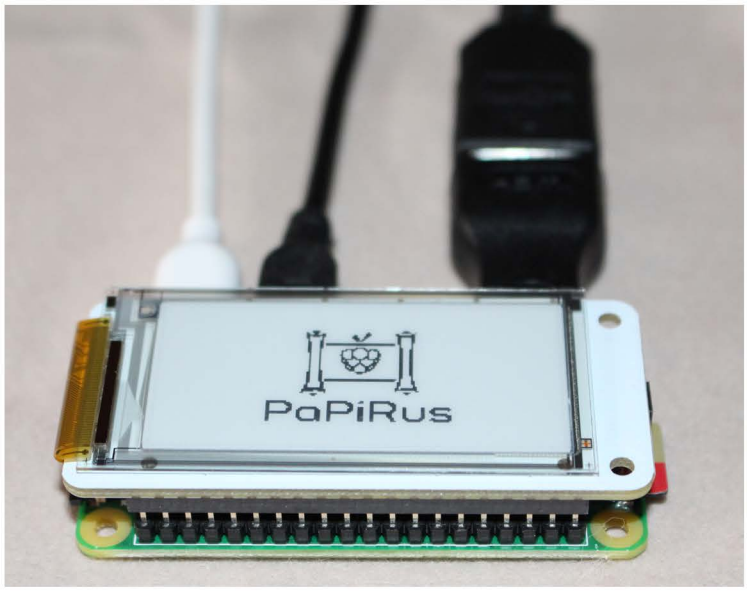

Figure 1 – PaPiRus Zero display module fitted with 2 inch (200 x 96 pixel) display sat on top of Raspberry Pi Zero W (Source: Pervasive Displays)

Figure 1 shows the two units linked together with a 2 inch 200 x 96 pixel e-paper display. The PaPiRus also features five small push buttons along the top edge of the PCB. These are connected to the Pi Zero's GPIO and can be accessed to create menu functions on the e-paper display as part of the end application – see Figure 2.

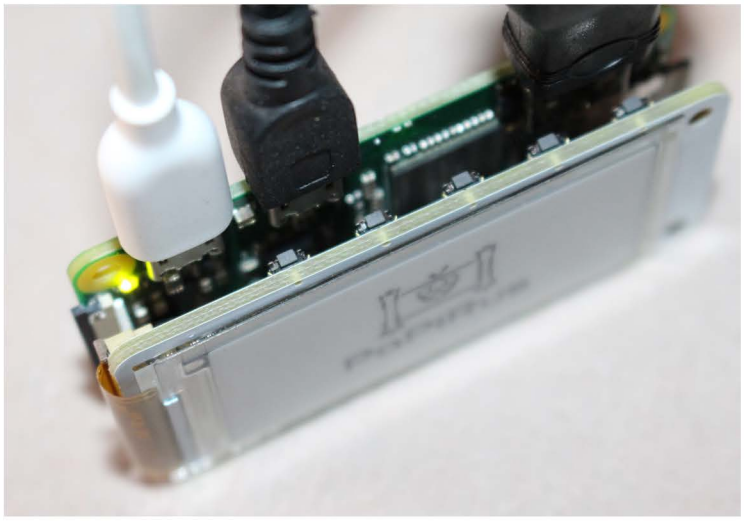

Figure 2 – Five push buttons on top of PaPiRus Zero module (Source: Pervasive Displays)

Creating a prototype e-paper design using these two units is an extremely straightforward process. First it is advisable to get your Raspberry Pi Zero W up and running. The Pi Zero Wireless was launched in March 2017, adding WiFi and Bluetooth wireless capabilities to the initial Pi Zero that was launched as a $5 computer in late 2015. According to the Raspberry Pi Foundation's blog post of May 3 2017, since its launch the Pi Zero W has already shipped 250,000 units.

There are a number of options as to which operating system you use on your Pi Zero W. For this article we opted for Raspbian Jessie[4] with the Pixel desktop. Once downloaded, use a disk image writer such as etcher.io[5] to write the image to a previously formatted 8 GB or larger microSD card. Carefully insert the card into the Pi Zero and connect up the cables as illustrated in Figure 1. The white cable is a 5 V DC power cable from a wall charger, or you can power it from a computer USB port; the black cable next to it is for attaching a USB keyboard and mouse. For the initial set-up it is easier to use the Pi Zero's desktop GUI through the HDMI, although a headless approach is also possible, which is covered later. The HDMI connector is on the far right, and couples through an adapter to a standard-size HDMI cable. Power up the Pi Zero and confirm that everything works correctly. Should you be planning to operate the Pi Zero and PaPiRus as a headless combination, you need to enable SSH on the Pi Zero, since by default it is disabled. This is also the case for using VNC. The easiest way is through the Raspberry Pi configuration tool using the desktop – see Figure 3.

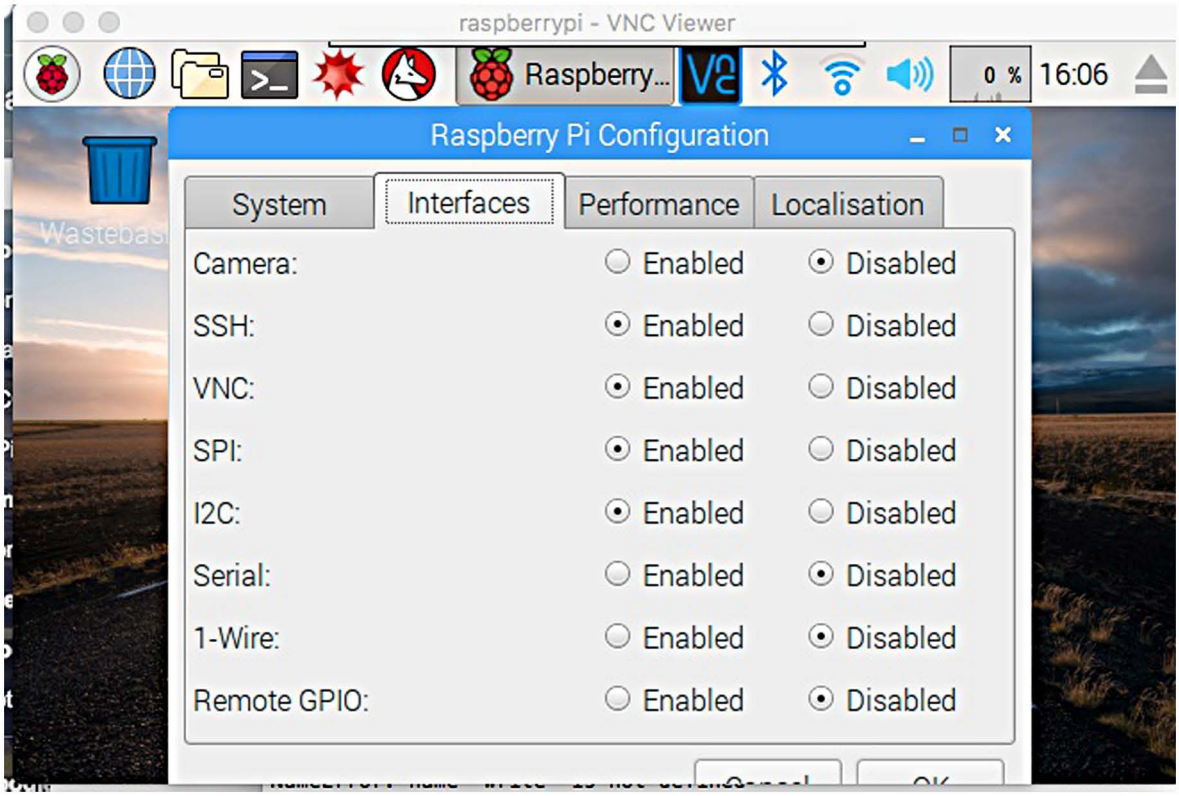

Figure 3 – Raspberry Pi configuration – enabling SSH and VNC

One you are happy that everything is functioning normally with the Pi Zero, power it down and add the PaPiRus e-paper display module. If you've not already done so, this requires the addition of a 40-pin header to be soldered onto the Pi Zero. Insert your selected EPD to the PaPiRus board using the ZIF socket and FPC connector. Take care at this stage, since the FPC cable is very delicate and getting the right alignment into the ZIF socket is crucial. Then mate the PaPiRus with the Pi Zero.

You are now ready to get the PaPiRus Zero ready for use, so power up the Pi Zero. The following steps can be performed headless through SSH or via a terminal window within the Pixel desktop. Detailed instructions can be found at Pi Supply's PaPiRus GitHub[6] page. Using the command

```
curl -sSL https://pisupp.ly/papiruscode | sudo bash
```

automates the installation process and loads a number of command line examples that are explained on GitHub. The installation process can be performed manually, but for pure convenience the automated process is best. During the set-up process the script will ask you to enter the EPD size you are using. The Python source of the command line examples can be found within the bin repository on GitHub.

A list of the executable examples can be found in Figure 4.

```
pi@raspberrypi:~/PaPiRus/bin $ ls
papirus-buttons  papirus-config  papirus-logo.bmp  papirus-temp      papirus-write
papirus-clear    papirus-draw    papirus-set       papirus-textfill
papirus-clock    papirus-gol     papirus-setup     papirus-twitter
pi@raspberrypi:~/PaPiRus/bin $ papirus-clock
panel = EPD 2.0 200 x 96  version=4 COG=2 FILM=231
```

Figure 4 – PaPiRus examples contained within the /home/pi/PaPiRus/bin directory

No design article would be complete without a "Hello World" example, so, using your preferred editor – we use Nano – create a Python code file testEPD.py using the text illustrated in Figure 5.

Figure 5 – PaPiRus Zero "Hello World" example

Write the file out, exit and then execute with the command sudo python testEPD.py

You should now see “Hello World” on your display. Then disconnect the power to the Pi Zero and the text remains displayed – absolutely no power is required by the display.

The PaPiRus GitHub page gives examples of using different fonts and how to display images. Some of the examples accept command line parameters, in this case to display an image such as papirus-draw/home/pi/papirus-logo.bmp

Images for display should be created in a 1-bit bitmap (BMP) file of the dimensions of your display, so for our 2 inch display it is 200 x 96 pixels. Rather than loading different fonts you could, for a more static application, create an image containing the desired text in the fonts required – however, that is a cumbersome process when updates are required.

There are hundreds of potential applications for EPDs using this combination of modules. Consider, for example, using them instead of a written label or caption for pieces of art in an art gallery or exhibition. Once programmed with the title of the work, the artist’s name and the price, they will continue to display the information without any power – see Figure 6.

Figure 6 – PaPiRus Zero and Raspberry Pi Zero W used for art gallery application

This application concept could be developed further with a couple of extra building blocks to ease the process of updating each display, especially in an exhibition where there might be hundreds of units in use. Techniques such as turning off all unrequired peripherals, like the HDMI interface, could be employed to save power, in addition to energy-harvesting techniques. An approach that turns off the Pi Zero W once programmed until text/image changes are required would help. Likewise, using a more convenient and energy-efficient way of communicating with the Pi Zero W other than Bluetooth or WiFi would take this concept into a commercially viable product.

This whitepaper has highlighted the ease with which an e-paper display can be implemented, in addition to showcasing just one of the many hundreds of potential applications.

Links:

1 – https://www.pi-supply.com/product/papirus-zero-epaper-screen-phat-pi-zero/

2 – http://www.pervasivedisplays.com/

3 – https://www.pi-supply.com/product/raspberry-pi-zero-w/

4 – https://www.raspberrypi.org/downloads/raspbian/

5 – https://etcher.io/

6 – https://github.com/PiSupply/PaPiRus