

Notes on PaPIRus

👤 Russell 📅 July 24, 2015 📁 hardware, Raspberry Pi, software 💬 Comments

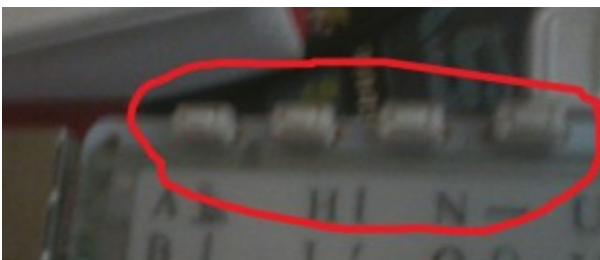


A couple of weeks ago that awfully nice Aaron Shaw of [Pi-Supply](#) Sent me a welfare package of some hardware that I have no ability to obtain for myself. I'd asked if he could bung me a few official Raspberry Pi psu's, A Black Hack3r & a Motor board that he being the nice chap he is did. However, he also included a few PiCrusts, PiSwitch, a Fish Dish (by the way I really do suck at soldering these days 😊)

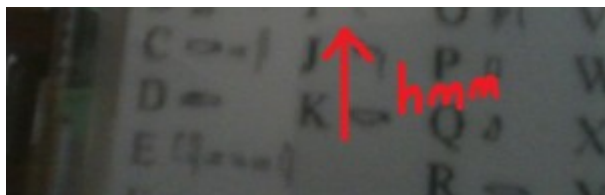


my only excuse is great rustiness (pun not intended 😊), old age (bad eyes, shaky hands) and bad tools, really bad tools, honest it's the tools really. I think not only are they rubbish but they are out to get me too. Not that i'm paranoid or anything. Also included was a PaPiRus ePaper HAT from his recent kickstarter and available on the [Pi-Supply website](#). I had some things I had to do before I could start playing with it so didn't start until yesterday afternoon so really haven't done much with it yet. I do have some ideas but I need to get some other things organised before I can put it all together.

There doesn't seem to be much (Any?) documentation for it yet so here are a few of the notes i've made as i've been playing with it and working out how to control the epaper screen, the rtc and which GPIO the 4 buttons seem to be on. By the way IMHO I think epaper is upsidedown as the buttons are at the "top" of the PCB.



I'm sure there is a way to "rotate" the display but so far I haven't found it.



To control the ePaper part of the PaPiRus you'll if you haven't already done it you'll need to enable spi & preferably devicetree using raspi-config. The options to do that are under Advanced Options. Once you have enabled spi and rebooted you'll need to install the PapiRus driver which you can grab either from <https://github.com/PiSupply/PaPiRus> or from my fork where i've fixed a missing ENVVAR & started tidying up the repo and adding a few bits and pieces <https://github.com/ukscone/PaPiRus> Follow the the installation instructions in the README.md in either of those two repo's and you should be good to go. There are a few demo programs in there although I think the python module isn't being installed correctly but I know sweet fanny adams about python so when I got errors saying that the progs couldn't find the EPD module I used a `sys.path.append()` hack and added the path to the module files just before the `import EPD` (& changed case)

```
'''This is a hack as I don't know why it's not finding the installed papirus module
without it'''
sys.path.append("/usr/local/lib/python2.7/dist-packages/papirus/")
from epd import EPD
```

Although you can use python to control the PaPiRus using python, you can also use it over a FUSE filesystem mounted on (by default) `/dev/epd`

```
pi@glooston ~ $ sudo service epd-fuse start
[ ok ] Starting EPD: epd_fuse.
pi@glooston ~ $ ls /dev/epd
BE current display LE temperature
command current_inverse display_inverse panel version
pi@glooston ~ $
```

File	Read/Write	Description
version	Read Only	The driver version number
panel	Read Only	String describing the panel and giving its pixel width and height
current	Read Only	Binary image that matches the currently

		displayed image (big endian)
display	Read Write	Image being assembled for next display (big endian)
temperature	Read Write	Set this to the current temperature in Celsius
command	Write Only	Execute display operation
BE	Directory	Big endian version of current and display
LE	Directory	Little endian version of current and display
Command	Byte	Description
-----	-----	-----
'C'	0x43	Clear the EPD, set `current` to all zeros, `display` is not affected
'U'	0x55	Erase `current` from EPD, output `display` to EPD, copy display to `current`
'P'	0x50	Partial Update. Only update changed pixels

The above table comes from [Adafruit's page on using the RePaper epaper display](#)

```
e.g. cat /dev/epd/panel
```

```
EPD 2.7 264x176 COG 2 FILM 231
```

```
echo 'C' >/dev/epd/command
```

to clear the display.

You can also find examples on how to use the ePaper display in [RePaper's github repo](#).

I've not confirmed this yet but the 4 buttons on the PCB seem to be on ~~GPIO.26 through GPIO.29 inclusive~~. BCM.16,BCM.26,BCM.20,BCM.21 (GPIO.25,27,28,29) I'm sure someone will confirm that eventually (unless I start fiddling with them over the next few days)

The PaPiRus has a battery backedup RTC and after a bit of time digging around (ok trying to find a combination of magnifying glasses that would let me read the teeny weeny insy wincy writing on all the ICs & a bit of googling I found out that the RTC chip is an NXP 8523T aka PCF8523 or as near as damn it.

To talk to the RTC chip you'll need to enable I2C, again easiest way is to do it in raspi-config. and although it seems a bit weird to have to do it still with devicetree you'll need to also modprobe (or add to /etc/modules) i2c-dev to get /dev/i2c-1 created. you'll also want to modprobe rtc-pcf8523 (or add to /etc/modules) and finally

```
sudo -s  
echo pcf8523 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

and can now use `hwclock -r` to read the time or `hwclock -w` or `hwclock -s` to set the rtc from the `systemclock` or set the `systemclock` from `hwclock`.

The first time you try to read the rtc when it is still in factory state you'll get an error using `hwclock -r` until you've set it using `hwclock -w` (you might have to do `hwclock -w` twice to get it to work, at least I had to). Obviously you'll want all that to be set to happen on boot but i haven't got around to that yet as i'm still fiddling.

I'm sure there is a lot more I haven't discovered yet but I will eventually and this should do as a good starting point for others.

Finally. The PaPiRis is a nice but of kit although it is a bit irksome I really haven't found a nice case for the Pi it's currently on as the 2.7" display is a bit close to the closures in the official case & I had to remove the top layer on a Pimoroni Pibow B+ Coupe case. I'm sure i'll find a nice case for it eventually.

