



Adafruit DPI Display Kippah

Created by lady ada



Last updated on 2019-02-05 10:23:22 PM UTC

Guide Contents

Guide Contents	2
Overview	3
Installation	7
Connect Display	7
Update & Upgrade	8
Install and Try raspi-gpio	10
Install Device Tree Blob	11
Update configuration	11
Touch screen support	13
Backlight	16
Downloads	18
Files	18
Schematic	18
Fabrication print	18
F.A.Q.	20
Can I use the DPI interface and HDMI or TV out at the same time?	20
How is that DTO blob made?	20

Overview



A TFT panel connected to a Raspberry Pi without the use of an HDMI decoder? What is this sorcery??? It's the DPI Kippah from Adafruit! This HAT-like* board snaps onto a Raspberry Pi B+, A+, Pi 2, Pi 3 or Zero and with a little software configuration, allows you to have what *normally* would go out the HDMI port come up on a nice little flat screen. * *Its not technically a HAT due to the lack of an on-board EEPROM, but its the same shape as a Pi HAT and its a covering of sorts, so we call it a kippah*

Compared to our lovely HDMI backpacks, you don't have the extra cost or expense of an HDMI encoder/decoder. And you get a nice ultra-fast 18-bit color display with optional touch support. We tested it and it works great with our 5" and 7" displays at 800x480. This display is 'native' so it gets all the graphics acceleration capabilities, instant refresh, etc. you would get from an HDMI display

The Kippah will also generate the backlight drive voltage (up to ~24V) from the 5V USB power on the Raspberry Pi

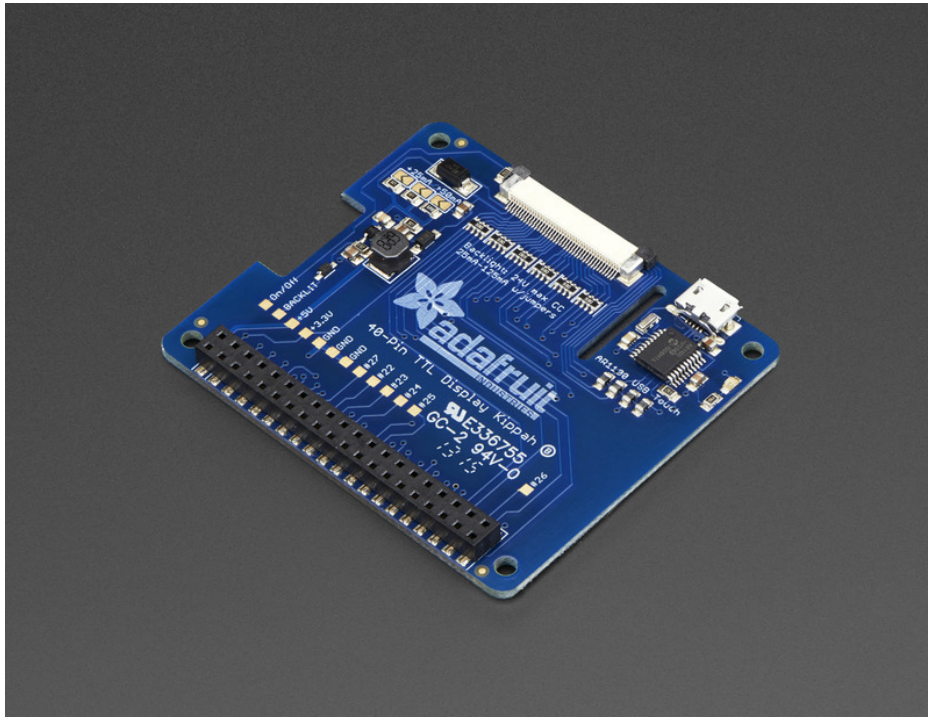


OK so, exciting right? But, what's the catch? The catch is **this add on board uses nearly every pin available on the Raspberry Pi and those pins are hardcoded, they cannot be moved or rearranged.** The pins used are GPIO 2 through 21 inclusive. That means you don't get the UART RX/TX pins (no console cable) and you don't get the standard user I2C pins, the EEPROM I2C pins, or hardware SPI pins. You do get to use pins #22, #23, #24, #25, #26 and #27, and the USB ports are fine to use too.

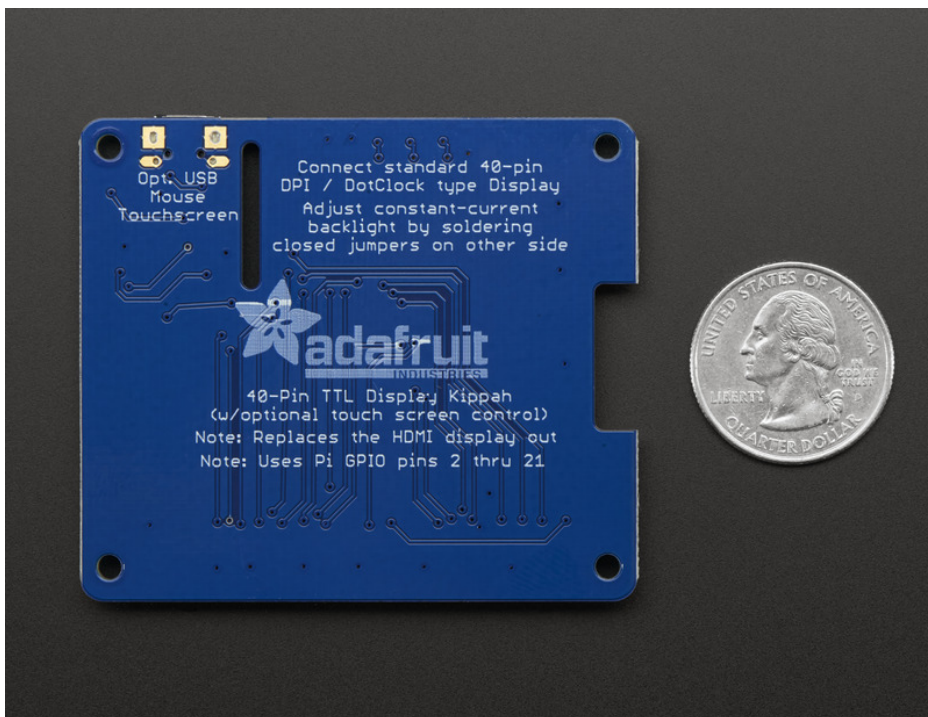
The other catch is that this display replaces the HDMI/NTSC output, so you can't have the DPI HAT and HDMI working at once, nor can you 'flip' between the two.

Also, there's no PWM's available so you can't have precision backlight control unless you somehow rig up an external PWM generator with a 555 or something.

Finally, we did test this setup with a straight-up Raspbian and after the software installs, it works great. However, we don't guarantee it will work with any other Raspberry Pi operating system or setup.



That said, if you don't need a bunch of GPIO, it's very easy to add a high quality display. Pick and choose whether you want a touch-screen or not, then choose the size of the display - 5" or 7" is best. You can also grab an FPC extension board and extend the display away from the Pi. For power-users, this is a very nice little accessory



Comes as one fully assembled and tested DPI Kippah circuit board. You may need a soldering iron to adjust the backlight by soldering closed a PCB jumper (check the tutorial on this). [TFT display \(https://adafru.it/Cfx\)](https://adafru.it/Cfx), [USB micro-B cable \(http://adafru.it/592\)](http://adafru.it/592), [FPC extension cable \(https://adafru.it/eVZ\)](https://adafru.it/eVZ), [bent wire stand \(https://adafru.it/d7M\)](https://adafru.it/d7M), and **Raspberry Pi not included** (but we do carry them in the shop!)



Installation

If you just plug in the DPI Kippah, it won't work on a fresh installation of Raspbian! You must set up the special device tree overlay configuration!

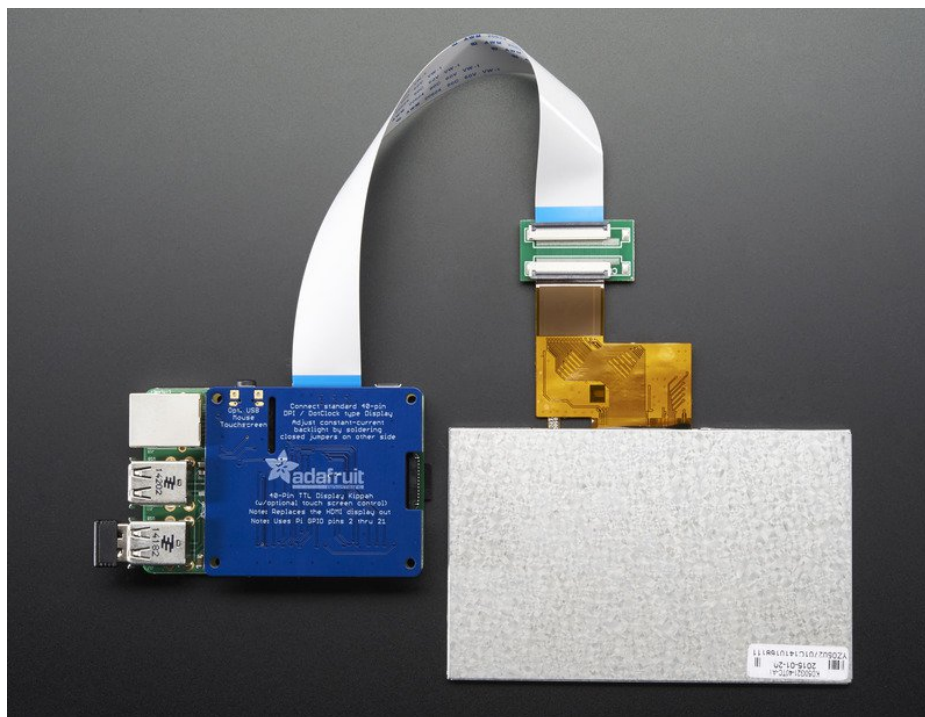
However, its not too bad, check it out below!

We've only tested this device tree overlay/firmware with Raspbian. Since its a bit of a hack, it doesn't work with the native pre-boot 'Noobs' screen. However, it does come up immediately with Raspbian (e.g. you get to see the rainbow square screen)

Note these instructions are tested to work with Raspbian Jessie and Wheezy as of 2016/04/11

Connect Display

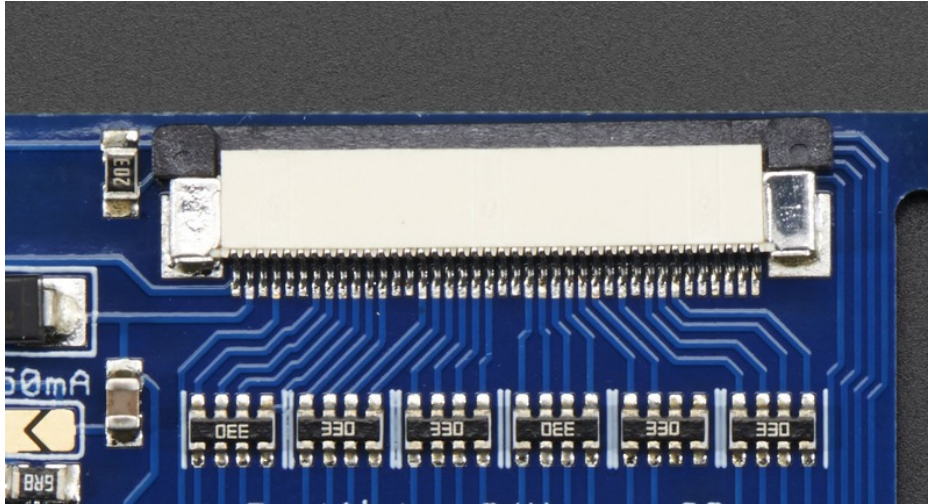
The DPI Kippah has a 40-pin TFT connector. This is a semi-standard connector. A majority of 3.5", 4.3", 5.0" and 7.0" dot-clock DPI displays have this 'standard 40-pin' connector.



Check the Downloads page for an example datasheet so you can check if your display is compatible. All Adafruit 40-pin TFT TTL displays work, we do not guarantee any other displays work.

In fact, if you connect a display that does not match the right pinout, you could easily fry the display if the 20V backlight pin ends up connected to a logic pin!

To connect, gently pull on the two black 'ears' on the FPC connector, and plug in the display so the gold/silver metallic pins are facing 'up' away from the PCB.



Update & Upgrade

Start by updating and upgrading your Raspberry Pi to the latest software.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

A screenshot of a terminal window titled 'COM12 - PuTTY'. The terminal shows the output of the command 'sudo apt-get update'. The output lists various package sources and their release information, including 'stable Release.gpg', 'wheezy Release.gpg', and 'wheezy Release'. It also shows the installation of packages and the translation of some packages. The terminal ends with a green progress bar indicating that 8 packages are being upgraded, totaling 3,196 kB out of 6,902 kB, which is 46% complete.

```
pi@photopi:~$ sudo apt-get update
Get:1 http://repository.wolfram.com stable Release.gpg [836 B]
Hit http://repository.wolfram.com stable Release
Get:2 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Get:3 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]
Get:4 http://archive.raspberrypi.org wheezy Release.gpg [490 B]
Get:5 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Get:6 http://raspberrypi.collabora.com wheezy Release [7,514 B]
Get:7 http://archive.raspberrypi.org wheezy Release [10.2 kB]
Hit http://repository.wolfram.com stable/non-free armhf Packages
Get:8 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,902 kB]
Get:9 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,214 B]
Get:10 http://archive.raspberrypi.org wheezy/main armhf Packages [118 kB]
Ign http://repository.wolfram.com stable/non-free Translation-en_GB
Ign http://repository.wolfram.com stable/non-free Translation-en
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
47% [8 Packages 3,196 kB/6,902 kB 46%]
```



```
COM12 - PuTTY
pi@photopi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages have been kept back:
 fake-hwclock libfm-data libfreetype6 libfreetype6-dev libgail-3-0 libgcc1
 libgfortran3 libgomp1 libgtk-3-0 libgtk-3-bin libgtk-3-common
 libjavascriptcoregtk-3.0-0 libpango1.0-0 libstdc++6 libwebkitgtk-3.0-0
 lxpanel omxplayer pcmanfm wolfram-engine zenity zenity-common
The following packages will be upgraded:
 apt apt-utils base-files bash bind9-host binutils ca-certificates cpio
 cups-bsd cups-client cups-common curl dbus dbus-x11 dpkg dpkg-dev e2fslibs
 e2fsprogs file firmware-brcm80211 gir1.2-glib-2.0 glib-networking
 glib-networking-common glib-networking-services gnome-themes-standard-data
 gnuPG gpgv krb5-locales libapt-inst1.5 libapt-pkg-dev libapt-pkg4.12
 libarchive12 libatk1.0-0 libatk1.0-data libavcodec53 libavformat53
 libavutil51 libbind9-80 libc-bin libc-dev-bin libc6 libc6-dev
 libcairo-gobject2 libcairo2 libcomerr2 libcups2 libcupsimage2 libcurl3
 libcurl3-gnutls libdbus-1-3 libdns88 libdpkg-perl libevent-2.0-5 libflac8
 libgcrpy11 libgd2-xpm libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common
 libgirepository-1.0-1 libglib2.0-0 libglib2.0-data libgnutls26
 libgssapi-krb5-2 libicu48 libisc84 libisc80 libiscfg82 libjasper1
 libjpeg8 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 libldap-2.4-2
 liblwpres80 libmagic1 liborc-0.4-0 libpcre3 libpixman-1-0 libpostproc52
```

We want to get the most recent kernel and firmware, so run

```
sudo apt-get install rpi-update
```

and then

```
sudo rpi-update
```

```
COM33 - PuTTY
Selecting previously unselected package rpi-update.
(Reading database ... 30369 files and directories currently installed.)
Preparing to unpack ../rpi-update_20140705_all.deb ...
Unpacking rpi-update (20140705) ...
Setting up rpi-update (20140705) ...
pi@raspberrypi:~$ sudo rpi-update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** Performing self-update
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Dload  Upload   Total   Spent    Left   Speed
100 10670  100 10670    0     0 36299    0  --:--:--  --:--:--  --:--:-- 36416
*** Relaunching after update
*** Raspberry Pi firmware updater by Hexxeh, enhanced by AndrewS and Dom
*** We're running for the first time
*** Backing up files (this will take a few minutes)
*** Backing up firmware
*** Backing up modules 4.1.19-v7+
#####
This update bumps to rpi-4.1.y linux tree
Be aware there could be compatibility issues with some drivers
Discussion here:
https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=113753
#####
*** Downloading specific firmware revision (this will take a few minutes)
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Dload  Upload   Total   Spent    Left   Speed
100  168    0  168    0     0   295    0  --:--:--  --:--:--  --:--:--   295
100 49.4M  100 49.4M    0     0 1105k    0  0:00:45  0:00:45  --:--:-- 1614k
*** Updating firmware
*** Updating kernel modules
```

Then reboot with

```
sudo reboot
```

```
COM33 - PuTTY
#####
*** Downloading specific firmware revision (this will take a few minutes)
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total   Spent    Left    Speed
100 168    0 168    0    0    640    0  --:--:--  --:--:--  --:--:--  643
100 49.4M 100 49.4M    0    0 1432k    0  0:00:35  0:00:35  --:--:--  850k
*** Updating firmware
*** Updating kernel modules
*** depmod 4.1.21+
*** depmod 4.1.21-v7+
*** Updating VideoCore libraries
*** Using HardFP libraries
*** Updating SDK
*** Running ldconfig
*** Storing current firmware revision
*** Deleting downloaded files
*** Syncing changes to disk
*** If no errors appeared, your firmware was successfully updated to 4bc6b67905
e3206f8cf4d9a316f3343aaaf14d62
*** A reboot is needed to activate the new firmware
pi@raspberrypi:~$ sudo reboot
[ 1012.463087] reboot: Restarting system
```

Install and Try raspi-gpio

To help us debug/make sure we have the right device tree blob, we'll use a tool called `raspi-gpio`

Install it with

```
sudo apt-get install raspi-gpio
```

Then you can run it with

```
sudo raspi-gpio get
```

When you run it this time, you'll see the first 'bank' of GPIO pins set to, essentially

```
GPIO nn: level=1 fsel=0 alt= func=INPUT
```

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ sudo apt-get install raspi-gpio  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  raspi-gpio  
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.  
Need to get 8,210 B of archives.  
After this operation, 47.1 kB of additional disk space will be used.  
Get:1 http://archive.raspberrypi.org/debian/ jessie/main raspi-gpio 0.20150914  
Fetched 8,210 B in 0s (11.4 kB/s)  
Selecting previously unselected package raspi-gpio.  
(Reading database ... 30369 files and directories currently installed.)  
Preparing to unpack .../raspi-gpio_0.20150914_armhf.deb ...  
Unpacking raspi-gpio (0.20150914) ...  
Setting up raspi-gpio (0.20150914) ...  
pi@raspberrypi:~$ sudo raspi-gpio get  
BANK0 (GPIO 0 to 27):  
GPIO 00: level=1 fsel=0 alt= func=INPUT  
GPIO 01: level=1 fsel=0 alt= func=INPUT  
GPIO 02: level=1 fsel=0 alt= func=INPUT  
GPIO 03: level=1 fsel=0 alt= func=INPUT  
GPIO 04: level=1 fsel=0 alt= func=INPUT  
GPIO 05: level=1 fsel=0 alt= func=INPUT  
GPIO 06: level=1 fsel=0 alt= func=INPUT  
GPIO 07: level=1 fsel=0 alt= func=INPUT  
GPIO 08: level=1 fsel=0 alt= func=INPUT  
GPIO 09: level=0 fsel=0 alt= func=INPUT  
GPIO 10: level=0 fsel=0 alt= func=INPUT  
GPIO 11: level=0 fsel=0 alt= func=INPUT  
GPIO 12: level=0 fsel=0 alt= func=INPUT  
GPIO 13: level=0 fsel=0 alt= func=INPUT
```

Install Device Tree Blob

Now download and install the DPI device tree blob ([tip o' the hat to aBUGsworstnightmare \(https://adafru.it/mFi\)](https://adafru.it/mFi)) from github by running:

```
cd ~  
wget https://raw.githubusercontent.com/adafruit/Adafruit-DPI-Kippah/master/dt-blob.bin  
sudo cp dt-blob.bin /boot/
```

in your Pi's command line, to change directories to the home directory, download the blob, then copy it in `/boot`

```
pi@raspberrypi: ~  
pi@raspberrypi:~$ wget https://raw.githubusercontent.com/adafruit/Adafruit-DPI-Kippah/master/dt-blob.bin  
--2016-04-12 00:09:20-- https://raw.githubusercontent.com/adafruit/Adafruit-DPI-Kippah/master/dt-blob.bin  
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 23.235.39.133  
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|23.235.39.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 52314 (51K) [application/octet-stream]  
Saving to: 'dt-blob.bin'  
  
dt-blob.bin      100%[=====>] 51.09K  --.-KB/s  in 0.02s  
  
2016-04-12 00:09:20 (3.14 MB/s) - 'dt-blob.bin' saved [52314/52314]  
pi@raspberrypi:~$ sudo cp dt-blob.bin /boot  
pi@raspberrypi:~$
```

Update configuration

Finally, we'll tell the Pi to use the attached DPI display. The following will work for our 5" and 7" 800x480 displays. Both touch and non-touch displays use the same setup here

Start by editing with

```
sudo nano /boot/config.txt
```

and add the following lines at the bottom

```
# Disable spi and i2c, we need these pins.
dtparam=spi=off
dtparam=i2c_arm=off

# Set screen size and any overscan required
overscan_left=0
overscan_right=0
overscan_top=0
overscan_bottom=0
framebuffer_width=800
framebuffer_height=480

# enable the DPI display
enable_dpi_lcd=1
display_default_lcd=1

# set up the size to 800x480
dpi_group=2
dpi_mode=87

# set up the hsync/vsync/clock polarity and format
dpi_output_format=454661

# set up the size to 800x480
hdmi_timings=800 0 40 48 88 480 0 13 3 32 0 0 0 60 0 32000000 6
```

For 4.3" TFT use the following:

```
# Disable spi and i2c, we need these pins.
dtparam=spi=off
dtparam=i2c_arm=off

# Set screen size and any overscan required
overscan_left=0
overscan_right=0
overscan_top=0
overscan_bottom=0
framebuffer_width=480
framebuffer_height=272

# enable the DPI display
enable_dpi_lcd=1
display_default_lcd=1

# set up the size to 480x272
dpi_group=2
dpi_mode=87

# set up the hsync/vsync/clock polarity and format
dpi_output_format=520197

# set up the size to 480x272
hdmi_timings=480 0 40 48 88 272 0 13 3 32 0 0 0 60 0 32000000 3
```

This sets up the screen, if you ever want to temporarily 'undo the DPI Hat install' just delete these lines

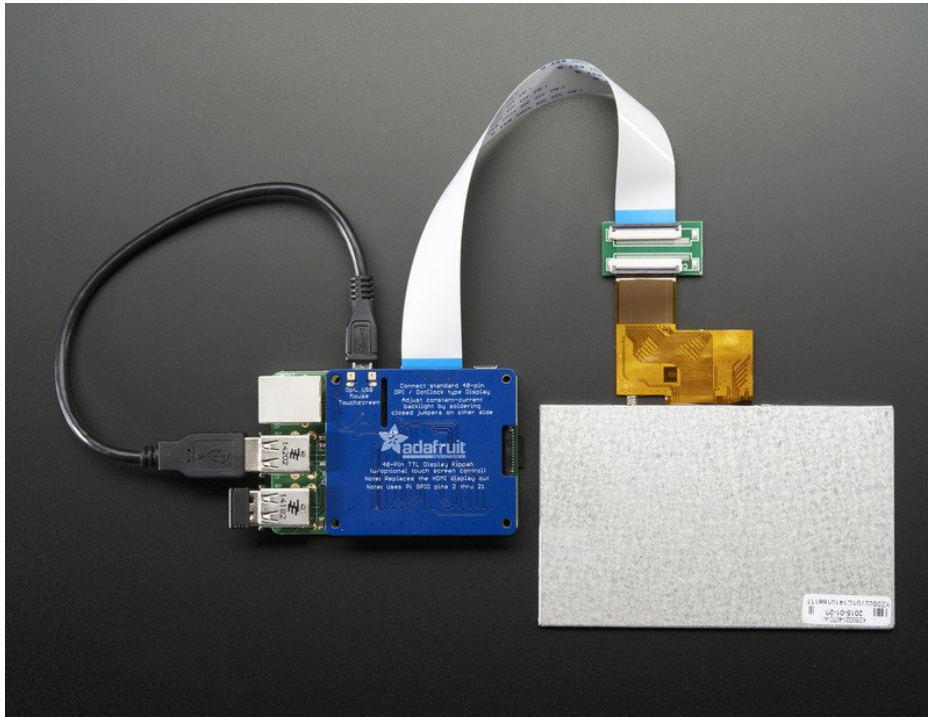
```
enable_dpi_lcd=1
display_default_lcd=1
```

To finish installation, just run **sudo reboot**

Touch screen support

If you have a DPI HAT with touchscreen circuitry installed *and* a touch-screen display, you can easily use it for touch screen support

A microUSB cable is required (not included) connect it from the MicroUSB connector on the HAT into one of the Pi's USB port



No drivers are required! However, you'll likely want to calibrate the screen. We have a calibration helper python script. Start by installing **python-pip** and **pyusb** version **1.0.0b1**

```
sudo apt-get install python-pip
sudo pip install pyusb==1.0.0b1
```

Then grab the code and example gradient

```
cd ~
wget http://adafru.it/ar1100py
mv ar1100py ar1100.py
wget http://adafruit-download.s3.amazonaws.com/gradient800x480.jpg
```

If you are running it on a 5" display, continue as is.

If you are running it on a 7" display, edit with **nano ar1100.py** and change this line:

```
writeeprom = CALIBRATED_5IN_800x480;
```

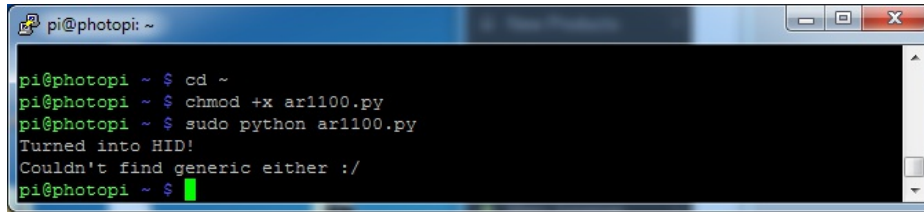
to

```
writeeprom = CALIBRATED_7IN_800x480;
```

Then run the calibrator with:

```
cd ~
chmod +x ar1100.py
sudo python ar1100.py
```

Its normal for the first time you run it, it will complain "Couldn't find generic either" just run it again!

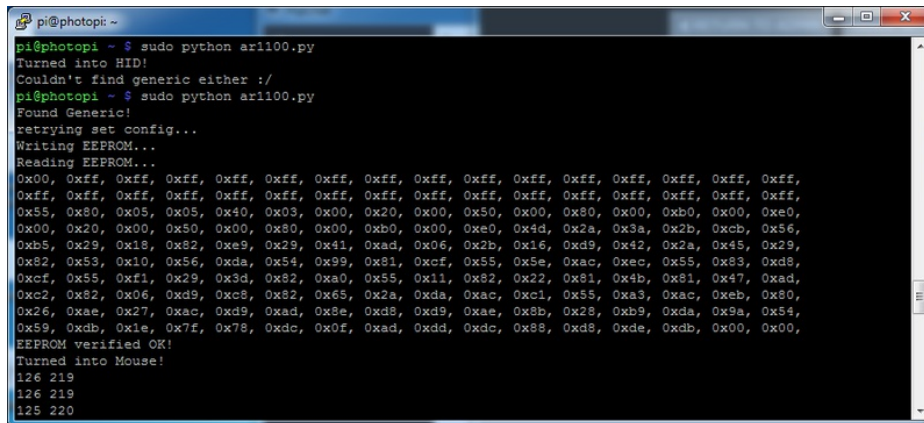


```
pi@photopi: ~  
pi@photopi ~ $ cd ~  
pi@photopi ~ $ chmod +x ar1100.py  
pi@photopi ~ $ sudo python ar1100.py  
Turned into HID!  
Couldn't find generic either :/  
pi@photopi ~ $
```

Just hit the up arrow on your keyboard and return, to rerun `sudo python ar1100.py`

This time it will continue, program the AR1100 with the calibration data, and give you a rainbow display.

When done, hit Escape on your keyboard



```
pi@photopi: ~  
pi@photopi ~ $ sudo python ar1100.py  
Turned into HID!  
Couldn't find generic either :/  
pi@photopi ~ $ sudo python ar1100.py  
Found Generic!  
retrying set config...  
Writing EEPROM...  
Reading EEPROM...  
0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,  
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,  
0x55, 0x80, 0x05, 0x05, 0x40, 0x03, 0x00, 0x20, 0x00, 0x50, 0x00, 0x80, 0x00, 0xb0, 0x00, 0xe0,  
0x00, 0x20, 0x00, 0x50, 0x00, 0x80, 0x00, 0xb0, 0x00, 0xe0, 0x4d, 0x2a, 0x3a, 0x2b, 0xcb, 0x56,  
0xb5, 0x29, 0x18, 0x82, 0xe9, 0x29, 0x41, 0xad, 0x06, 0x2b, 0x16, 0xd9, 0x42, 0x2a, 0x45, 0x29,  
0x82, 0x53, 0x10, 0x56, 0xda, 0x54, 0x99, 0x81, 0xcf, 0x55, 0x5e, 0xac, 0xec, 0x55, 0x83, 0xd8,  
0xcf, 0x55, 0xf1, 0x29, 0x3d, 0x82, 0xa0, 0x55, 0x11, 0x82, 0x22, 0x81, 0x4b, 0x81, 0x47, 0xad,  
0xc2, 0x82, 0x06, 0xd9, 0xc8, 0x82, 0x65, 0x2a, 0xda, 0xac, 0xc1, 0x55, 0xa3, 0xac, 0xeb, 0x80,  
0x26, 0xae, 0x27, 0xac, 0xd9, 0xad, 0x8e, 0xd8, 0xd9, 0xae, 0x8b, 0x28, 0xb9, 0xda, 0x9a, 0x54,  
0x59, 0xdb, 0x1e, 0x7f, 0x78, 0xdc, 0x0f, 0xad, 0xdd, 0xdc, 0x88, 0xd8, 0xde, 0xdb, 0x00, 0x00,  
EEPROM verified OK!  
Turned into Mouse!  
126 219  
126 219  
125 220
```

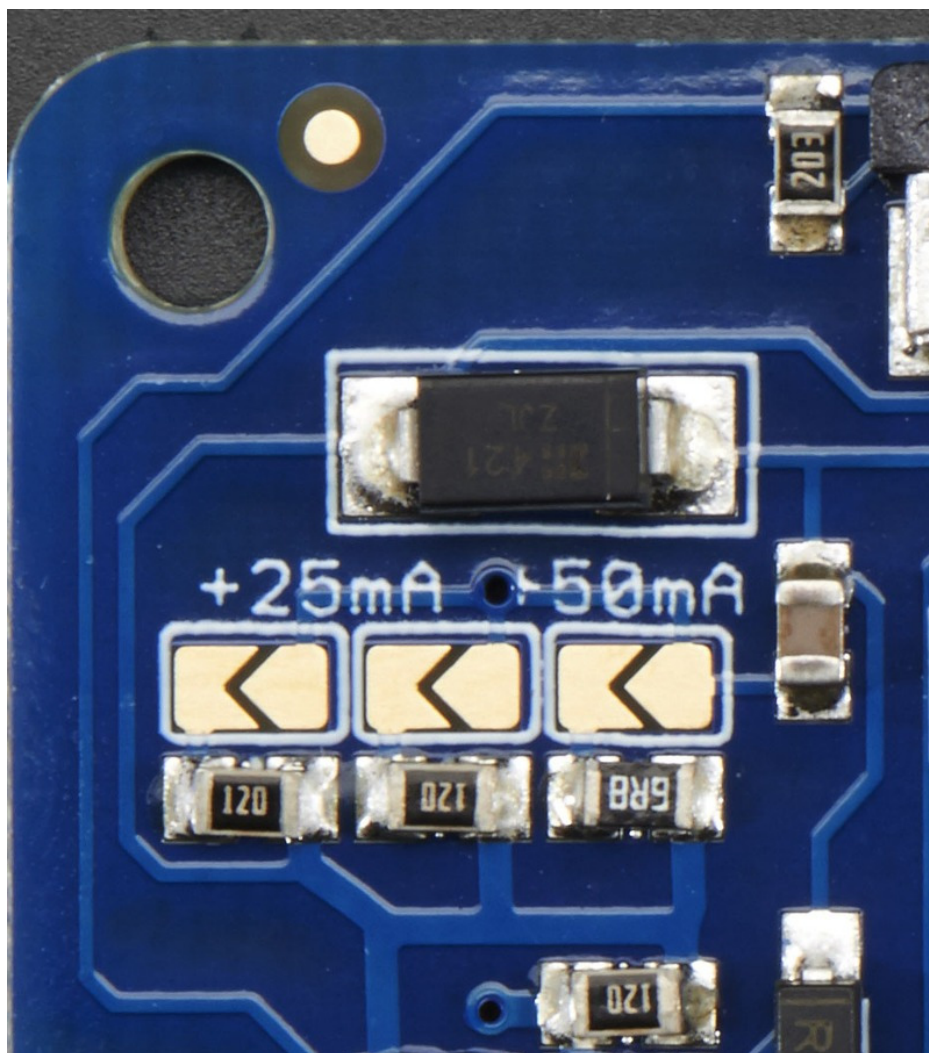
Backlight

The DPI Kippah has a backlight booster that can support any 40pin TFT display we carry. However, since backlights do vary from display to display, you may need to configure the backlight to match your display. Here's the constant-current draws of our displays

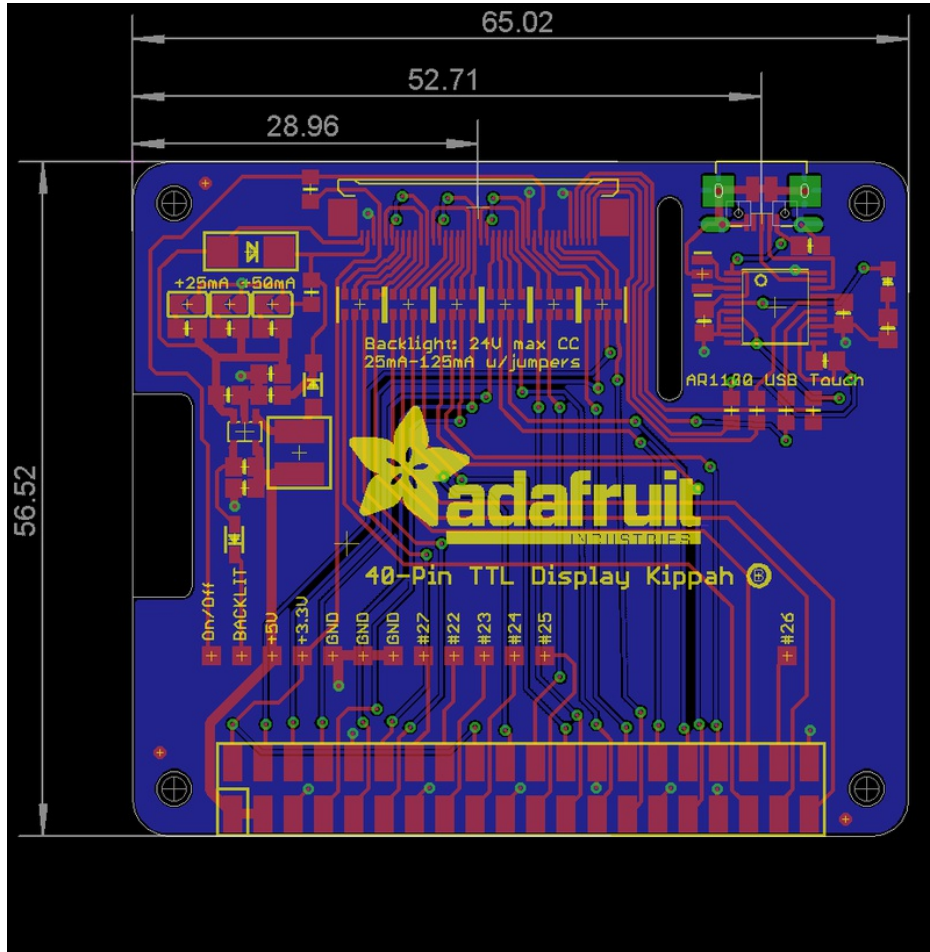
- 4.3" 480x272 - 1 strand of 7 LEDs in series, **25mA** constant current
- 5.0" 800x480 - 2 strands 6 LEDs in series, **50mA** constant current
- 7.0" 800x480 - 6 strands of 3 LEDs in series, **150mA** constant current (although 100-150mA can be used)

By default, we ship the DPI Kippot with 25mA backlight. You'll still be able to see the display on a 5" or 7" display, **but it will be dim!** That's OK, all you have to do is configure the Kippah to match your display.

Since we dont know what display you're going to use, we leave this up to you. It's easy to do, just grab your soldering iron and heat it up, then grab some solder and melt it to short out the jumpers on the Kippah here:



The left and middle jumper each add 25mA to the default 25mA. The right jumper adds 50mA. So if you want to have a 50mA backlight, short only the left jumper. For 75mA, short the right most. For 100mA solder short the left and right and for 125mA short all three!



F.A.Q.

Can I use the DPI interface and HDMI or TV out at the same time?

Nope! The DPI/HDMI/TV outputs are all the same hardware driver so you can only have one at a time. To enable/disable the DPI display, comment out or uncomment

```
enable_dpi_lcd=1  
display_default_lcd=1
```

in /boot/config.txt and reboot. It cannot be done on the 'fly'

How is that DTO blob made?

If you'd like to check out the device tree overlay and how to customize, you can read some more here:

<http://blog.reasonablycorrect.com/raw-dpi-raspberry-pi/>

Run:

```
sudo apt-get install device-tree-compiler  
wget https://raw.githubusercontent.com/robertely/dpi666/master/setup/dt-blob-dpi.dts  
sudo dtc -I dts -O dtb -o /boot/dt-blob.bin dt-blob-dpi.dts
```

to create your own!