Adafruit PiTFT 3.5" Touch Screen for Raspberry Pi

Created by lady ada

Last updated on 2016-11-11 01:07:00 AM UTC

Guide Contents

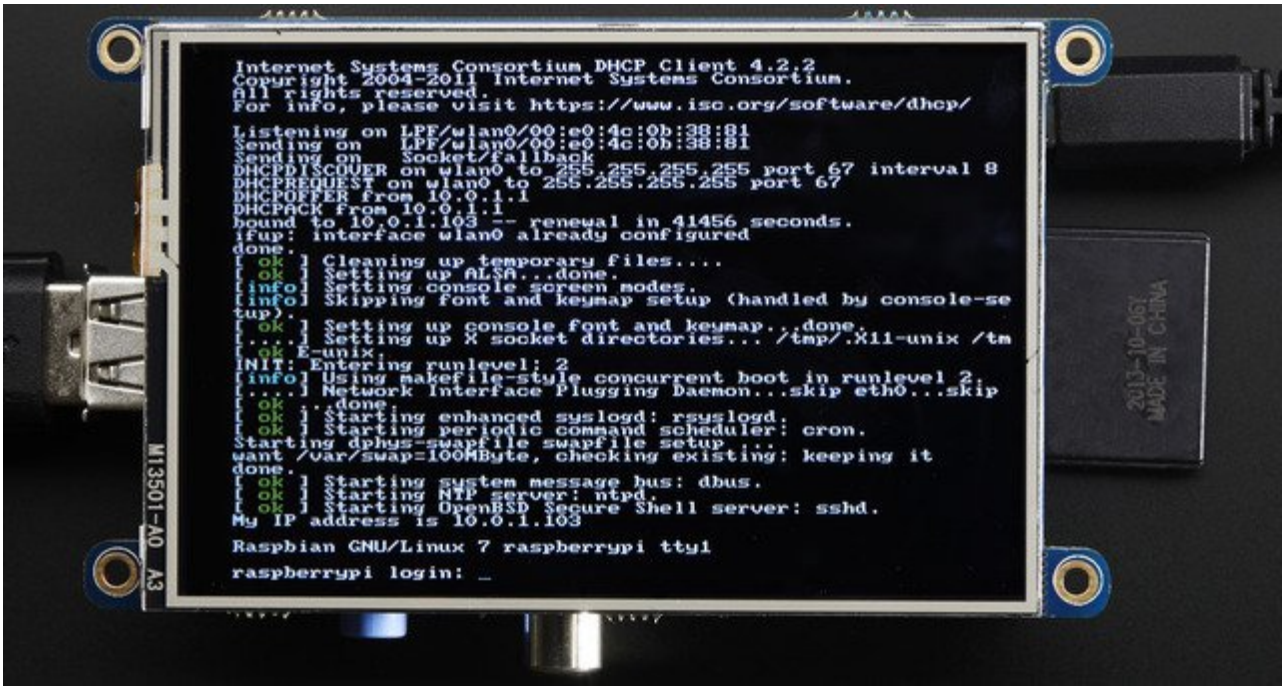
Guide Contents 2
Overview 4
Easy Install 8
Ready to go image 8
DIY Installer script 8
Step 1. Expand Filesystem 9
Step 2. Install new Kernel 9
Step 3. Enable & Configure the PiTFT 10
Detailed Install 12
Before you start 13
Download & Install Kernel 13
Resistive Touchscreen Manual Install & Calibrate 21
Setting up the Touchscreen 21
Running evtest 23
AutoMagic Calibration Script 24
Manual Calibration 26
X Calibration 28
Console Configuration 31
Turn off Console Blanking 34
- Raspbian Jessie 34
- Raspbian Wheezy 34

Displaying Images 36
Playing Videos 38
How To Play Videos 38
Converting/Resizing Videos 40
Using FBCP 44
Backlight Control 45
PWM Backlight Control with GPIO 18 45
On / Off Using STMPE GPIO 46
For older versions of PiTFT Kernel 46
More Tips 48

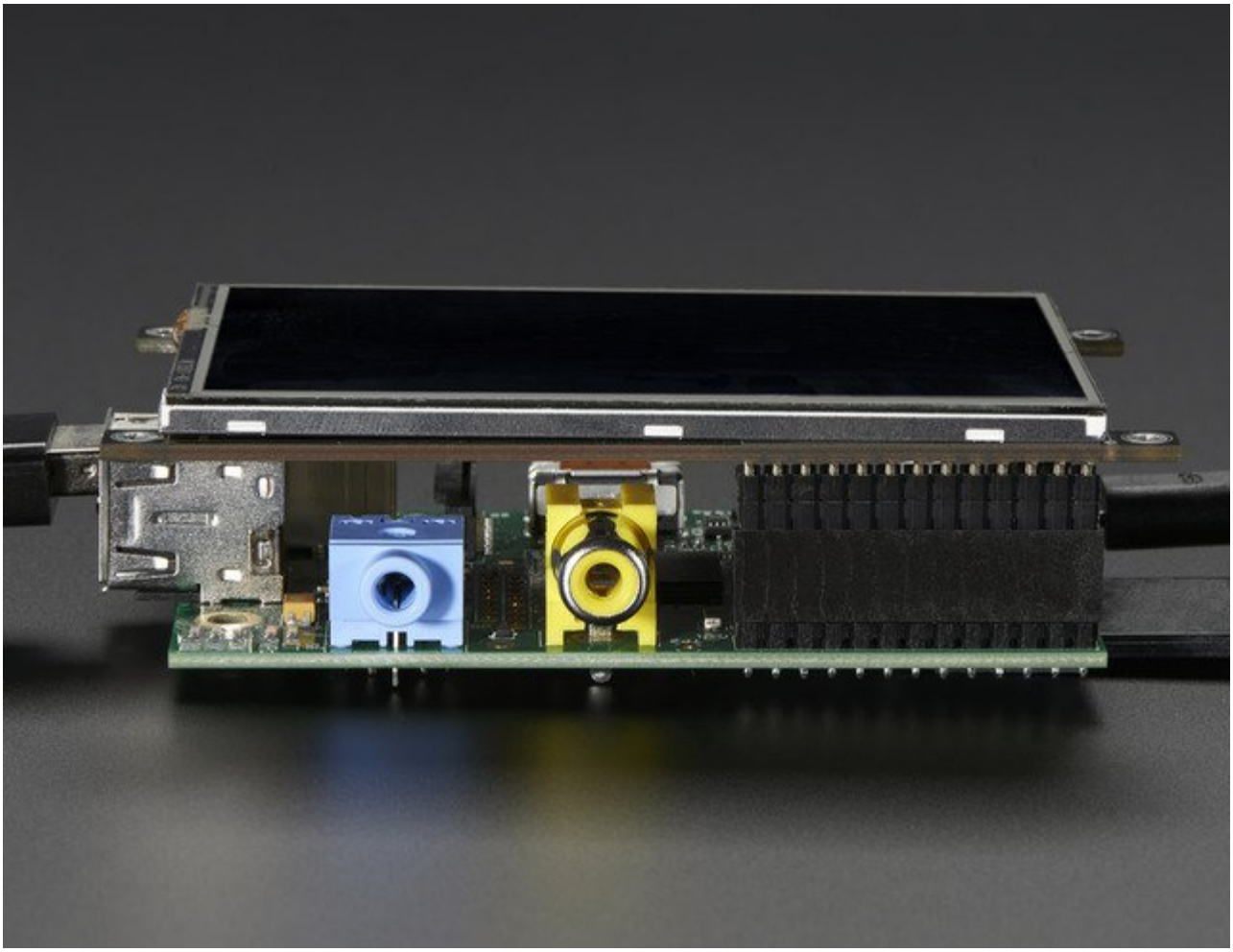
Making it easier to click icons in X 48
Right-click on a touchscreen 49
Gesture Input 50
Installation 50
Usage 50
PiTFT PyGame Tips 54
Install pip & pygame 54
Ensure you are running SDL 1.2 55
HELP! (FAQ) 58
Downloads 64
Files 64
Layout and Schematic for PiTFT Plus 3.5" 64
Layout and Schematic for original PiTFT 3.5" 66

Overview

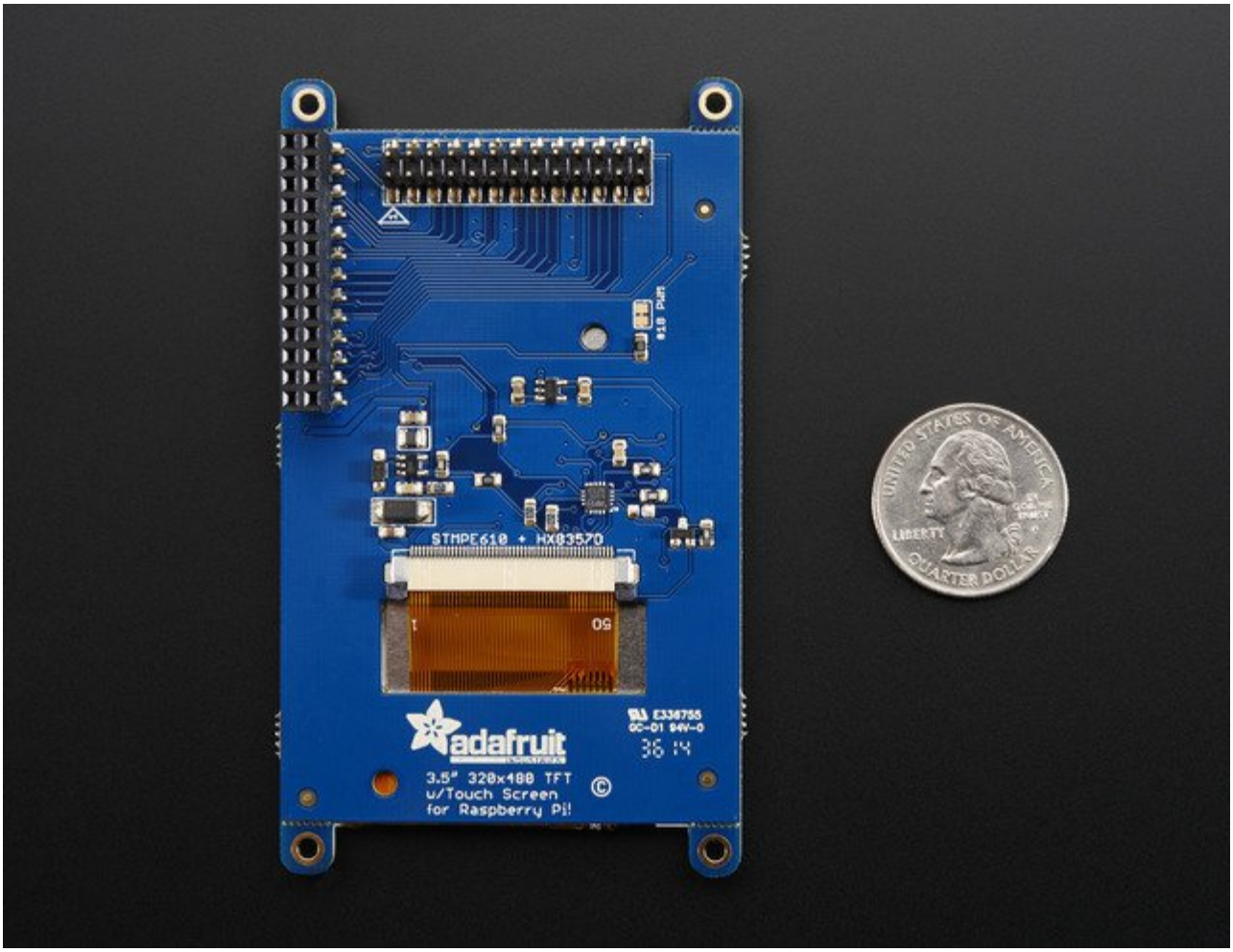
Is this not the cutest, little display for the Raspberry Pi? It features a**3.5" display with 480x320** 16-bit color pixels and a resistive touch overlay, but is only slightly larger than[our popular original](http://adafru.it/dDE). The plate uses the high speed SPI interface on the Pi and can use the mini display as a console, X window port, displaying images or video etc. Best of all it plugs right in on top!



It's designed to fit nicely onto the Pi Model A or B but also works perfectly fine with the Model B+ as long as you don't mind the PCB overhangs the USB ports by 5mm.



Uses the hardware SPI pins (SCK, MOSI, MISO, CE0, CE1) as well as GPIO #25 and #24. GPIO #18 can be used to PWM dim the backlight if you like. All other GPIO are unused. There's a 2x13 header on the bottom, you can connect a standard Pi GPIO cable to it to use any of the other pins ask you like



Best of all, it comes **fully assembled** and ready to plug into your Pi! You can use this as a display for running the X interface, or pygame. You can also have an HDMI display seperately connected.

Easy Install

The PiTFT requires kernel support and a couple other things to make it a nice stand-alone display. We have a detailed step-by-step setup for hackers who want to tweak, customize or understand the PiTFT setup. If you just want to get going, check out the following for easy-install instructions!

Ready to go image

f you want to start with a fresh image, we have two for Raspbian. There's the larger 'classic Jessie' image that will boot into X by default, and requires a 8G image, it has a lot more software installed. There's also the smaller 'Jessie Lite' that will boot into the command line, and can be burned onto a 2G card! Click below to download and install into a new SD card. Unzip and follow the classic SD card burning tutorials (http://adafru.it/aMW)

This image is customized for the Resistive Touch 3.5" TFT, also known as PID #2097! Not for the 2.2", 2.4", 2.8" or 3.2" PiTFTs
Download Jessie-based PiTFT 3.5" Resistive Image for Pi 1, Pi 2 and Pi 3 (Sept 23, 2016 image, made Nov 10)
http://adafru.it/siF
Download Jessie Lite-based PiTFT 3.5" Resistive Image for Pi 1, Pi 2 and Pi 3 (Sept 23, 2016 image, made Nov 10)
http://adafru.it/sja

Older Images:

- Raspbian Jessie 2016/03/25-based image (http://adafru.it/mAb)
- Raspbian Jessie 2016/03/25-based image (http://adafru.it/mAG)
- Raspbian Jessie 2015/09/24-based image (http://adafru.it/iDD)
- Raspbian Wheezy 2015/09/24-based image (http://adafru.it/idy)
- Raspbian 2014/09/09-based image (http://adafru.it/e10)
- Raspbian 2015/03/12 image (http://adafru.it/eUE)

DIY Installer script

If you don't want to download an image, you can install our custom kernel and run our installation package helper from inside your existing Raspbian install. It will configure your Pi for PiTFT joy

The helper is available for perusal here (http://adafru.it/eIn) if you are interested in how it works

Step 1. Expand Filesystem

Start by expanding the filesystem **This is required!!!**

```
    sudo raspi-config
(expand filesystem)
sudo reboot
```

Step 2. Install new Kernel

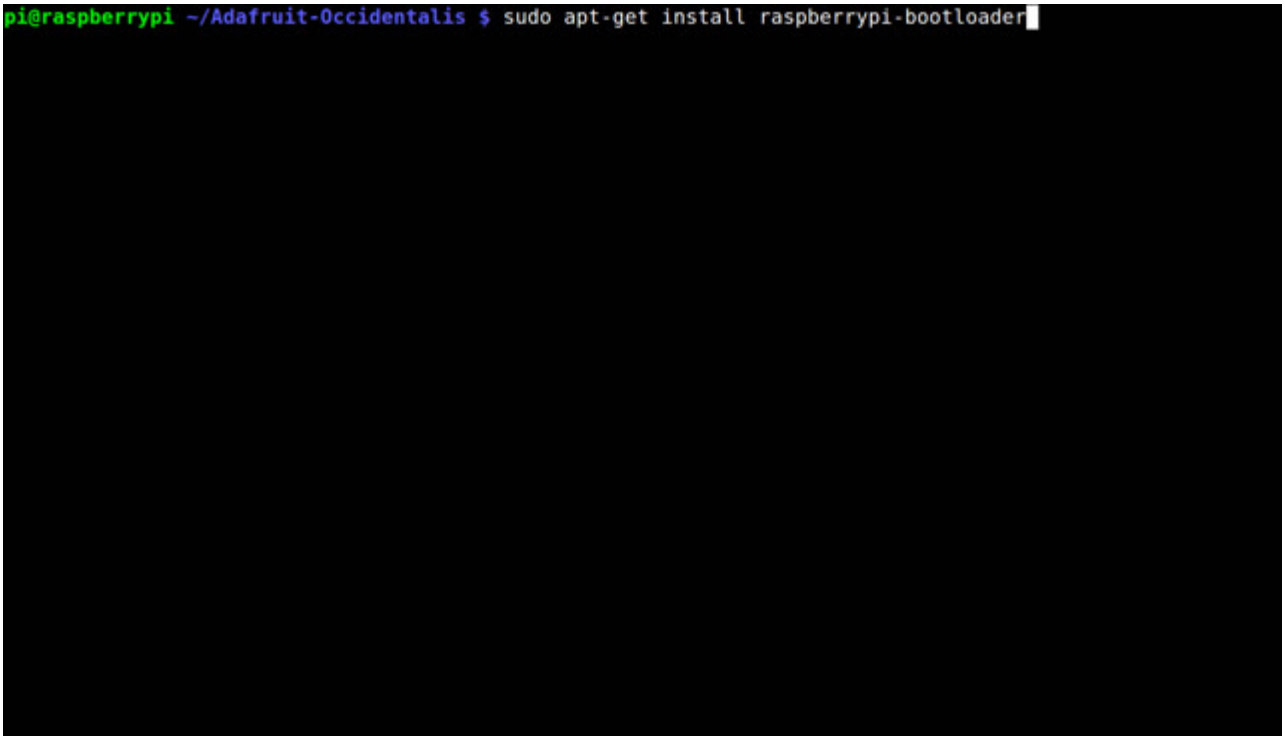
Then, once the filesys is expanded, download and install t he new kernel by running the following commands:

```
curl -SLs https://apt.adafruit.com/add-pin | sudo bash
sudo apt-get install -y raspberrypi-bootloader adafruit-pitft-helper raspberrypi-kernel
```

The first command adds **apt.adafruit.com** to your repository list, so you can grab code directly from adafruit's servers

The next line does the actual download and installation, it'll take a while because there's a lot

of software to replace for PiTFT support.



It's normal for the Pi to pause and/or take a while at this step for many minutes, theres a lot of kernel software to replace

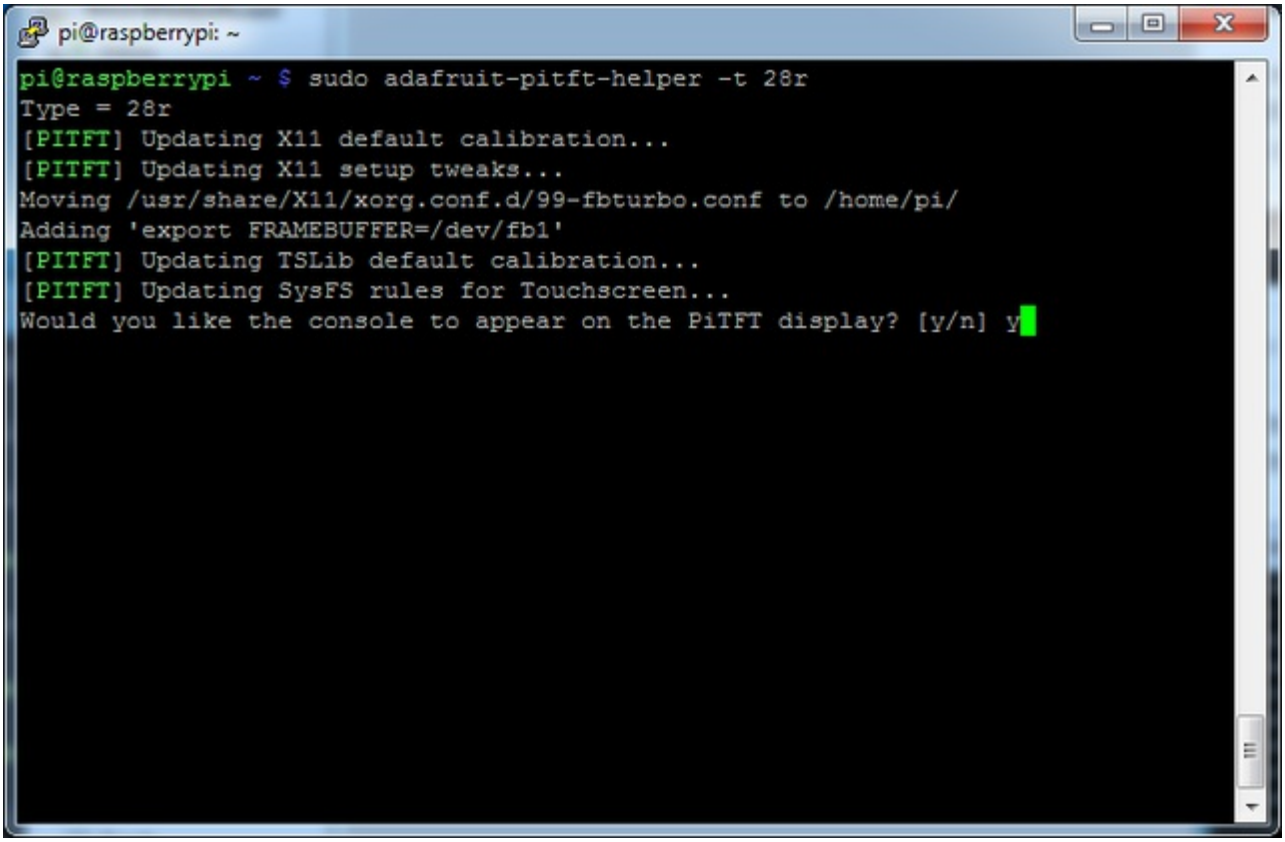
Step 3. Enable & Configure the PiTFT

OK now the kernel and helper are installed, all you have to do is run the helper which will configure the kernel device tree overlays and add the few configurations to make the console show up, etc.

```
sudo adafruit-pitft-helper -t 35r
```

This will install the "3.5 inch Resistive" type of PiTFT into the current install.

At the end you will be prompted on whether you want the text console to appear on the PiTFT. Answer Y or N depending on your personal desires!



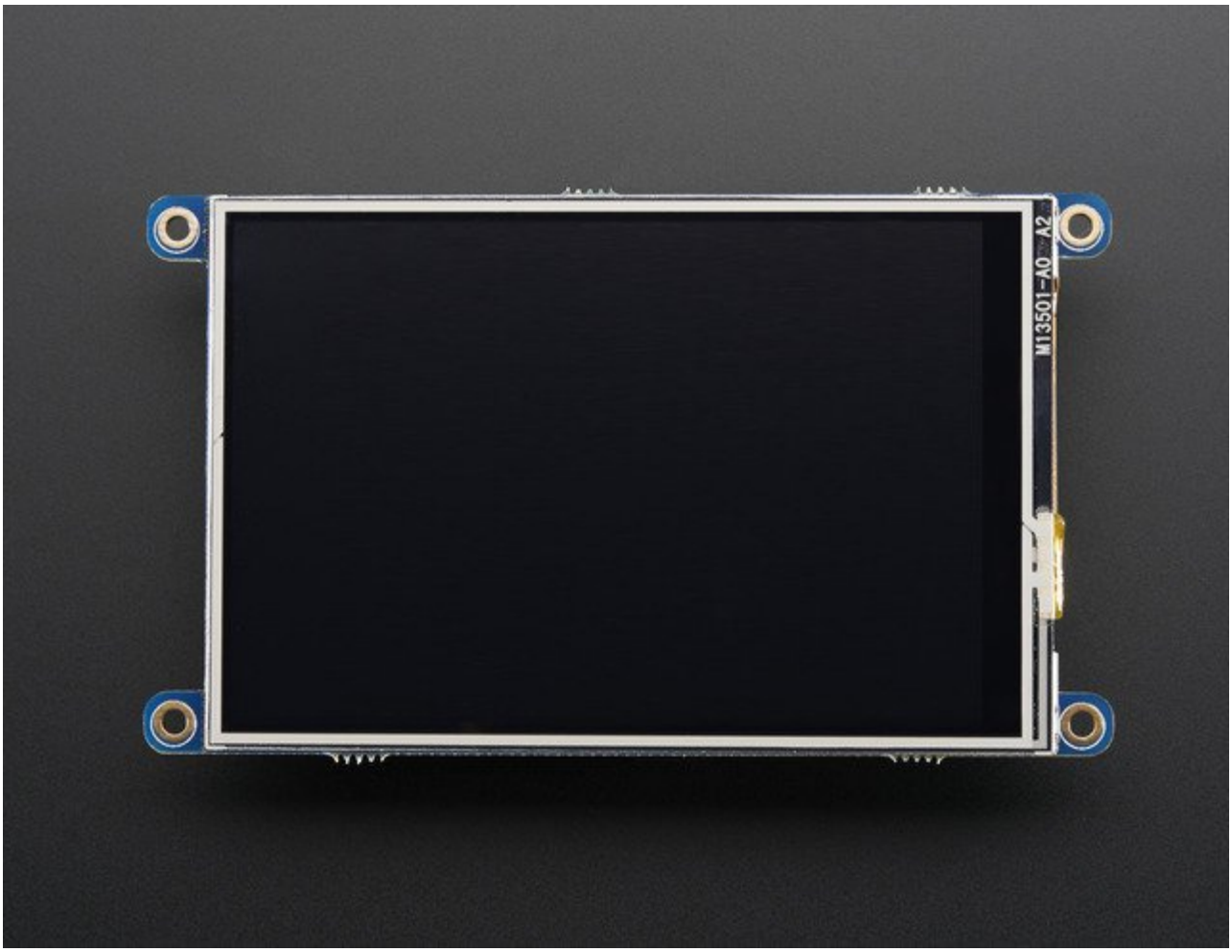
That's it!

Run **sudo reboot** to try out your fancy new PiTFT :)

Detailed Install

If you've grabbed our Easy Install image, or use the script, this step is not required, it's already done! This is just for advanced users who are curious on how to configure and customize the kernel install

In the next few steps we'll cover the**detailed** installation procedure. Chances are, you should grab the Easy Install image or script. If you have some interest in the details of how we install the PiTFT setup, read on!



In order to add support for the 3.5" TFT and touchscreen, we'll need to install a new Linux Kernel. Lucky for you, we created a kernel package that you can simply install *over* your current Raspbian (or Raspbian-derived) install instead of needing a whole new image. This makes it easier to keep your install up-to-date.

To use our kernel .deb files you must be using Raspbian or derivative. This wont work with

Arch or other Linux flavors. As Raspbian is the official OS for the Pi, that's the only Linux we will support! Others can recompile their own kernel using our github commits (http://adafru.it/aPa)but we have no tutorial or support or plans for such.

Before you start

You'll need a working install of Raspbian with network access.If you need help getting that far, check out our collection of Pi tutorials (http://adafru.it/aWq).

We'll be doing this from a console cable connection, but you can just as easily do it from the direct HDMI/TV console or by SSH'ing in. Whatever gets you to a shell will work!

Also, run **sudo apt-get update** !

To run these all the setup and config commands you'll need to be logged into a proper Terminal - use ssh, a console cable, or the main text console (on a TV). The WebIDE console may not work.

Download & Install Kernel

The only way we're distributing the PiTFT kernel packages right now is thru apt.adafruit.com so you'll still need to run:

```
curl -SLs https://apt.adafruit.com/add-pin | sudo bash
```

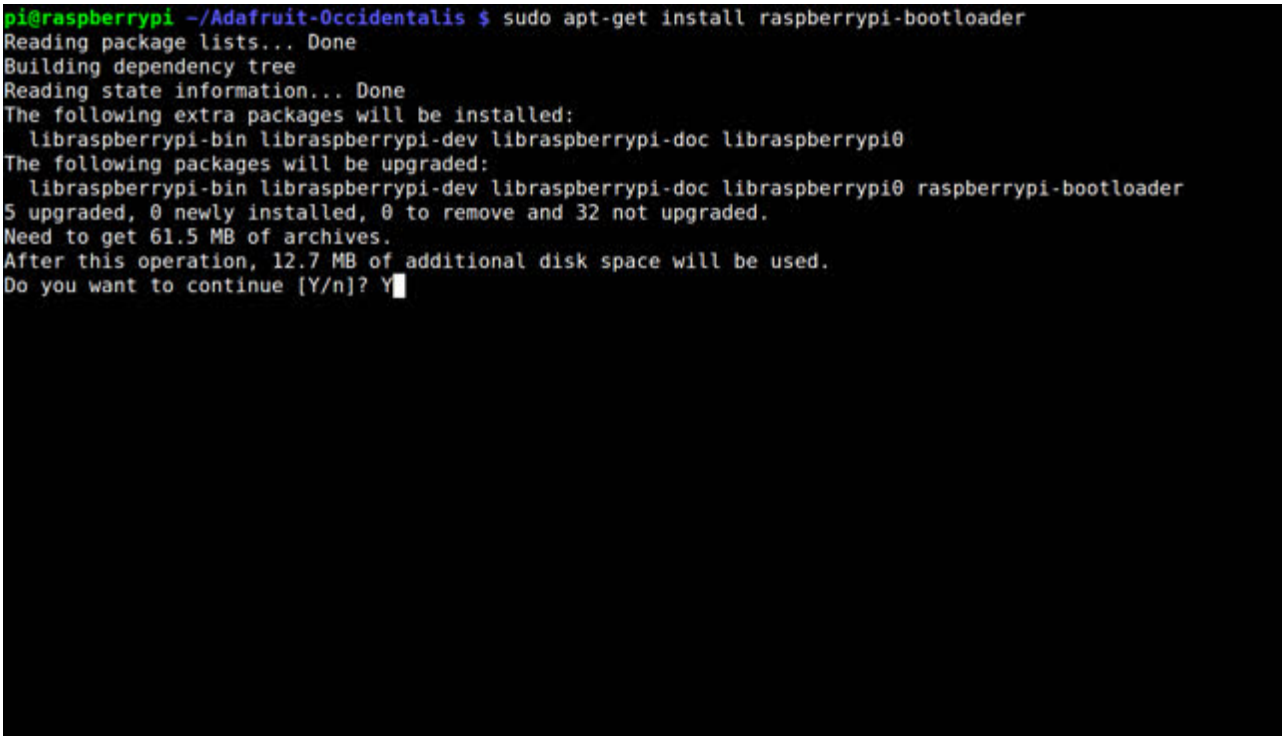
To add apt.adafruit.com to your list of software sources

```
pi@raspberrypi ~ $ curl -SLs https://apt.adafruit.com/add-pin | sudo bash
```

Then install the kernel with

sudo apt-get install raspberrypi-bootloader

```
pi@raspberrypi ~/Adafruit-Occidentalis $ sudo apt-get install raspberrypi-bootloader
```

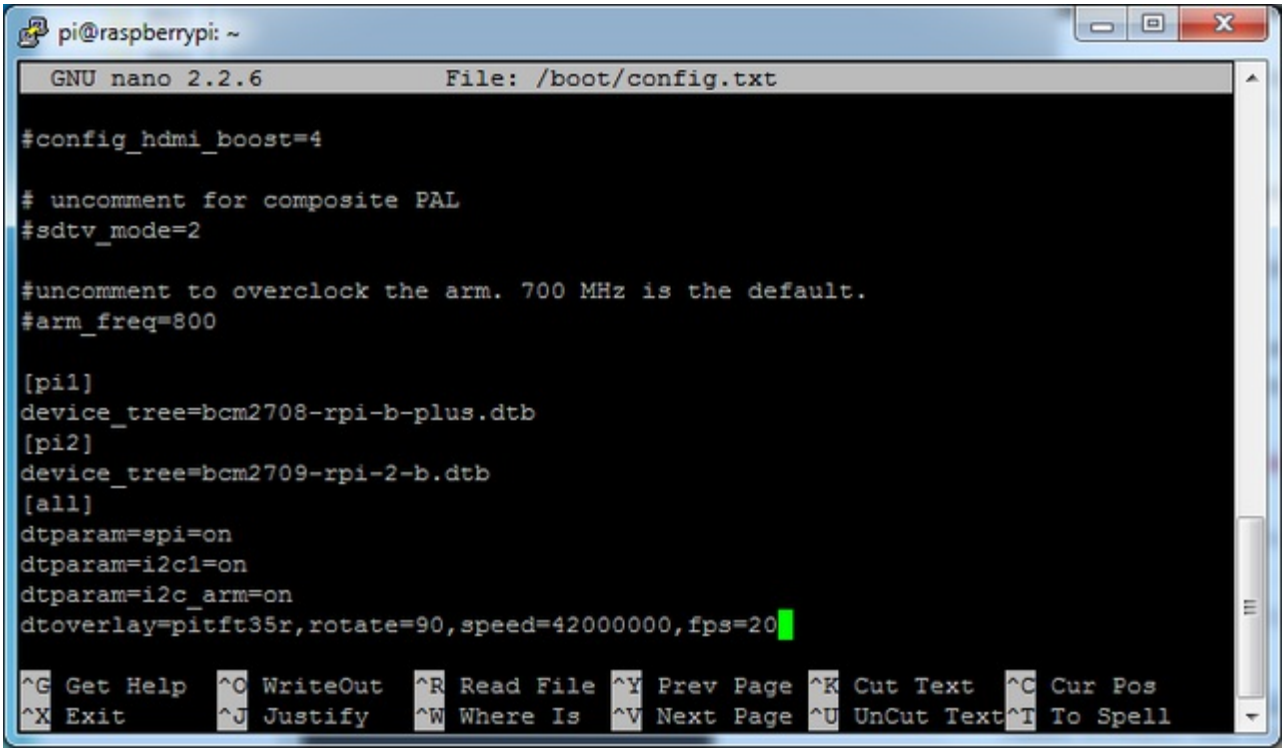


OK since you're not going to run the helper, lets add the device tree overlay manually. Edit /boot/config.txt with

sudo nano /boot/config.txt

and add the following lines at the end:

```
[pi1]
device_tree=bcm2708-rpi-b-plus.dtb
[pi2]
device_tree=bcm2709-rpi-2-b.dtb
[all]
dtparam=spi=on
dtparam=i2c1=on
dtparam=i2c_arm=on
dtoverlay=pitft35r,rotate=90,speed=42000000,fps=20
```



The **rotate=** variable tells the driver to rotate the screen **0 90 180** or **270** degrees.
0 is portrait, with the bottom near theUSB jacks
90 is landscape, with the bottom of the screen near the headphone jack
180 is portrait, with the top near the USB jacks
270 is landscape, with the top of the screen near the headphone jack.
You can change this file with **nano** and reboot to make the change stick.

The **speed=** variable tells the driver how to fast to drive the display. 42MHz (**42000000**) is a good place to start but if your screen is acting funny, try taking it down to 16MHz (**16000000**) *especially* if you're doing something like using a GPIO extender to put the screen away from the Pi.

Save the file. Now we'll just reboot to let it all sink in.

> **sudo shutdown -h now** (if you don't have the TFT installed, shutdown, place the TFT on the Pi and re-power)

or

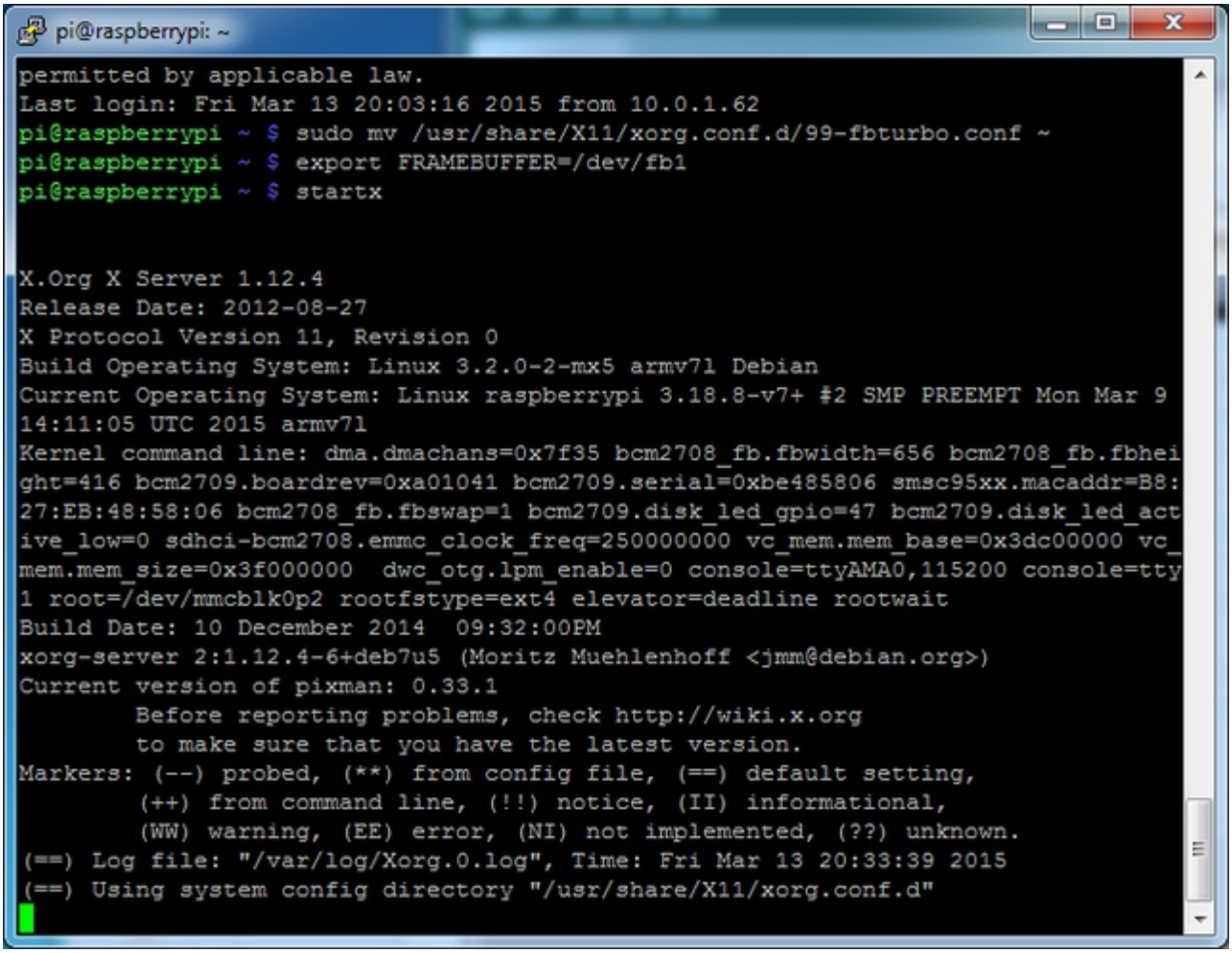
> **sudo reboot** (if you have the TFT plate installed already)

When the Pi restarts, the attached PiTFT should start out all white and then turn black. That means the kernel found the display and cleared the screen. If the screen did not turn black, that means that likely there's something up with your connection or kernel install. Solder anything that needs resoldering!

Now that you're rebooted, log back in on the console/TV/SSH. There's nothing displayed on the screen yet, we'll do a test to make sure everything is perfect first!

Run the following commands to startx on the/**dev**/**fb1** framebuffer, a.k.a PiTFT screen:

```
sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~
export FRAMEBUFFER=/dev/fb1
startx
```



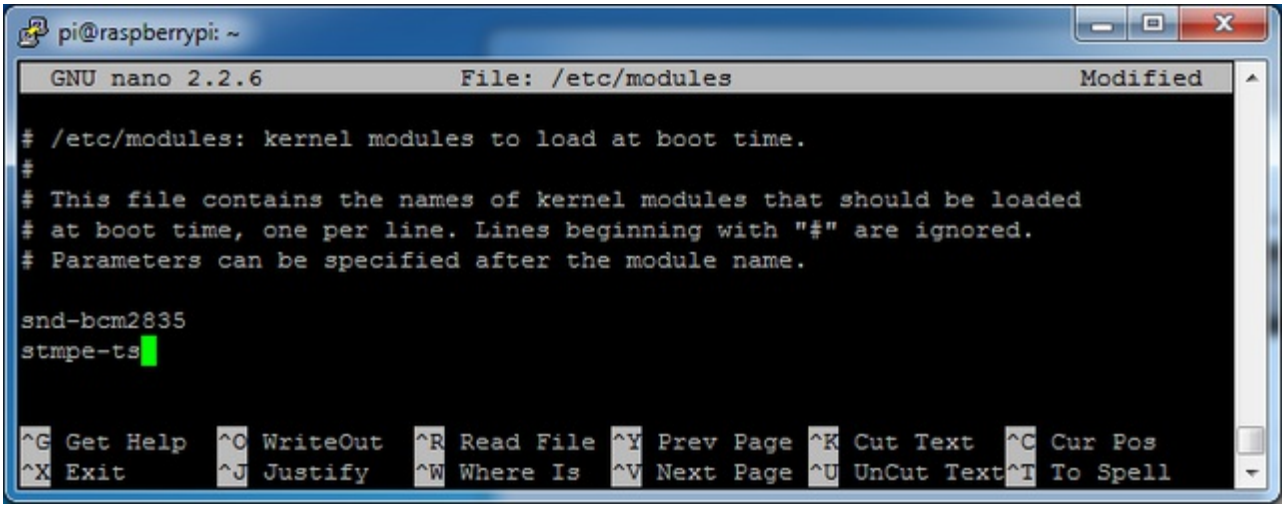
You should see the Pi desktop show up on the TFT! Congrats, you've completed the first test perfectly.

Hit Control-C in the console to quit the X server so we can continue configuration

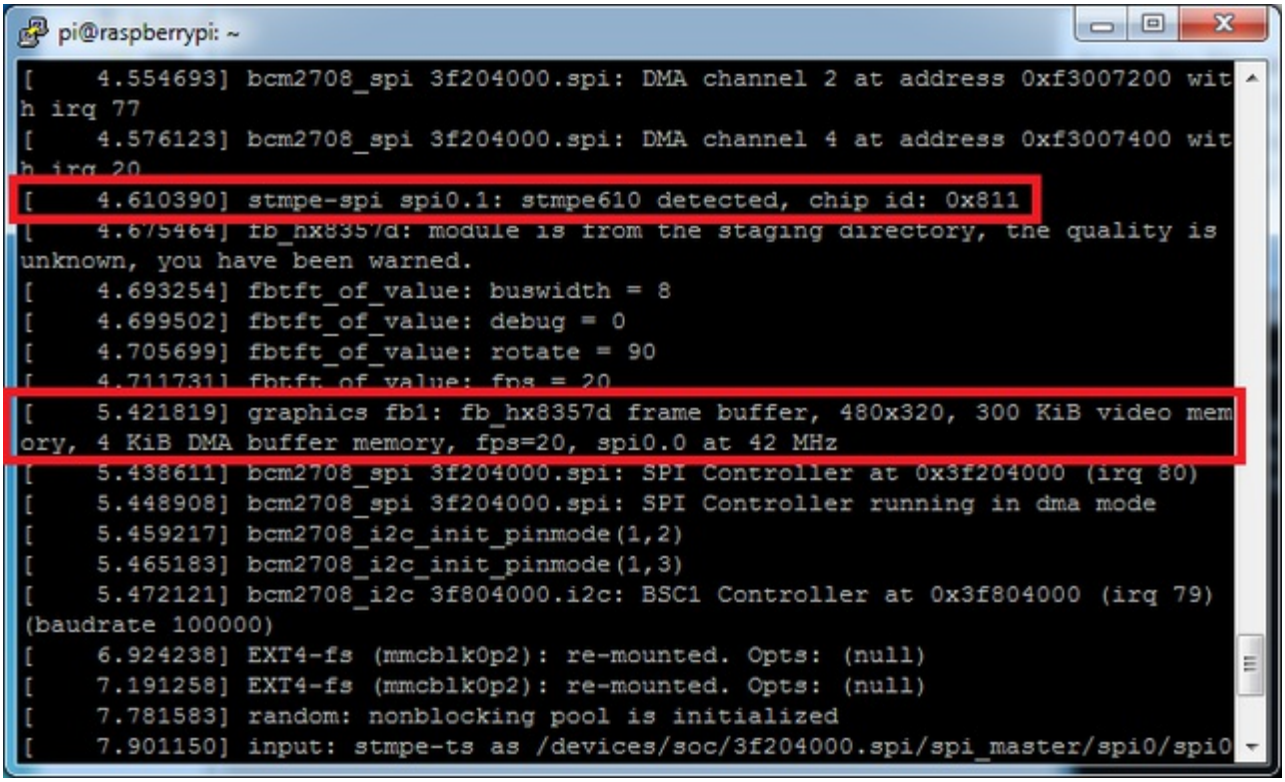
Next up we'll add support for the touch screen automatically on boot. Edit the module list with

sudo nano /etc/modules

and add **stmpe-ts** on a line at the end



Save the file and reboot the Pi with**sudo reboot** and look at the console output (or run **dmesg** in the console window after logging in) you will see the modules install. Look in particular for the STMPE610 detection and the HX5387D screen frequency as highlighted here

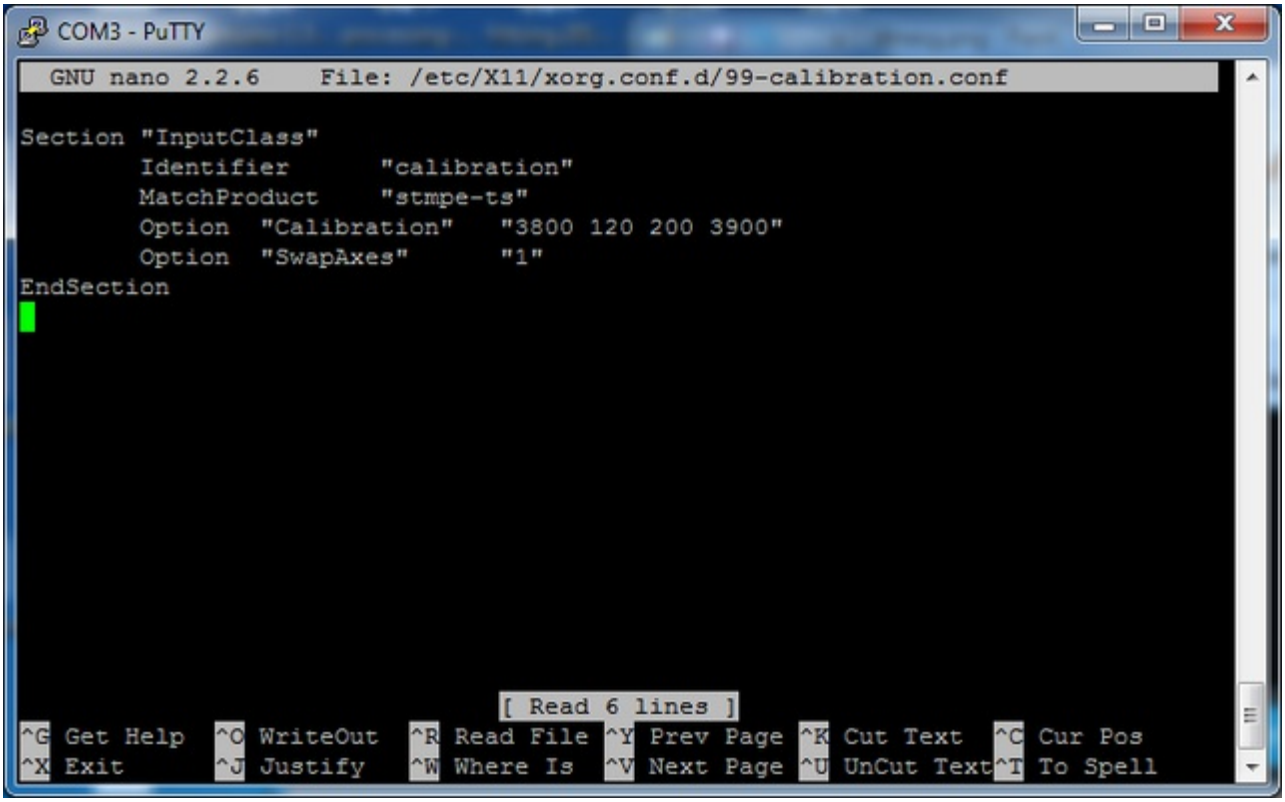


We can set up the touchscreen for**rotate=90** configuration by doing the following (for more delicate calibration or for other rotate=XX values, see the next section)
Create the directory and new calibration configuration file:

sudo mkdir /etc/X11/xorg.conf.d
sudo nano /etc/X11/xorg.conf.d/99-calibration.conf

and enter in the following lines, then save.

```
Section "InputClass"
        Identifier      "calibration"
        MatchProduct    "stmpe-ts"
        Option  "Calibration"   "3800 120 200 3900"
        Option  "SwapAxes"      "1"
EndSection
```



You can now try to run X again with

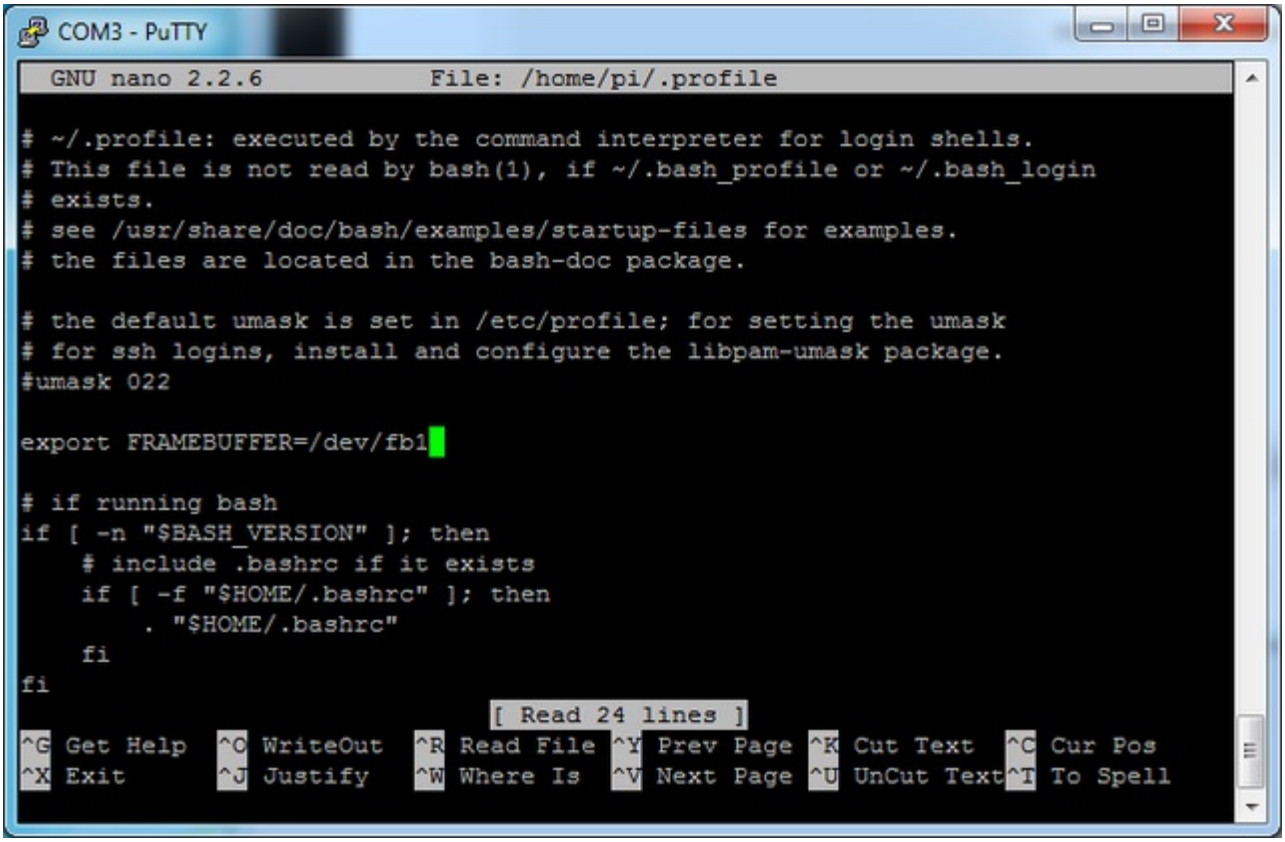
FRAMEBUFFER=/dev/fb1 startx

Type Control-C to quit **X**

If you don't ever want to have to type FRAMEBUFFER=/dev/fb1 before startx, you can make it a default state by editing your profile file: **sudo nano ~/.profile** and adding

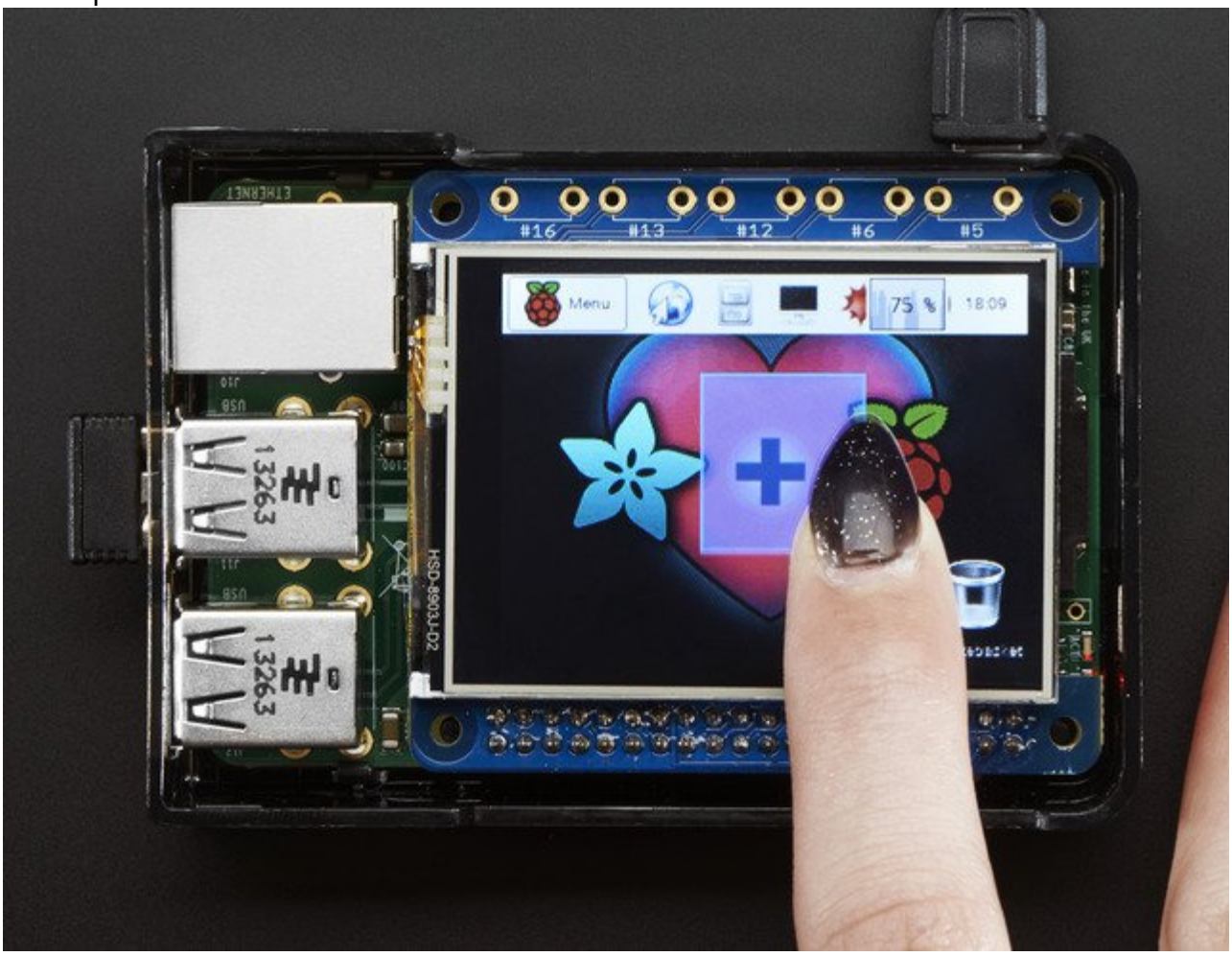
export FRAMEBUFFER=/dev/fb1

near the top and saving the file. Then reboot to reload the profile file. It will now always assume you want to use /dev/fb1


COM3 - PuTTY
GNU nano 2.2.6 File: /home/pi/.profile
~/.profile: executed by the command interpreter for login shells.
This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
exists.
see /usr/share/doc/bash/examples/startup-files for examples.
the files are located in the bash-doc package.
the default umask is set in /etc/profile; for setting the umask
for ssh logins, install and configure the libpam-umask package.
#umask 022
export FRAMEBUFFER=/dev/fb1
# if running bash
if [-n "$BASH_VERSION"]; then
# include .bashrc if it exists
if [-f "$HOME/.bashrc"]; then
. "$HOME/.bashrc"
fi
fi
[Read 24 lines]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Resistive Touchscreen Manual Install & Calibrate

If you've grabbed our Easy Install image, or used the installer script, this step is not required, it's already done! This is just for advanced users who are curious on how to configure and customize the touchscreen
This procedure is identical for the 2.4", 2.8", 3.2" and 3.5" Resistive PiTFTs. Not for use with the Capacitive PiTFT!



Setting up the Touchscreen

Now that the screen is working nicely, we'll take care of the touchscreen. There's just a bit of calibration to do, but it isn't hard at all.

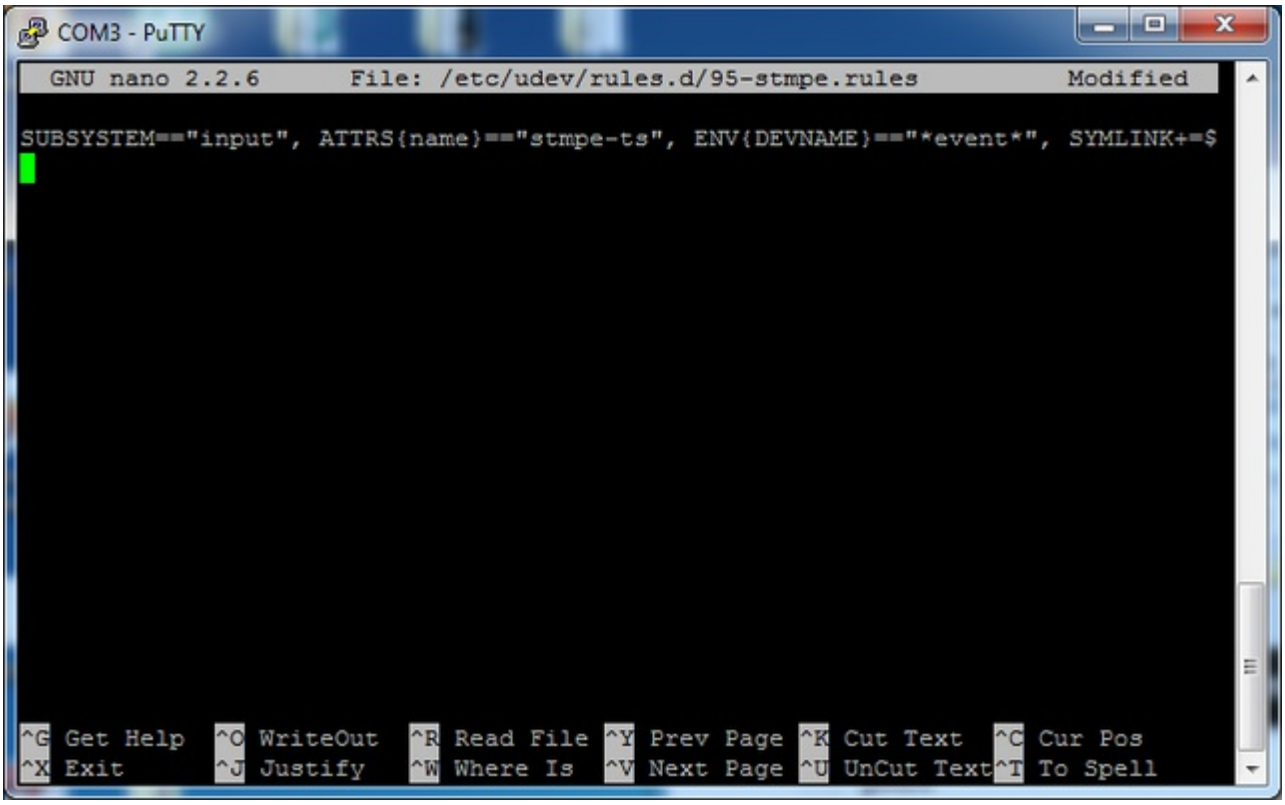
Before we start, we'll make a **udev** rule for the touchscreen. That's because the **eventX** name of the device will change a lot and its annoying to figure out what its called depending on whether you have a keyboard or other mouse installed.

Run

```
sudo nano /etc/udev/rules.d/95-stmpe.rules
```

to create a new **udev** file and copy & paste the following line in:

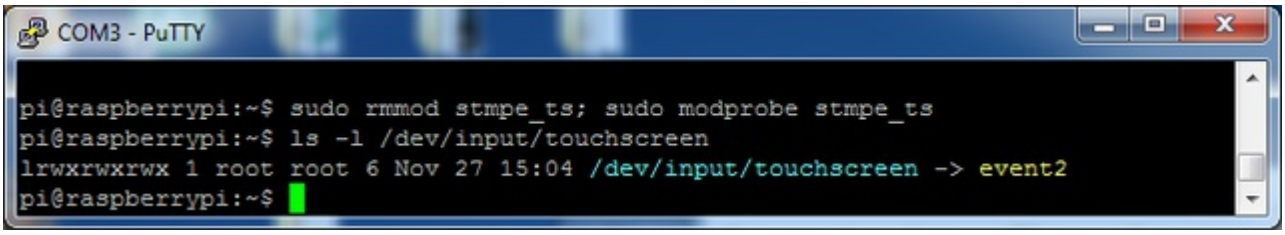
```
SUBSYSTEM=="input", ATTRS{name}=="stmpe-ts", ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```



Remove and re-install the touchscreen with

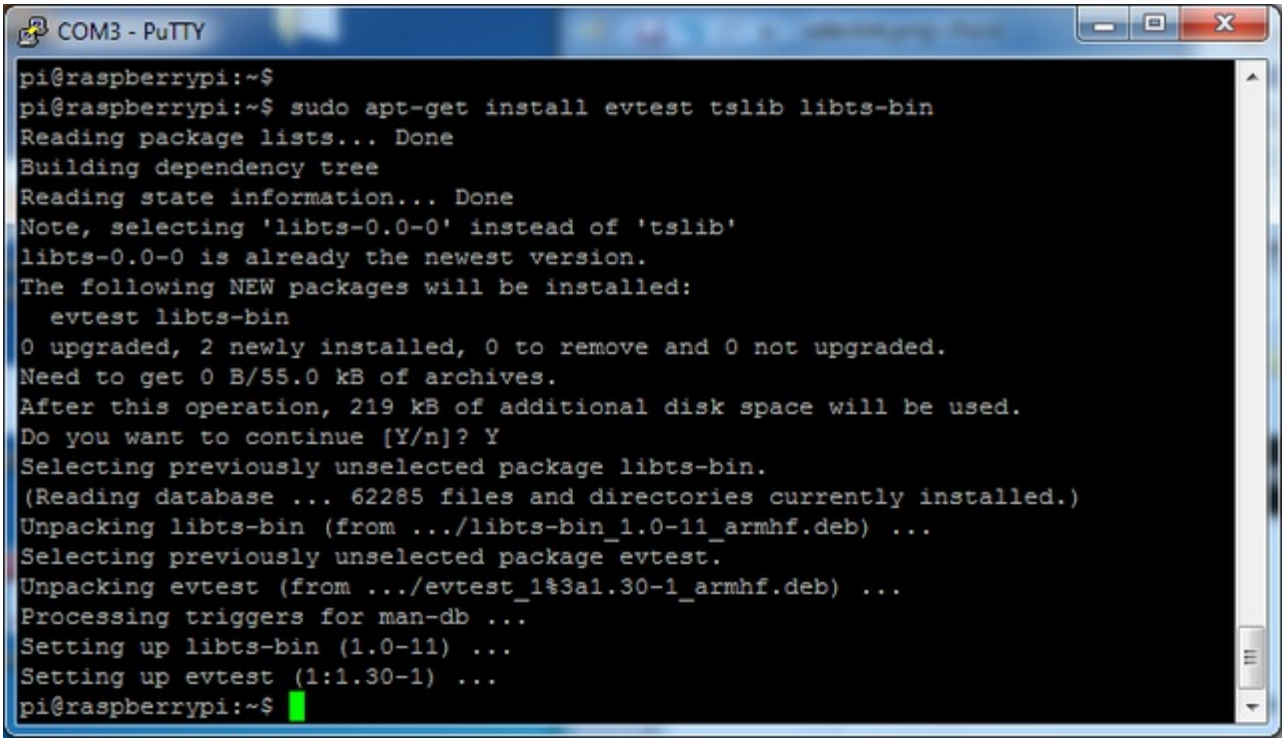
```
sudo rmmod stmpe_ts; sudo modprobe stmpe_ts
```

Then type **ls -l /dev/input/touchscreen**
It should point to **eventX** where X is some number, that number will be different on different setups since other keyboards/mice/USB devices will take up an event slot



There are some tools we can use to calibrate & debug the touchscreen. Install the "event test" and "touchscreen library" packages with

```
sudo apt-get install evtest tslib libts-bin
```

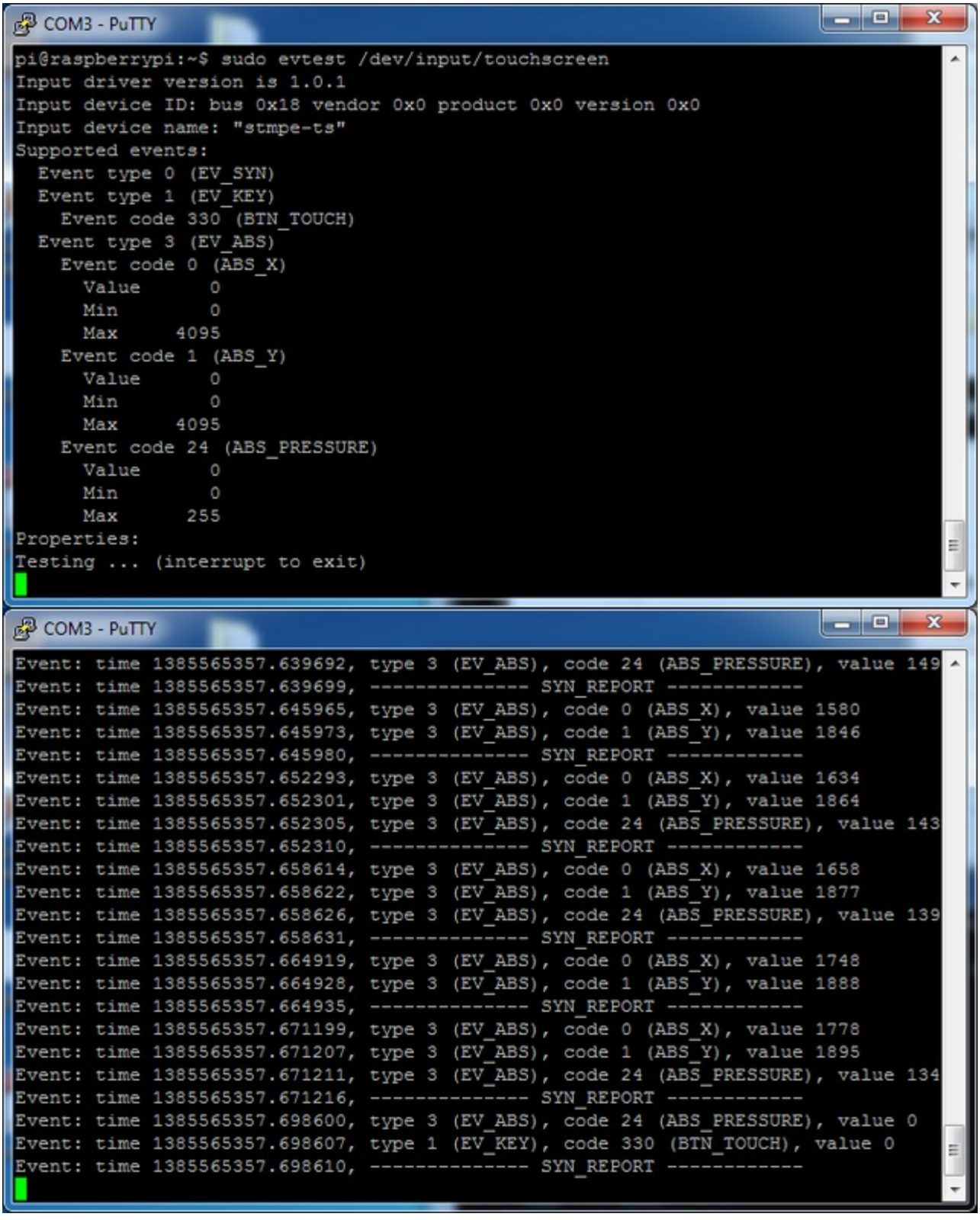


Running evtest

Now you can use some tools such as

```
sudo evtest /dev/input/touchscreen
```

which will let you see touchscreen events in real time, press on the touchscreen to see the reports.



AutoMagic Calibration Script

If you rotate the display you need to recalibrate the touchscreen to work with the new screen

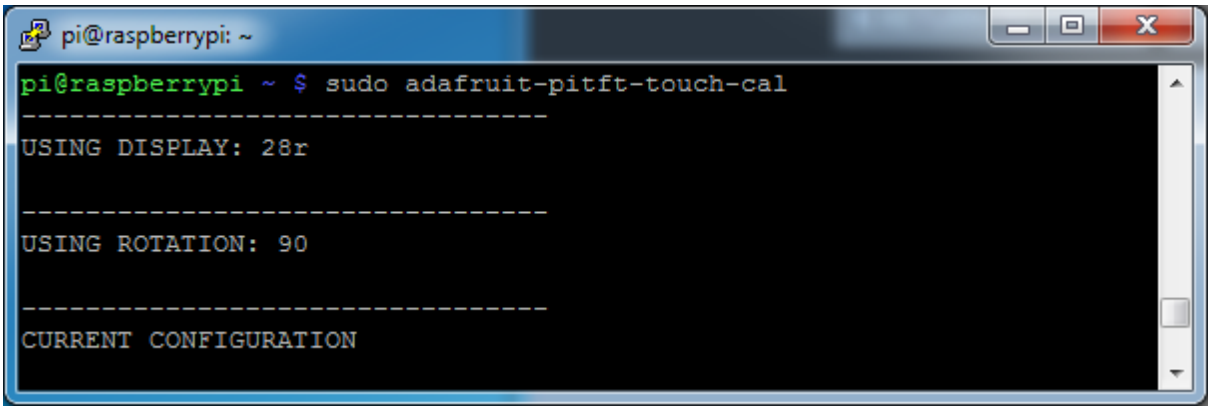
orientation. You can manually run the calibration processes in the next section, or you can run a small Python script which will automatically set a default touchscreen calibration based on the screen orientation.

[This helper is automatically installed for you but if you'd like you can check it out here on github](http://adafru.it/eIu) (http://adafru.it/eIu)

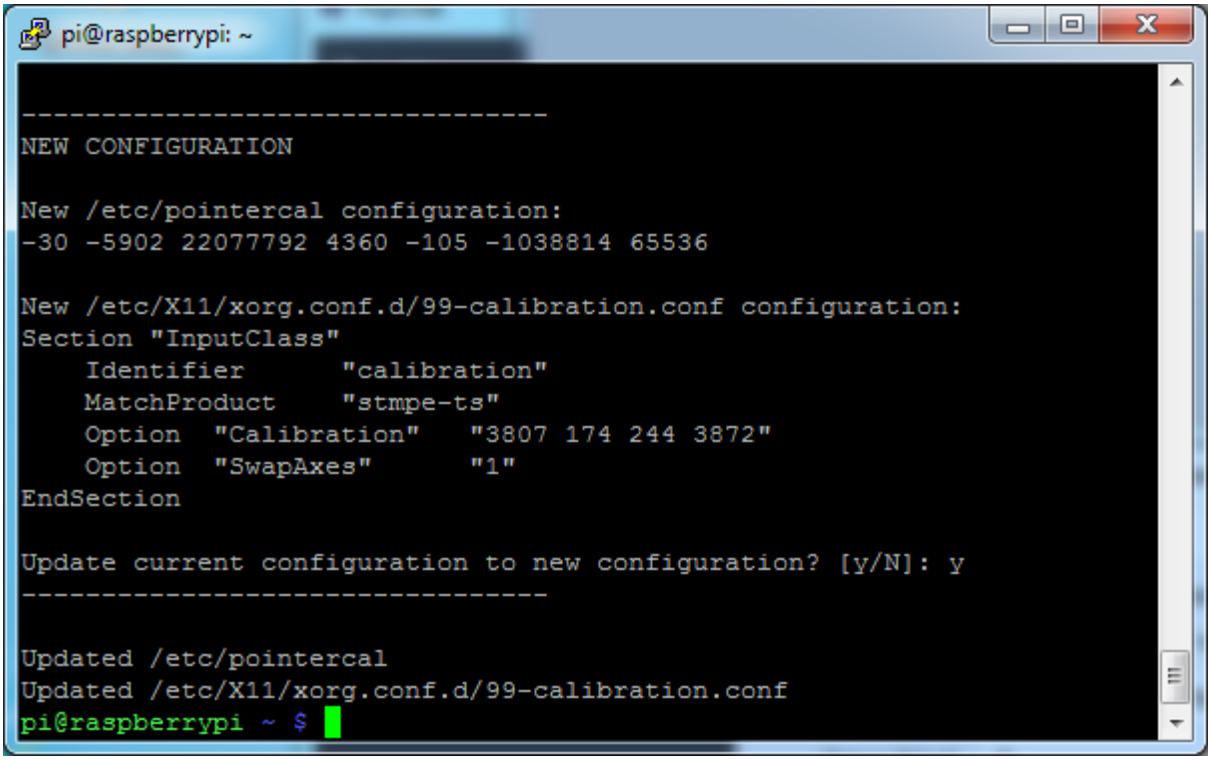
Run it at the command line with

```
sudo adafruit-pitft-touch-cal
```

it will try to figure out what display you have installed and the rotation it's set up for



By default the script will attempt to read the screen orientation by examining the PiTFT module configuration with modprobe. If the script can read the orientation it will print out the current orientation, the current touchscreen calibration values, and the new touchscreen calibration values baesd on the current orientation. Before updating the calibration the script will ask you to confirm that you'd like to make the change. Press **y** and enter to confirm.



Try using this default calibration script to easily calibrate your touchscreen display. Note that the calibration values might not be exactly right for your display, but they should be close enough for most needs. If you need the most accurate touchscreen calibration, follow the steps in the next section to manually calibrate the touchscreen.

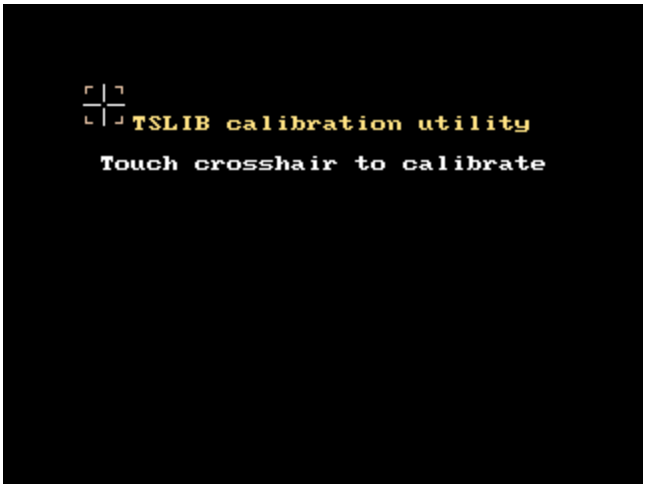
Manual Calibration

If the "automagic" calibration technique isn't working for you, or you have some other setup where you need to carefully calibrate you can do it 'manually'

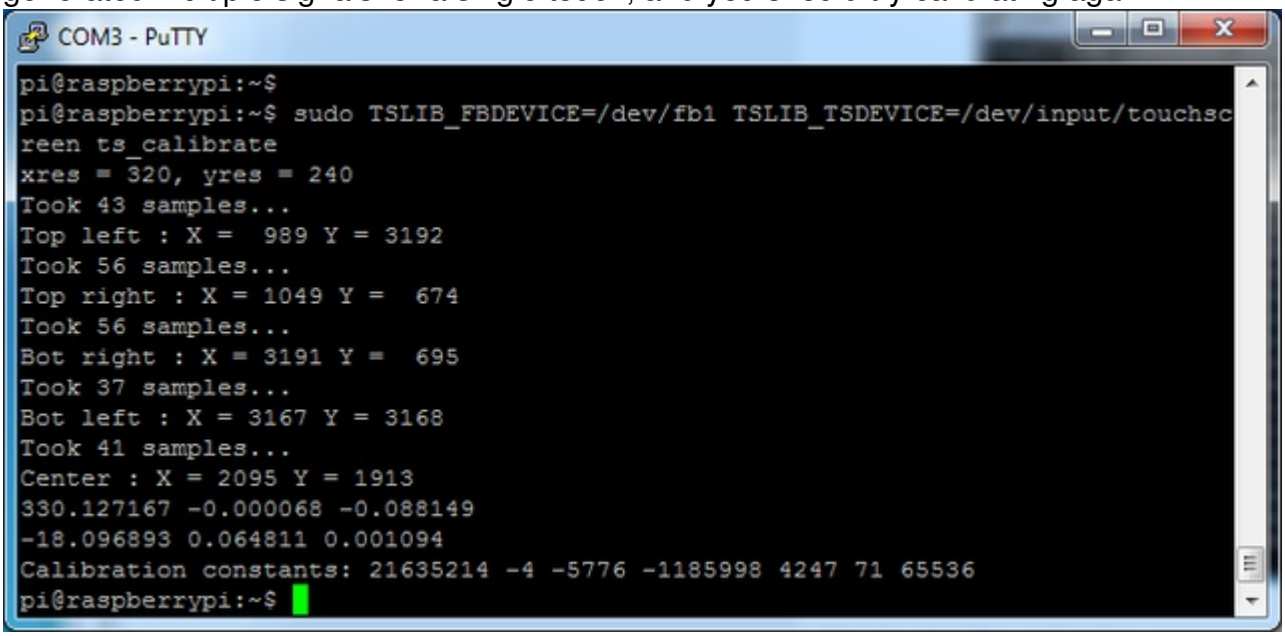
You will want to calibrate the screen once but shouldn't have to do it more than that. We'll begin by calibrating on the command line by running

```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_calibrate
```

follow the directions on the screen, touching each point. Using a stylus is suggested so you get a precise touch. Don't use something metal, plastic only!



You should see five crosshair targets. If you see less than that, the touchscreen probably generated multiple signals for a single touch, and you should try calibrating again.



Next you can run

```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_test
```

which will let you draw-test the touch screen. Go back and re-calibrate if you feel the screen isn't precise enough!

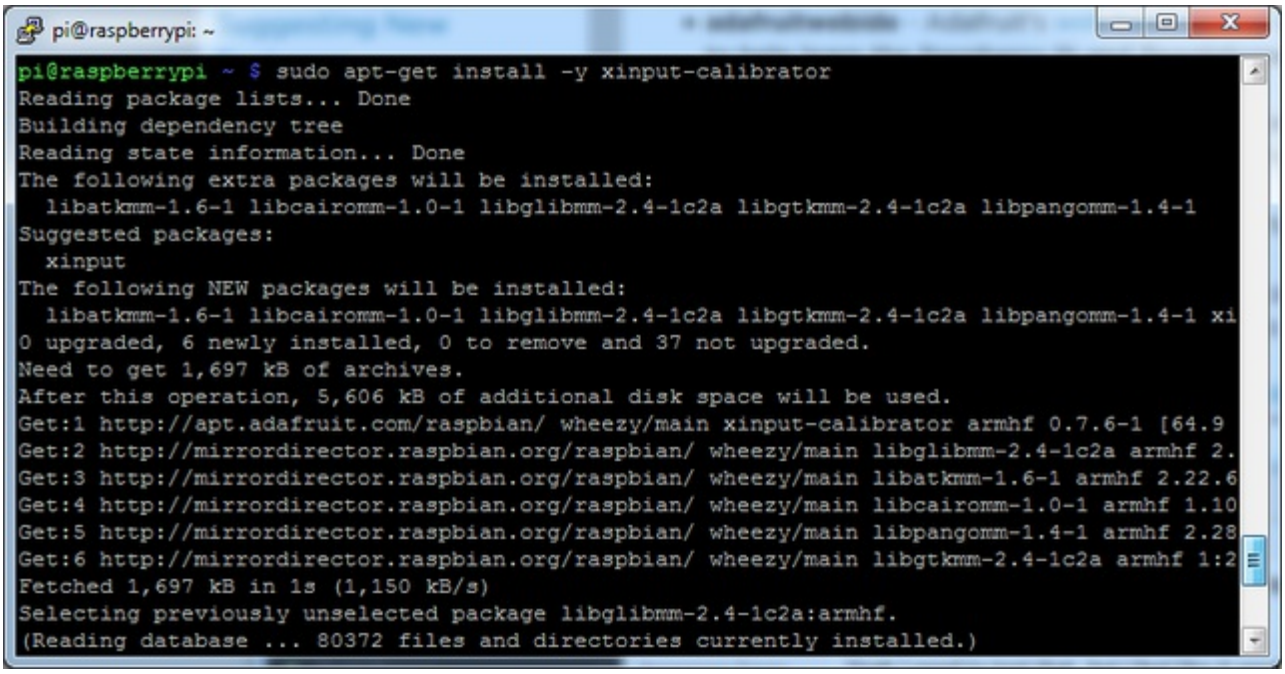
X Calibration

You can also calibrate the X input system but you have to use a different program called **xinput_calibrator**

You can do this if the calibration on the screen isnt to your liking or any time you change the **rotate=XX** module settings for the screen. Since the screen and touch driver are completely separated, the touchscreen doesn't auto-rotate

Normally you'd have to compile it but we have a ready to go package for you so run:

```
sudo apt-get install -y xinput-calibrator
```



Before you start the xinput_calibrator you will need to delete the old calibration data so run

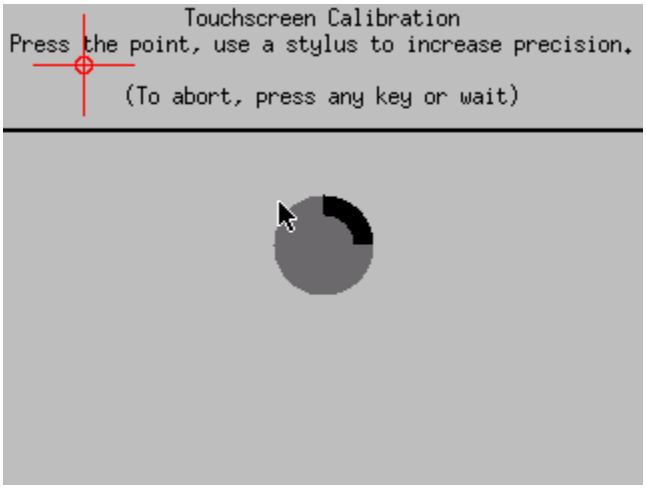
```
sudo rm /etc/X11/xorg.conf.d/99-calibration.conf
```

Before running `startx` and the calibrator - otherwise it gets really confused!

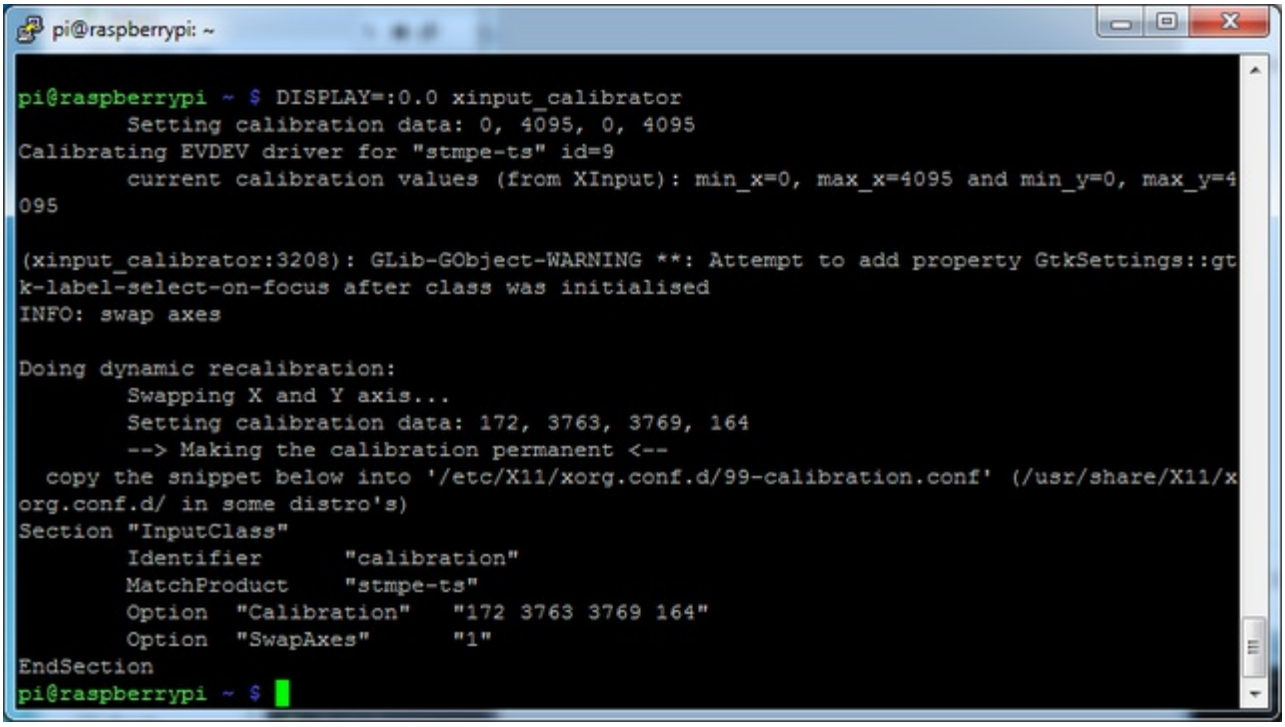
Now you'll have to run the xcalibrator while also running X. You can do this by **startx** and then opening up the terminal program and running the **xinput_calibrator** command (which is challenging to do on such a small screen) OR you can do what we do which is run startx in a SSH/Terminal shell and then run the xinput_calibrator from the same shell, which requires the following command order:

```
FRAMEBUFFER=/dev/fb1 startx &
DISPLAY=:0.0 xinput_calibrator
```

Follow the directions on screen



Once complete you'll get something like:



Run **sudo nano /etc/X11/xorg.conf.d/99-calibration.conf** and copy the

```
Section "InputClass"
        Identifier      "calibration"
        MatchProduct    "stmpe-ts"
        Option  "Calibration"   "172 3763 3769 164"
        Option  "SwapAxes"      "1"
EndSection
```

or whatever you got, into there. You can quit X if you want by typing**fg** to bring that command into the foreground, and then Control-C to quit.

Depending on the 'rotation' of the screen, when you do this calibration, you may need to comment out the SwapAxes part with a # and/or swap the numbers around so looks like:

```
Option "Calibration" "119 3736 3850 174"
```

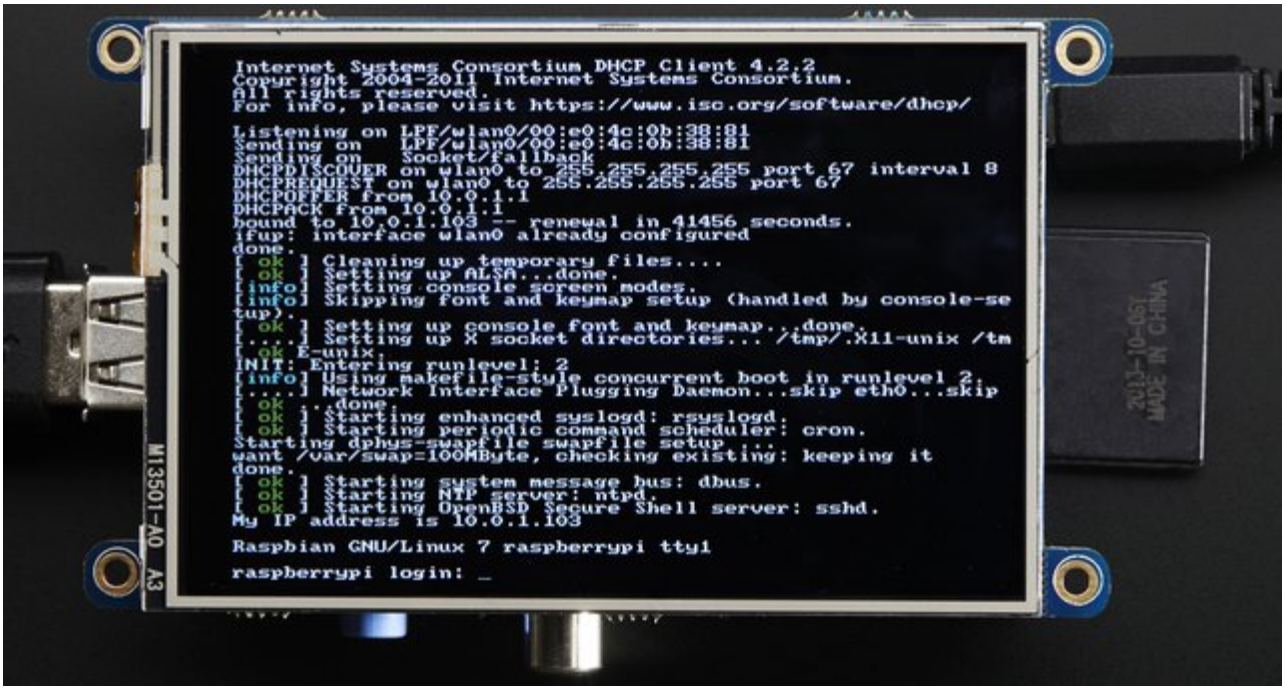
to

```
Option "Calibration" "3736 119 174 3850"
```

Your touchscreen is now super calibrated, hurrah!

Console Configuration

If you've grabbed our Easy Install image, or use the script, this step is not required, it's already done! This is just for advanced users who are curious on how to configure and customize the console



One fun thing you can do with the display is have it as your main console instead of the HDMI/TV output. Even though it is small, with a good font you can get 20 x 40 of text. For more details, check out https://github.com/notro/fbtft/wiki/Boot-console (http://adafru.it/cXQ)

First up, we'll update the boot configuration file to use the TFT framebuffer/**dev/fb1** instead of the HDMI/TV framebuffer /dev/fb0

```
sudo nano /boot/cmdline.txt
```

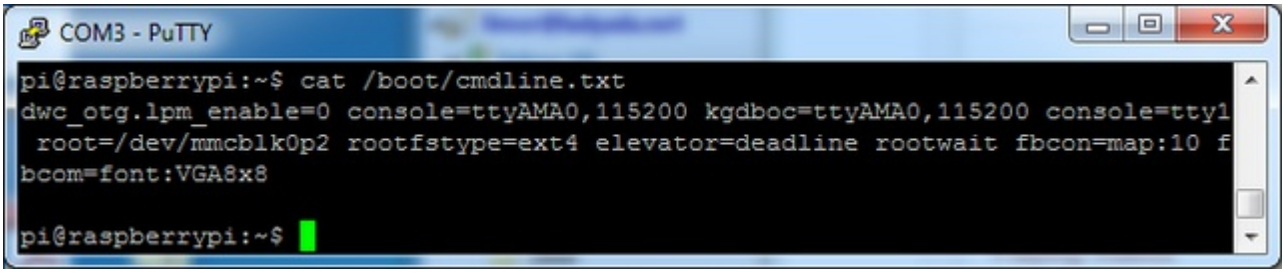
you can also edit it by putting the SD card into a computer and opening the same file.

At the end of the line, find the text that says`rootwait` and right after that, enter in: `fbcon=map:10 fbcon=font:VGA8x8` then save the file.

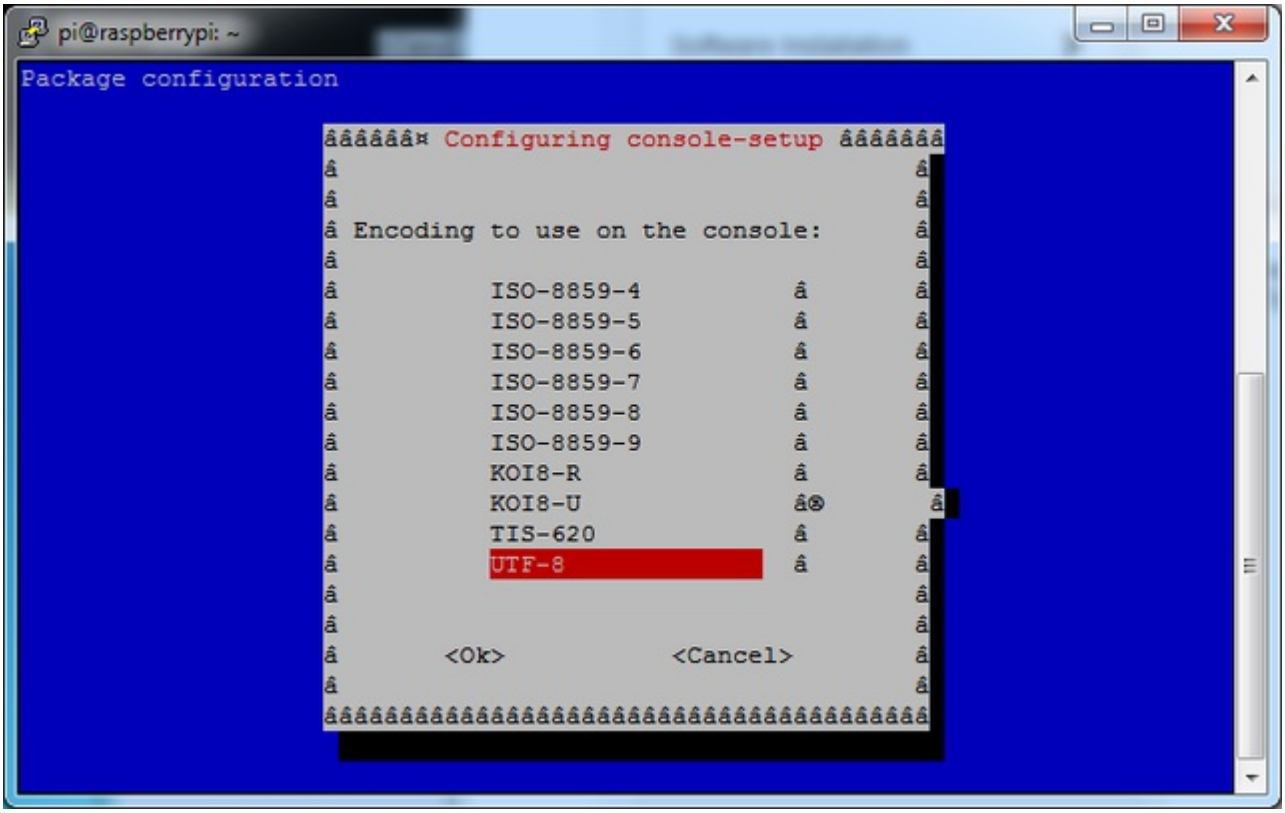
On the next boot, it will bring up the console.

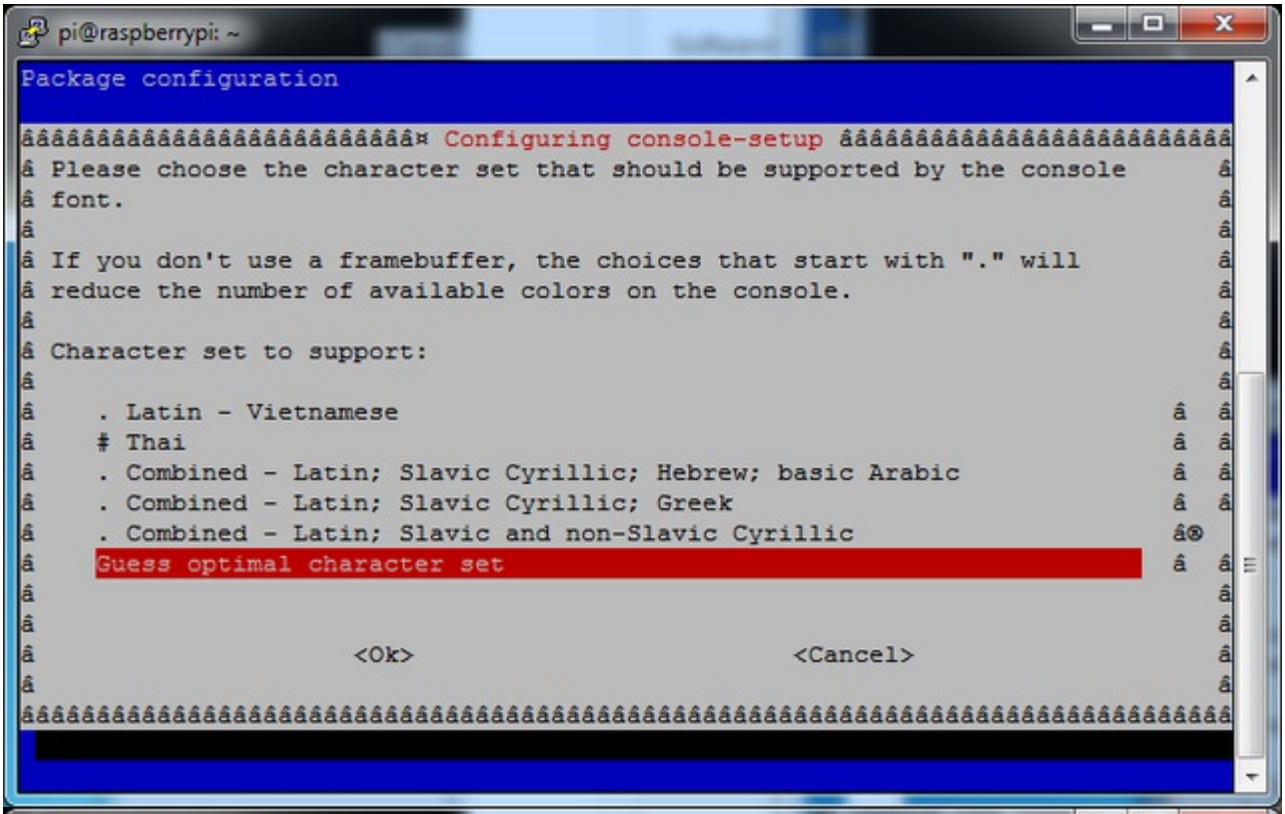
Note that the kernel has to load up the display driver module before it can display anything on it so you won't get the rainbow screen, a NooBs prompt, or a big chunk of

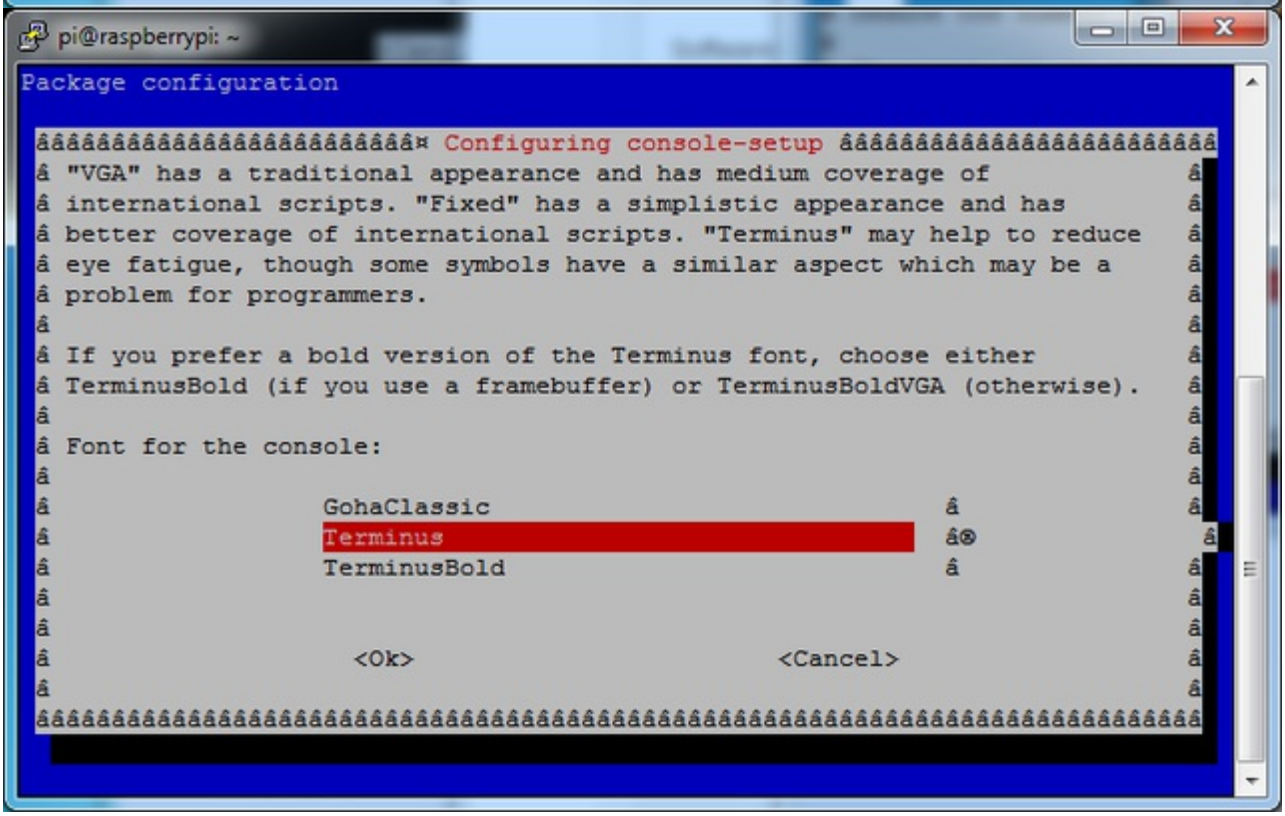
the kernel details since the module is loaded fairly late in the boot process.

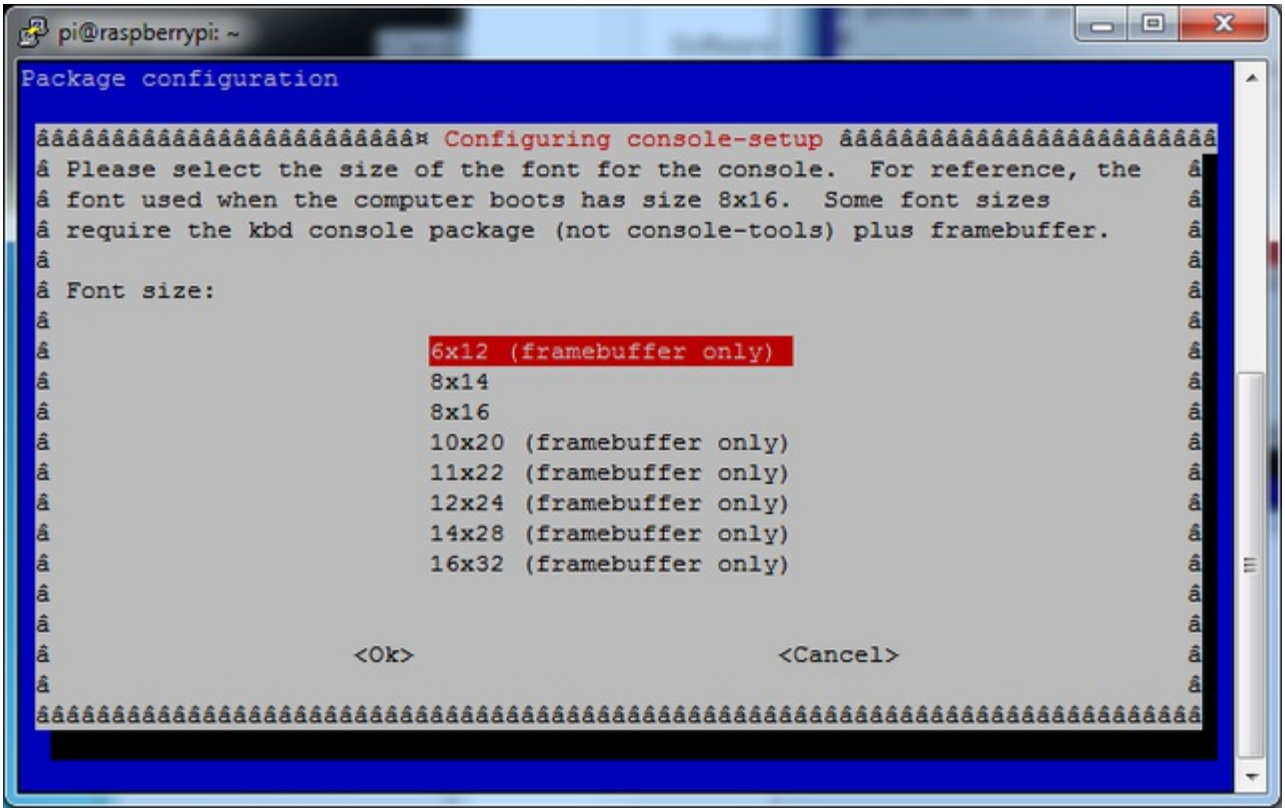


I think the VGA8x8 font is a bit chunky, you probably want 12x6 which is what is shown in the photo above. To change the font, run **sudo dpkg-reconfigure console-setup** and go thru to select Terminus 6x12




pi@raspberrypi: ~
Package configuration
Configuring console-setup
Please choose the character set that should be supported by the console font.
If you don't use a framebuffer, the choices that start with "." will reduce the number of available colors on the console.
Character set to support:
. Latin - Vietnamese
Thai
. Combined - Latin; Slavic Cyrillic; Hebrew; basic Arabic
. Combined - Latin; Slavic Cyrillic; Greek
. Combined - Latin; Slavic and non-Slavic Cyrillic
Guess optimal character set
<Ok>
<Cancel>


pi@raspberrypi: ~
Package configuration
Configuring console-setup
"VGA" has a traditional appearance and has medium coverage of international scripts. "Fixed" has a simplistic appearance and has better coverage of international scripts. "Terminus" may help to reduce eye fatigue, though some symbols have a similar aspect which may be a problem for programmers.
If you prefer a bold version of the Terminus font, choose either TerminusBold (if you use a framebuffer) or TerminusBoldVGA (otherwise).
Font for the console:
GohaClassic
Terminus
TerminusBold
<Ok>
<Cancel>



Turn off Console Blanking

You may notice the console goes black after 30 minutes, this is a sort of 'power saving' or 'screensaver' feature.

Raspbian Jessie

Add the following line to /etc/rc.local

```
sudo sh -c "TERM=linux setterm -blank 0 >/dev/tty0"
```

on the line before the final `exit 0`

Raspbian Wheezy

You can disable this by editing **/etc/kbd/config** and looking for

```
BLANK_TIME=30
```

and setting the blank time to 0 (which turns it off)

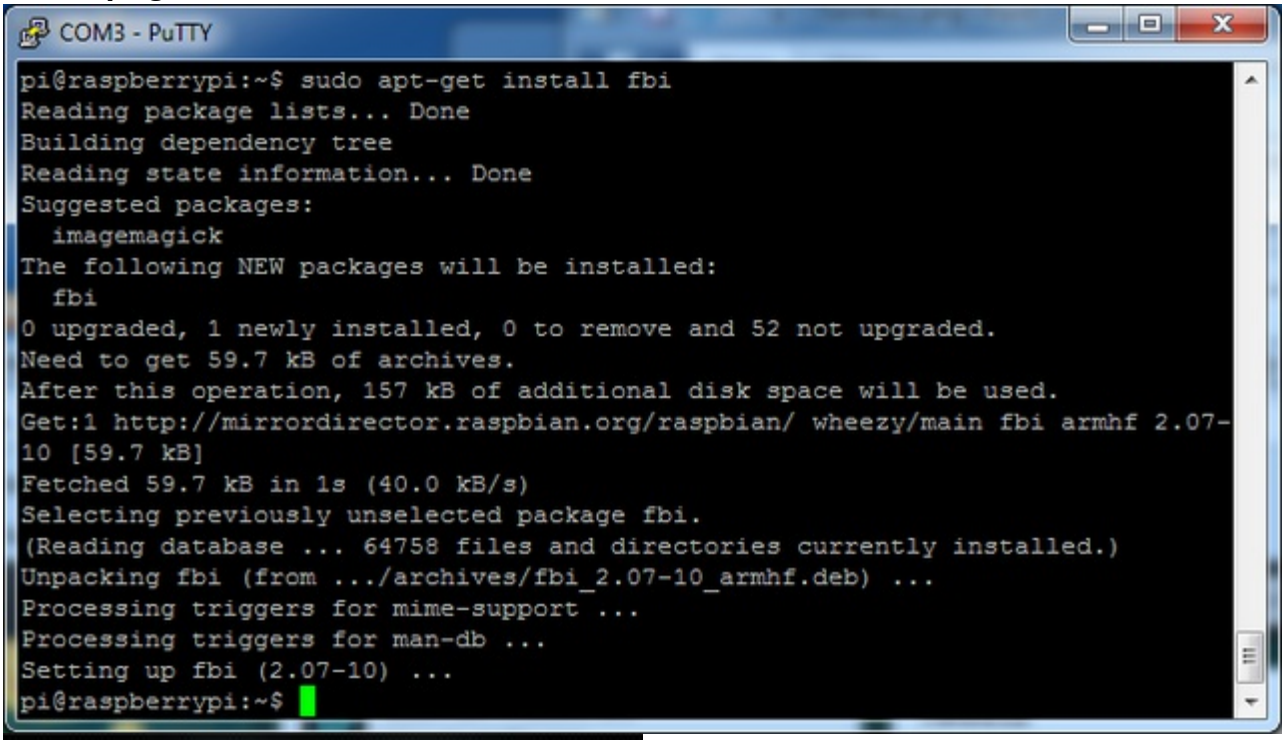
```
BLANK_TIME=0
```

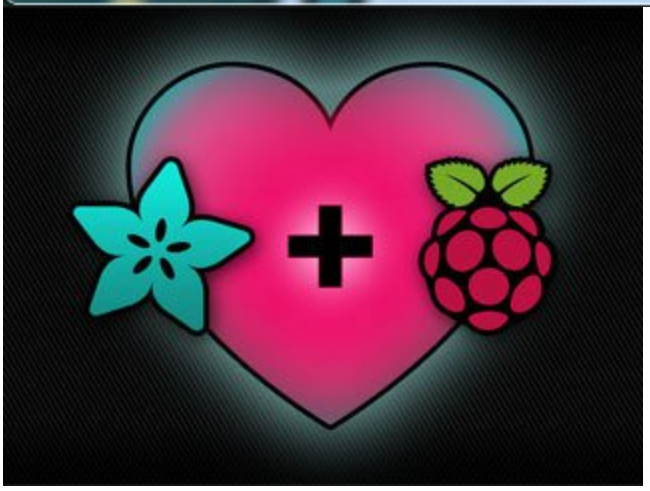
#16
#13
#12
#6
#5
Starting dphys-swapfile swapfile setup .
..
want /var/swap=100MByte, checking existi
ng: keeping it
done.
[ok] Starting NTP server: ntpd.
[....] Starting OpenBSD Secure Shell ser
[ok sshd.
Raspbian GNU/Linux 7 raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Mon Apr 27 17:51:08 UTC 2015
on tty1
Linux raspberrypi 3.18.8+ #2 PREEMPT Mon
Mar 9 14:05:46 UTC 2015 armv6l
The programs included with the Debian GN
U/Linux system are free software;
the exact distribution terms for each pr
ogram are described in the
individual files in /usr/share/doc/*/cop
yright.
Debian GNU/Linux comes with ABSOLUTELY N
O WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi ~ $ FRAMEBUFFER=/dev/fb1
startx -- -dpi 60_
HSD-8903J-D2

Displaying Images

You can display every day images such as GIFs, JPGs, BMPs, etc on the screen. To do this we'll install **fbi** which is the **frame buffer image** viewer (not to be confused with the FBI agency!)

sudo apt-get install fbi will install it





Grab our lovely wallpapers with

```
wget http://adafruit-download.s3.amazonaws.com/adapiluv320x240.jpg
wget http://adafruit-download.s3.amazonaws.com/adapiluv480x320.png (http://adafru.it/cXU)
```

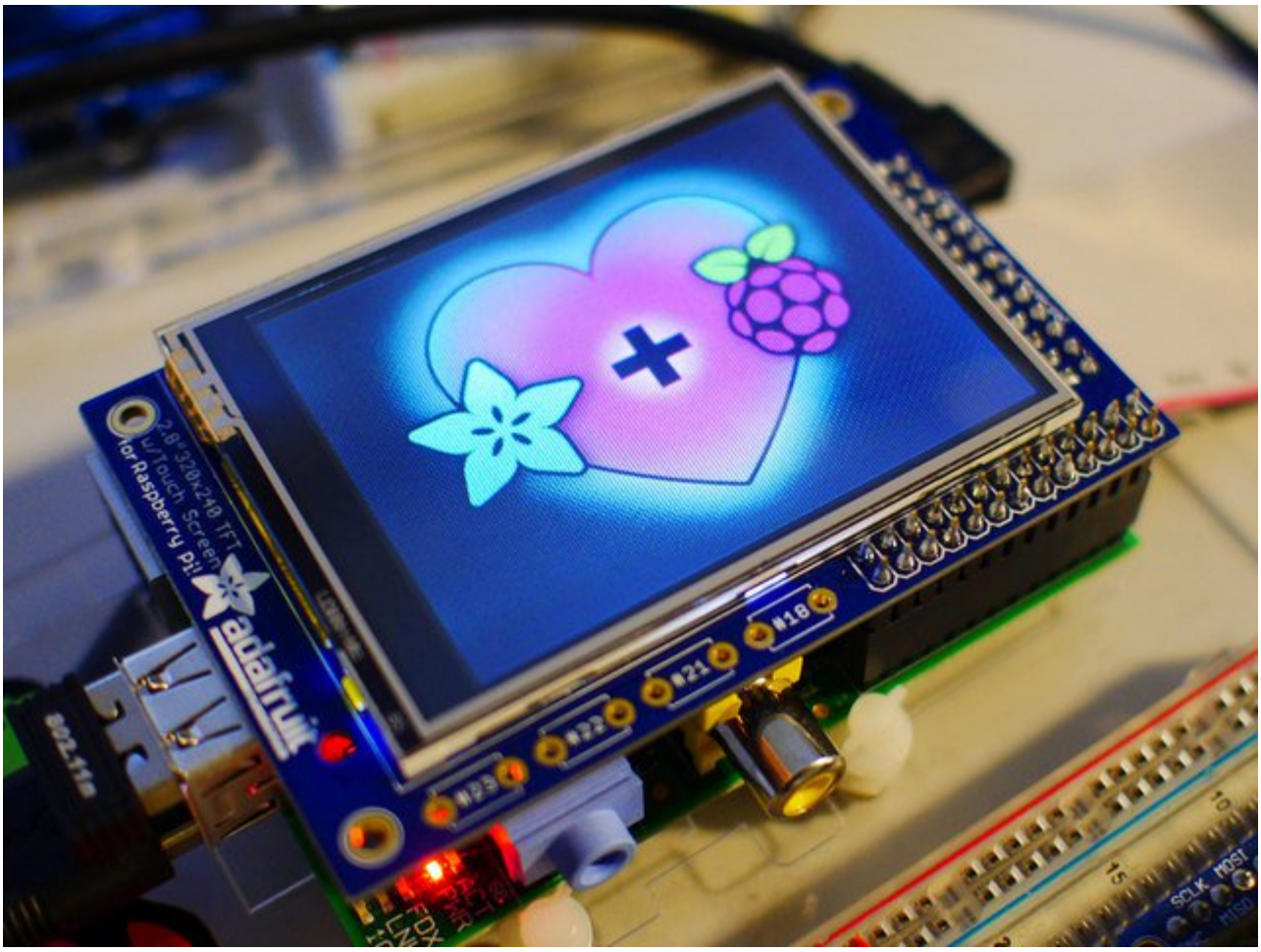
For 320x240 PiTFTs (2.2", 2.4", 2.8" or 3.2") view it with

```
sudo fbi -T 2 -d /dev/fb1 -noverbose -a adapiluv320x240.jpg
```

or for 3.5" PiTFTs:

```
sudo fbi -T 2 -d /dev/fb1 -noverbose -a adapiluv480x320 (http://adafru.it/cXU).jpg
```

That's it!



Playing Videos

How To Play Videos

You can play many types of videos on the screen, using mplayer you don't even need to run X and you can script the movies to play using Python. We'll show you how to just play one video for now.
To demo, we'll use an mp4 of Big Buck Bunny for 320 pixel wide screens. Below we show you how to create/resize videos, but to make it easy, just download our version with:

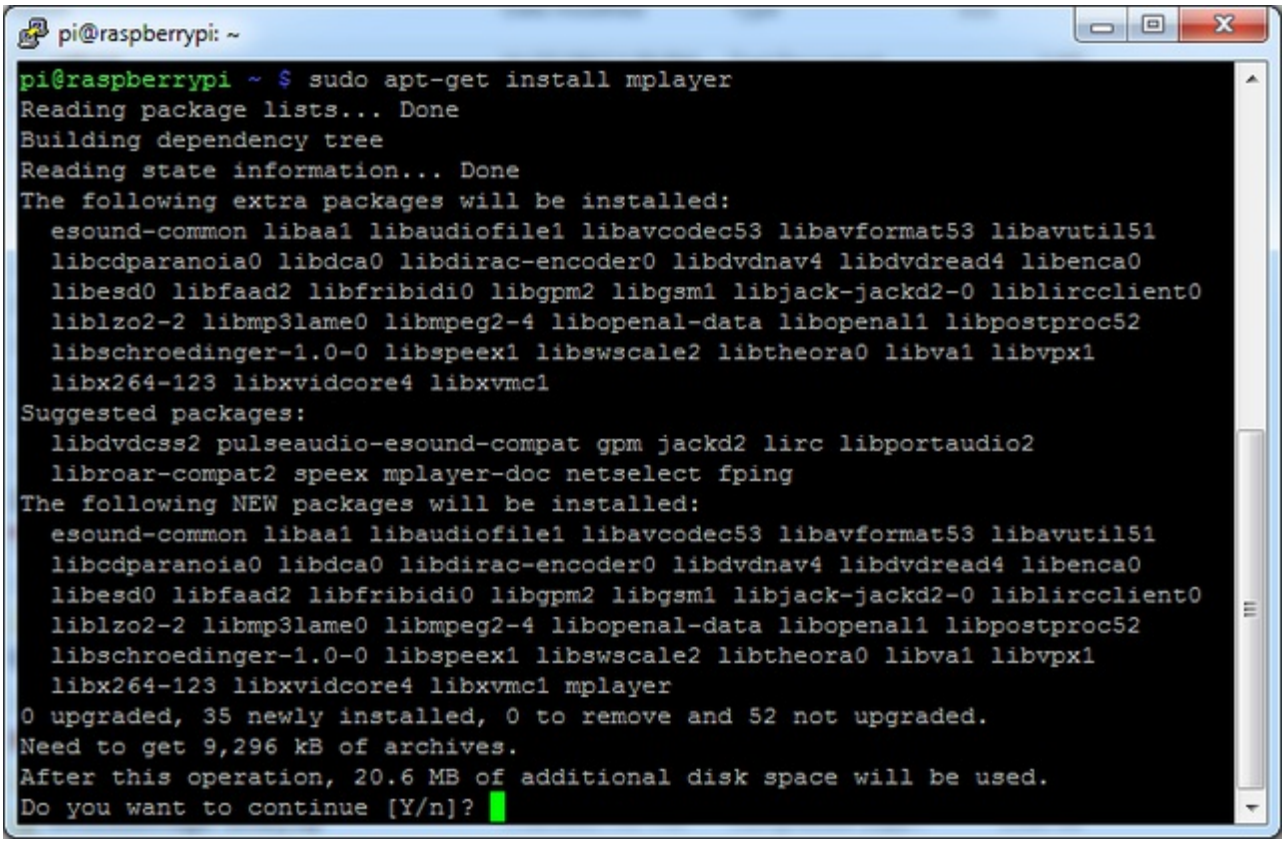
wget http://adafruit-download.s3.amazonaws.com/bigbuckbunny320p.mp4 (http://adafru.it/cXR)

The video is 30MB which is a lot if you haven't expanded your SD card yet. Before you do this, run sudo raspi-config to expand the SD card so you don't run out of space!

If you don't have **mplayer** yet, run

```
sudo apt-get update
sudo apt-get install mplayer
```

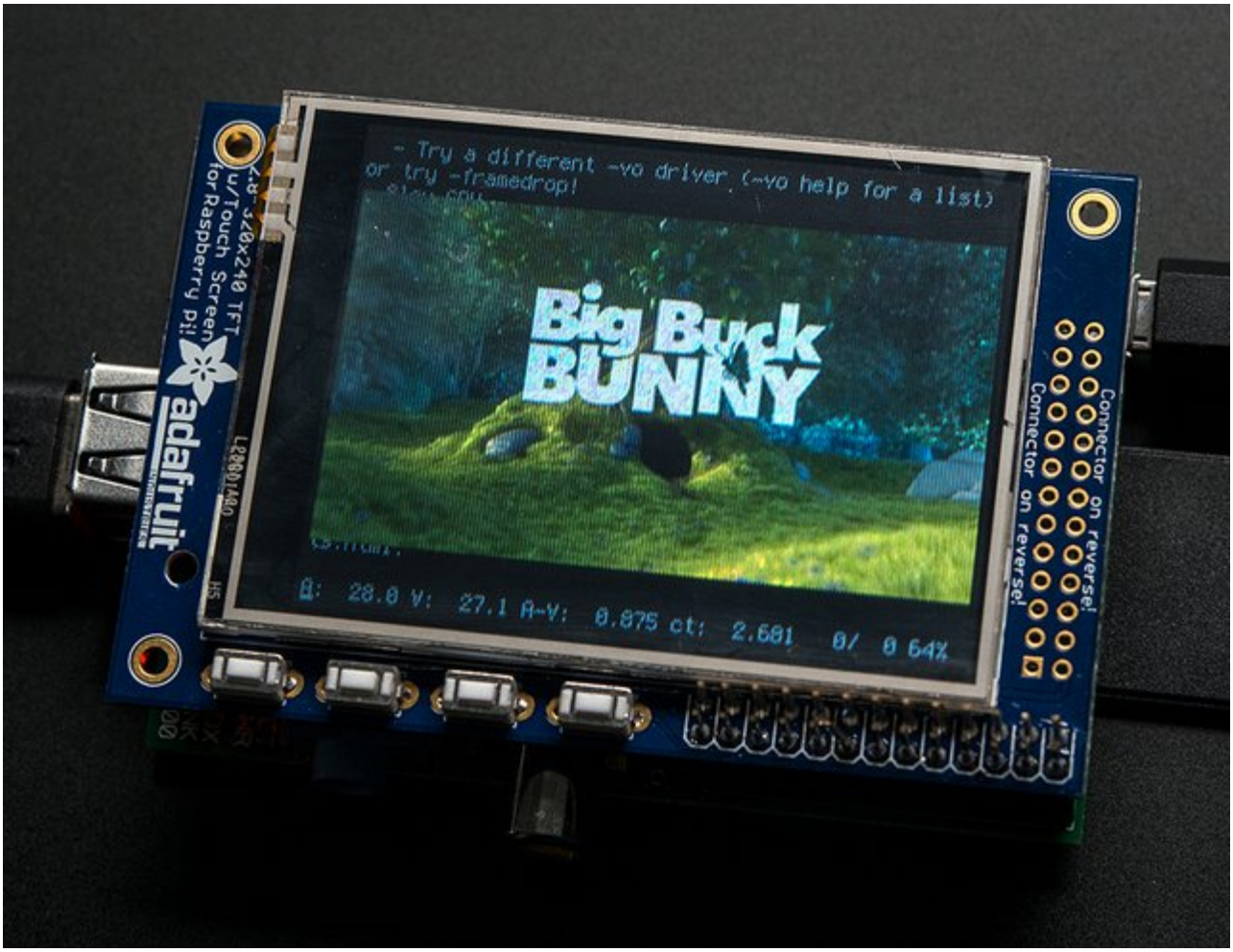
to install it. It may take a few minutes to complete



OK now you just have to run:

```
sudo SDL_VIDEODRIVER=fbcon SDL_FBDEV=/dev/fb1 mplayer -vo sdl -framedrop bigbuckbunny320p.mp4
```

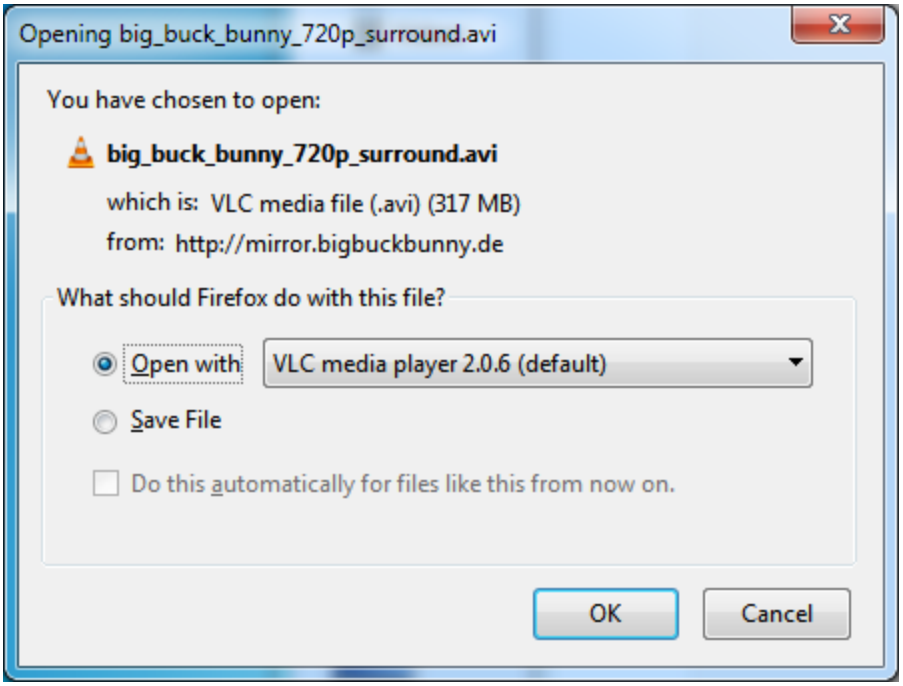
If your video is not sized for 320 wide, you may need to add a `-zoom` after `-framedrop` so that it will resize - note that this is quite taxing for the Pi, so it may result in a choppy or mis-synced video!



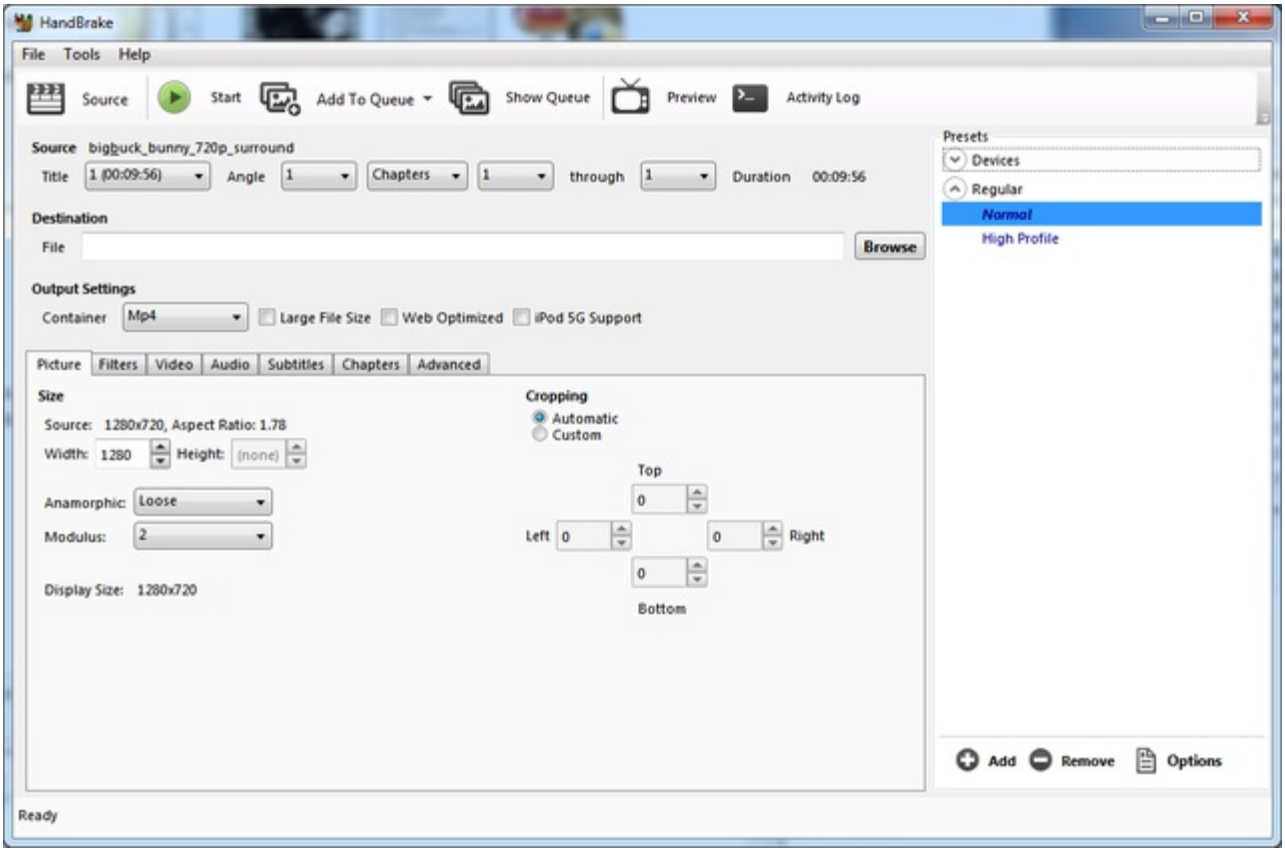
Converting/Resizing Videos

It's possible to play full length videos on the TFT plate, but since the screen is small and the Pi cant use hardware accelleration to play the videos its best to scale them down to 320x240 pixels. This will be easier for the Pi to play and also save you tons of storage space. For this demo, we'll be using the famous Big Buck Bunny (http://adafru.it/cXS) video, which is creative commons and also very funny!

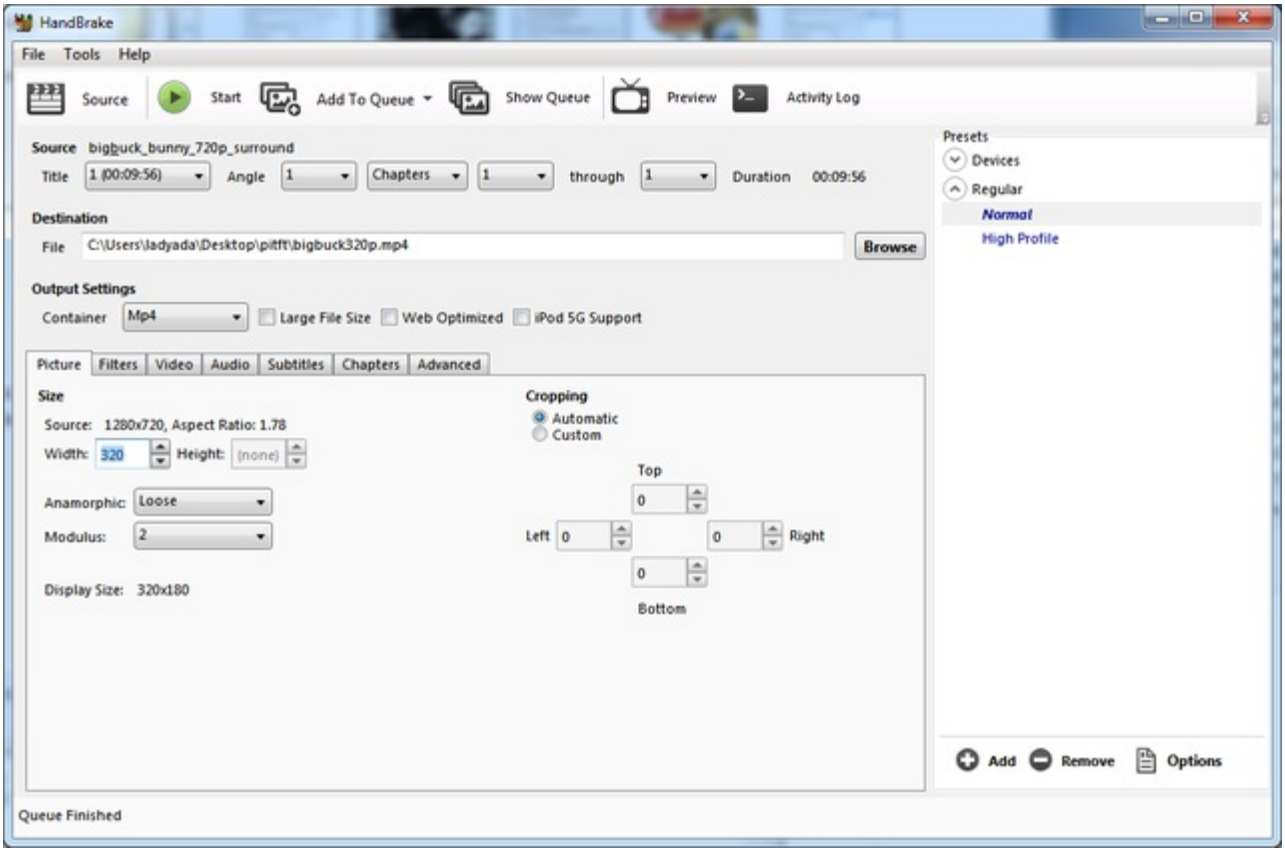
You can download it from the link above, we'll be using the 720p AVI version.



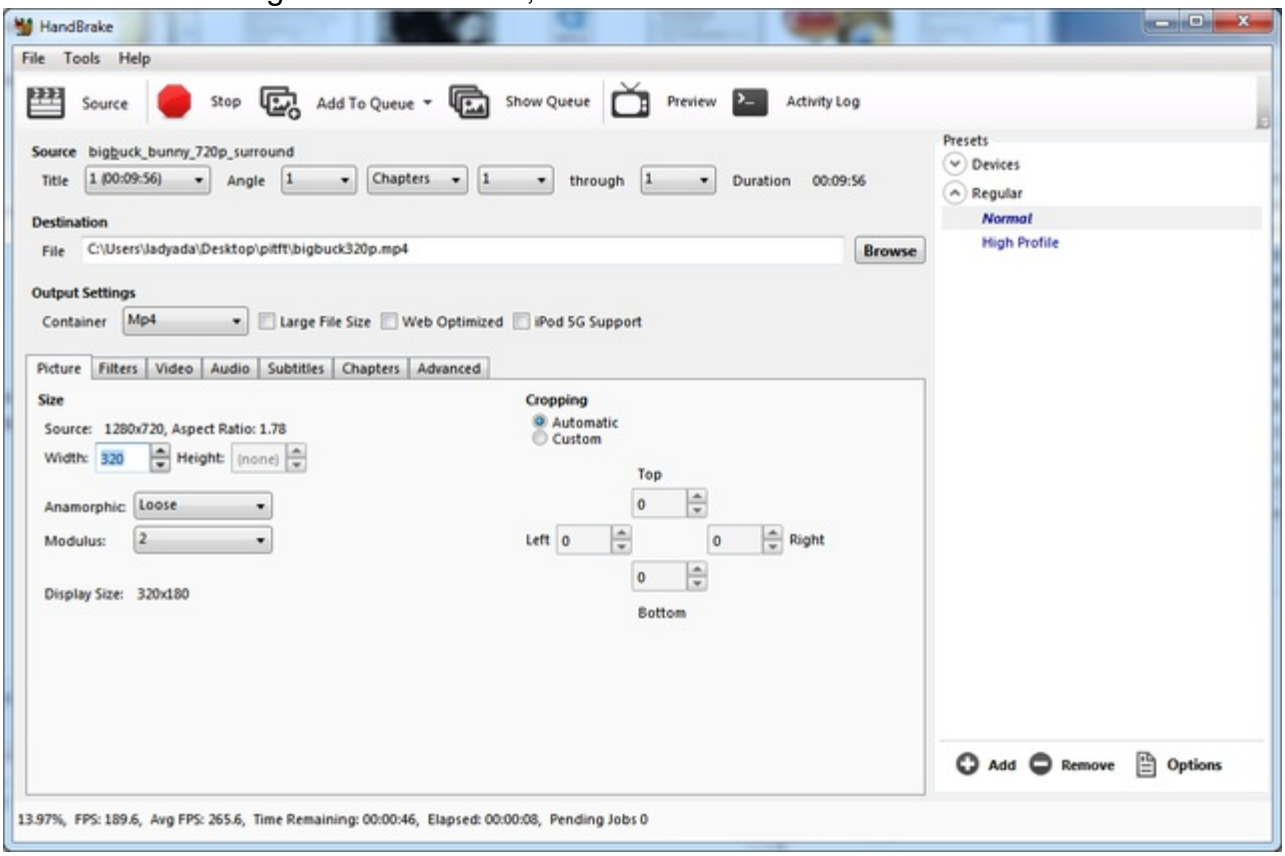
To do the conversion itself, we suggest [HandBrake](http://adafru.it/cXT) which works great and is open source so it runs on all operating systems! Download and install from the link. Then run the installed application and open up the AVI file from before. The app will pre-fill a bunch of information about it.



Under **Destination** click **Browse...** to select a new MP4 file to save. Then under**Picture** change the **Width** to 320 (the height will be auto-calculated)



Click **START** to begin the conversion, it will take a minute or two.



That's it! You now have a smaller file. Don't forget to play it on your computer to make sure it

plays right before copying it to your Pi

Using FBCP

The Ideal: Adafruit's PiTFT displays are razor sharp. Whereas small composite screens on the Raspberry Pi usually require some video scaling (resulting in blurriness), PiTFT uses the GPIO header, digitally controlled pixel-by-pixel for a rock steady image. Though not a *lot* of pixels, it works great for retro gaming (and the display neatly stacks above the board, no side protuberances for video cables).

The Downside: this GPIO link entirely bypasses the Pi's video hardware, including the graphics accelerator. Many games and emulators *depend* on the GPU for performance gains. So the PiTFT has traditionally been limited to just a subset of specially-compiled emulators that can work and run well enough without the GPU.

The Solution: our latest PiTFT drivers, along with a tool called *fbcp* (framebuffer copy), careful system configuration, and (optionally) the more potent Raspberry Pi 2 board open the doors to many more gaming options. Existing emulator packages (such as RetroPie, with *dozens* of high-performance emulators and ports) — previously off-limits to the PiTFT — can run quite effectively now!

Click here to go to our FBCP tutorial!
http://adafru.it/fbe

Backlight Control

The backlight of the 2.8" PiTFT has 4 LEDs in series and it draws ~75mA at all times, controlled by a transistor. The PiTFT 3.5" display has 6 LEDs in a row, and we use a boost converter to get the 5V from the Pi up to the ~20V needed to light up all the LEDs.

There might be times you'd like to save some power and turn off the backlight. The screen and touchplate will still work, you just can't see anything. We designed the board with the STMPE610 touchscreen controller which has 2 extra GPIO and tied one of them to control the backlight. You can use the command line to control the backlight.

By default, the backlight's on...but you can control it in two ways!

PWM Backlight Control with GPIO 18

If you want precise control, you can use the PWM output on GPIO 18. There's python code for controlling the PWM but you can also just use the kernel module and shell commands.

You'll need to make sure the STMPE control is not 'active' as the STMPE GPIO overrides the PWM output.

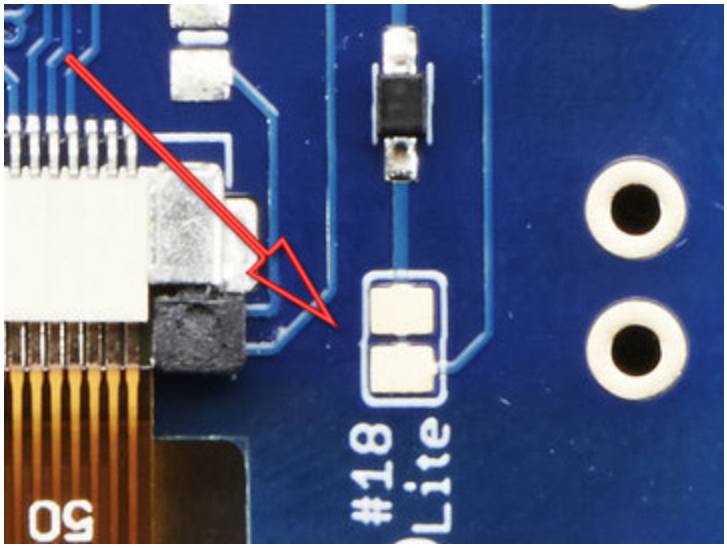
```
sudo sh -c 'echo "1" > /sys/class/backlight/soc\:backlight/brightness'
```

(Or if you are running an old kernel before the backlight object, try **sudo sh -c "echo 'in' > /sys/class/gpio/gpio508/direction"**)

OK now you can set the GPIO #18 pin to PWM mode using WiringPi's **gpio** command

With these basic shell commands, you can set the GPIO #18 pin to PWM mode with 1000 Hz frequency, set the output to 100 (out of 1023, so dim!), set the output to 1023 (out of 1023, nearly all the way on) and 0 (off)

```
gpio -g mode 18 pwm
gpio pwmc 1000
gpio -g pwm 18 100
gpio -g pwm 18 1023
gpio -g pwm 18 0
```

- 

If you'd like to not have #18 control the backlight, simply cut the solder jumper, the tiny trace between the two large gold pads marked **Lite #18**

On / Off Using STMPE GPIO

Another option is to just turn it on and off using the extra GPIO created by the touchscreen driver

Thanks to the raspberry Pi overlay system, this GPIO is already set up for you in a file called **/sys/class/backlight/soc:backlight/brightness**

To turn the backlight off run

```
sudo sh -c 'echo "0" > /sys/class/backlight/soc\:backlight/brightness'
```

To turn it back on, run

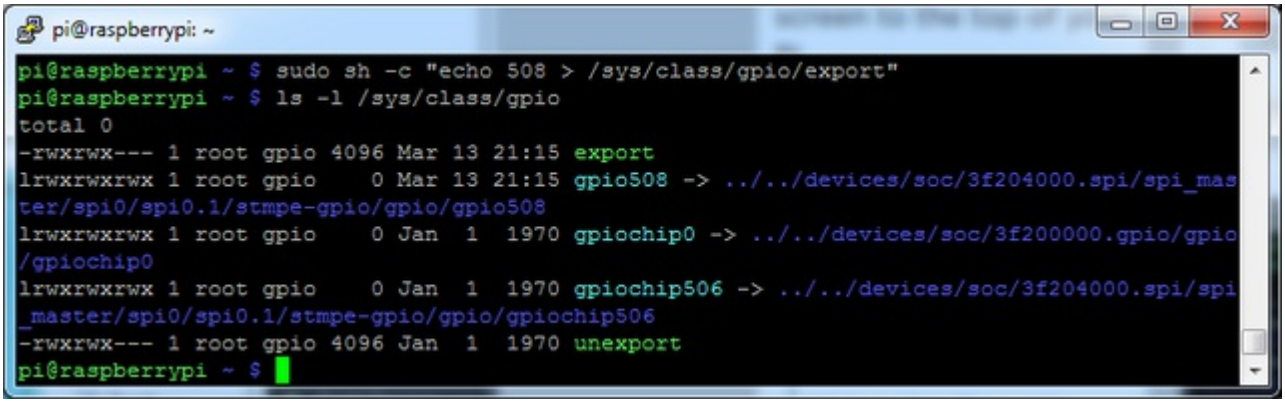
```
sudo sh -c 'echo "1" > /sys/class/backlight/soc\:backlight/brightness'
```

For older versions of PiTFT Kernel

On older versions of the PiTFT kernel/overlay, the GPIO was not tied to the backlight device. Start by getting access to the GPIO by making a device link

```
sudo sh -c "echo 508 > /sys/class/gpio/export"
ls -l /sys/class/gpio
```

For some *really* old versions, the GPIO pin was #252 not #508 so substitute that if you're running something from 2014 or earlier



Once you verify that you see GPIO #508, then you can set it to an output, this will turn off the display since it will output 0 by default

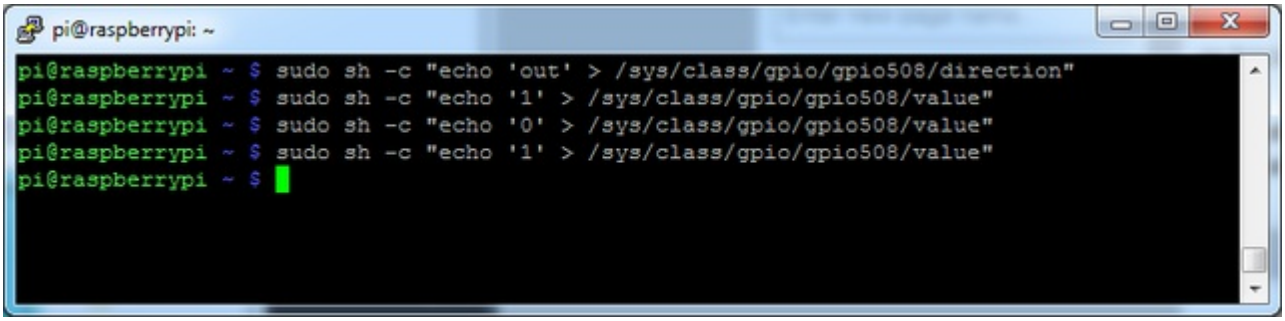
sudo sh -c "echo 'out' > /sys/class/gpio/gpio508/direction"

Then turn the display back on with

sudo sh -c "echo '1' > /sys/class/gpio/gpio508/value"

or back off

sudo sh -c "echo '0' > /sys/class/gpio/gpio508/value"

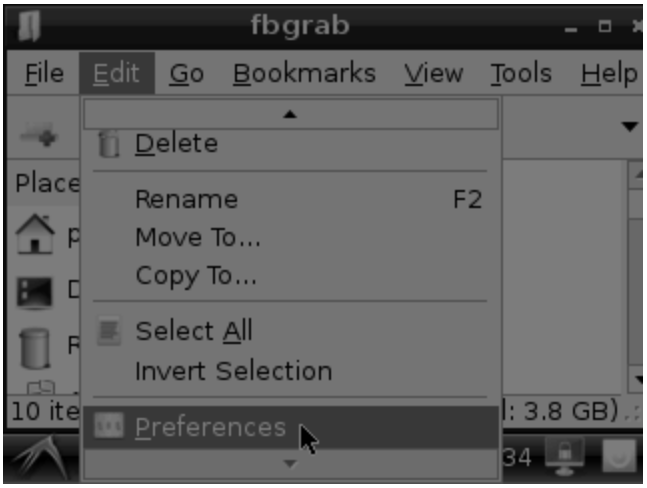


More Tips

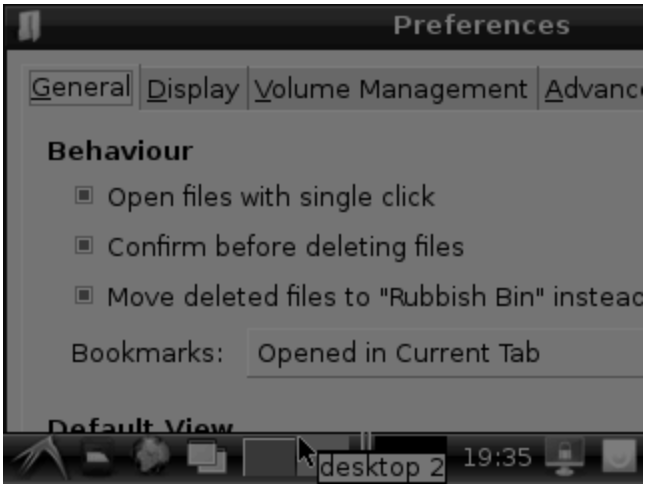
Making it easier to click icons in X

If you want to double-click on icons to launch something in X you may find it annoying to get it to work right. In LXDE you can simply set it up so that you only need to single click instead of double.
From LXDE launch the file manager (sorry these pix are grayscale and from the 2.8" TFT, still figuring out how to screenshot the framebuffer!)

Then under the **Edit** menu, select **Preferences**



Then select **Open files with single click** and close the window (you'll need to drag it over to get to the X button



Right-click on a touchscreen

Obviously if you have a touchscreen, it cannot tell what finger you are pressing with. This means that all 'clicks' are left clicks. But if you want a right-click, you *can* do it.

Just add the following lines into your InputClass of/**etc/X11/xorg.conf.d/99-calibration.conf** after the calibration section

```
Option "EmulateThirdButton" "1"
Option "EmulateThirdButtonTimeout" "750"
Option "EmulateThirdButtonMoveThreshold" "30"
```

So for example your file will look like:

```
Section "InputClass"
   Identifier      "calibration"
   MatchProduct    "stmpe-ts"
   Option  "Calibration"   "3800 120 200 3900"
   Option  "SwapAxes"      "1"
   Option "EmulateThirdButton" "1"
   Option "EmulateThirdButtonTimeout" "750"
   Option "EmulateThirdButtonMoveThreshold" "30"
EndSection
```

This makes a right mouse click emulated when holding down the stylus for 750 ms.

(Thx adamaddin! (http://adafru.it/fH3))

Gesture Input

With the PiTFT touchscreen and xstroke (http://adafru.it/dD0) you can enter text in applications by drawing simple character gestures on the screen! Check out the video below for a short demonstration and overview of gesture input with xstroke:

Installation

Unfortunately xstroke hasn't been actively maintained for a few years so there isn't a binary package you can directly install. However compiling the tool is straightforward and easy with the steps below. Credit for these installation steps goes to mwilliams03 at ozzmaker.com (http://adafru.it/dD1).

First install a few dependencies by opening a command window on the Pi and executing:

```
sudo apt-get -y install build-essential libxft-dev libxpm-dev libxtst-dev
```

Now download, compile, and install xstroke by executing:

```
cd ~
wget http://mirror.egtvedt.no/avr32linux.org/twiki/pub/Main/XStroke/xstroke-0.6.tar.gz
tar xfv xstroke-0.6.tar.gz
cd xstroke-0.6
./configure
sed -i '/^X_LIBS = / s/$/ -lXrender -lX11 -lXext -ldl/' Makefile
make
sudo make install
```

If the commands above execute successfully xstroke should be installed. If you see an error message, carefully check the dependencies above were installed and try again.

Once xstroke is installed you will want to add a couple menu shortcuts to start and stop xstroke. Execute the following commands to install these shortcuts:

```
wget https://github.com/adafruit/PiTFT_Extras/raw/master/xstroke.desktop
wget https://github.com/adafruit/PiTFT_Extras/raw/master/xstrokekill.desktop
sudo cp xstroke*.desktop /usr/share/applications/
```

Usage

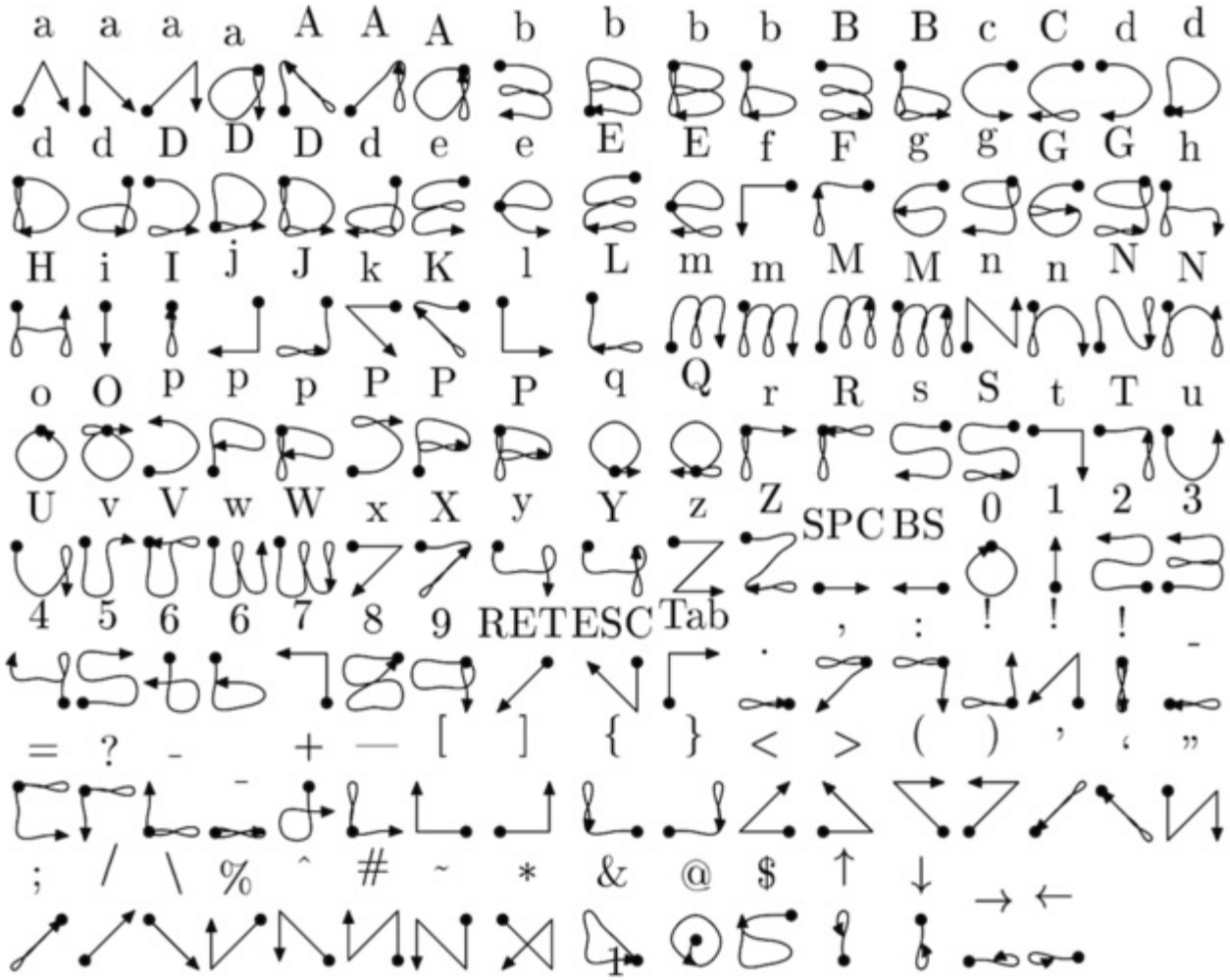
To use xstroke I highly recommend using a plastic stylus instead of your finger. Also calibrate the touchscreen for X-Windows (http://adafru.it/dD2) so you have the best control over the cursor possible.

Don't use a ballpoint pen or sharp metal stylus as it could scratch or damage the touchscreen! Start X-Windows on the PiTFT and open the LXDE menu by clicking the icon in the lower left corner. Scroll up to the **Accessories** menu at the top and notice the new**XStroke** and **XStroke Kill** commands.

Click the **XStroke** menu option to start xstroke. You should see a small pencil icon appear on the bottom right side of the screen. The pencil icon means xstroke is running, however by default it's not yet looking for gesture input.

Open an application that takes text input, such as LXTerminal. To enable gesture input click the xstroke pencil icon. You should see the pencil turn green and the text 'abc' written over top of the icon. You might need to click the icon a few times to get the click to register in the right spot.

When xstroke is looking for gesture input you can drag the mouse cursor in a gesture anywhere on the screen to send specific key strokes. Here's a picture of the possible gestures you can send:



(credit to Carl Worth for the image above)

To draw a gesture from the above image, press anywhere on the screen, start from the circle in the gesture, and follow the gesture pattern towards the arrow. As you draw a gesture you should see a blue line displayed that shows what you've drawn. Lift up the stylus when you get to the end of the gesture at the arrow. If xstroke recognizes the gesture it will send the appropriate key press to the active window. Try drawing a few characters from the image above to get the hang of writing gestures.

A few very useful gestures are backspace (which deletes a character), return/enter, and space. To draw a backspace gesture just draw a line going from the right side of the screen to the left side. The gesture for return/enter is a diagonal line from the top right to bottom left. Finally a space is a straight line from the left to the right.

Note that when xstroke is looking for gestures you might not be able to click or control the cursor as you normally would expect. To stop xstroke's gesture recognition carefully press the xstroke pencil icon again until the 'abc' text disappears. I've found this process can be a little

finicky as the icon is very small and any movement will be interpreted as a gesture. Use a light touch and try a few times to click the icon.

If you get stuck completely and can't disable xstroke by clicking the icon, connect to the Raspberry Pi in a terminal/SSH connection and run 'killall xstroke' (without quotes) to force xstroke to quit. The normal way to stop xstroke is to navigate to the **Accessories** -> **XStroke Kill** command, but you might not be able to do that if xstroke is listening for gesture input.

Have fun using xstroke to control your Pi by writing gestures on the PiTFT screen!

PiTFT PyGame Tips

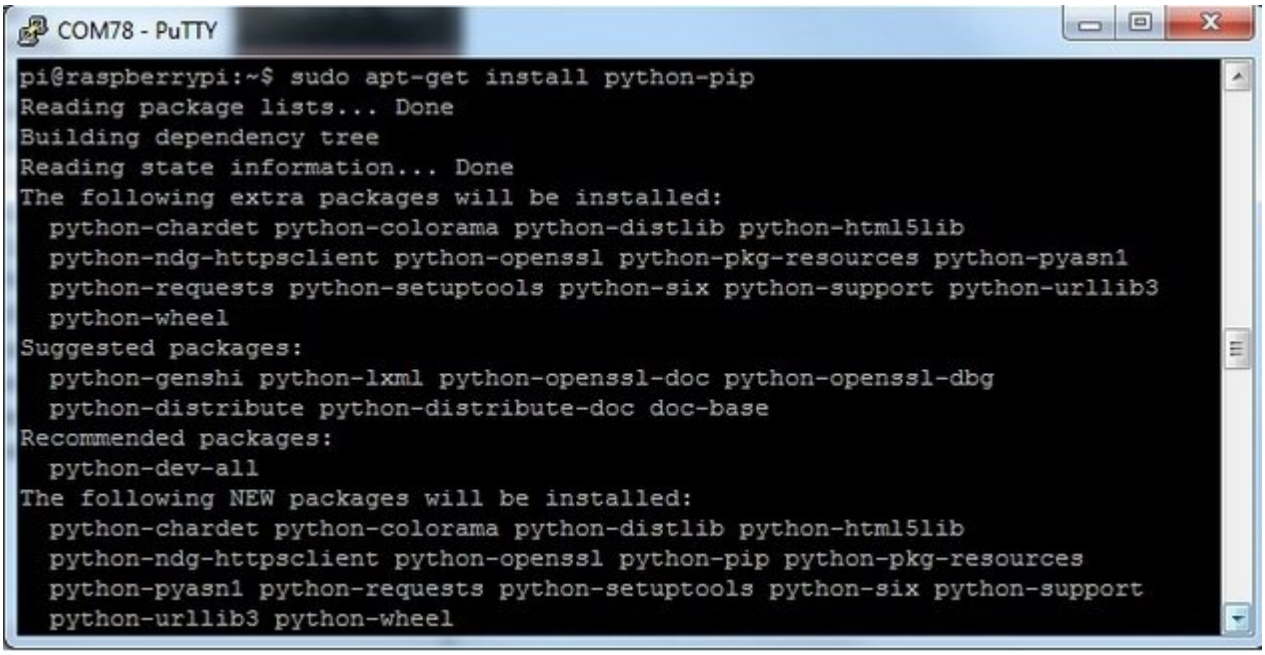
Since the PiTFT screen is fairly small, you may need to write custom UI programs. Pygame is the easiest way by far to do this.

Jeremy Blythe has an excellent tutorial here on getting started.(http://adafru.it/saw)

However, *before* you follow that link you'll want to set up pygame for the best compatibility:

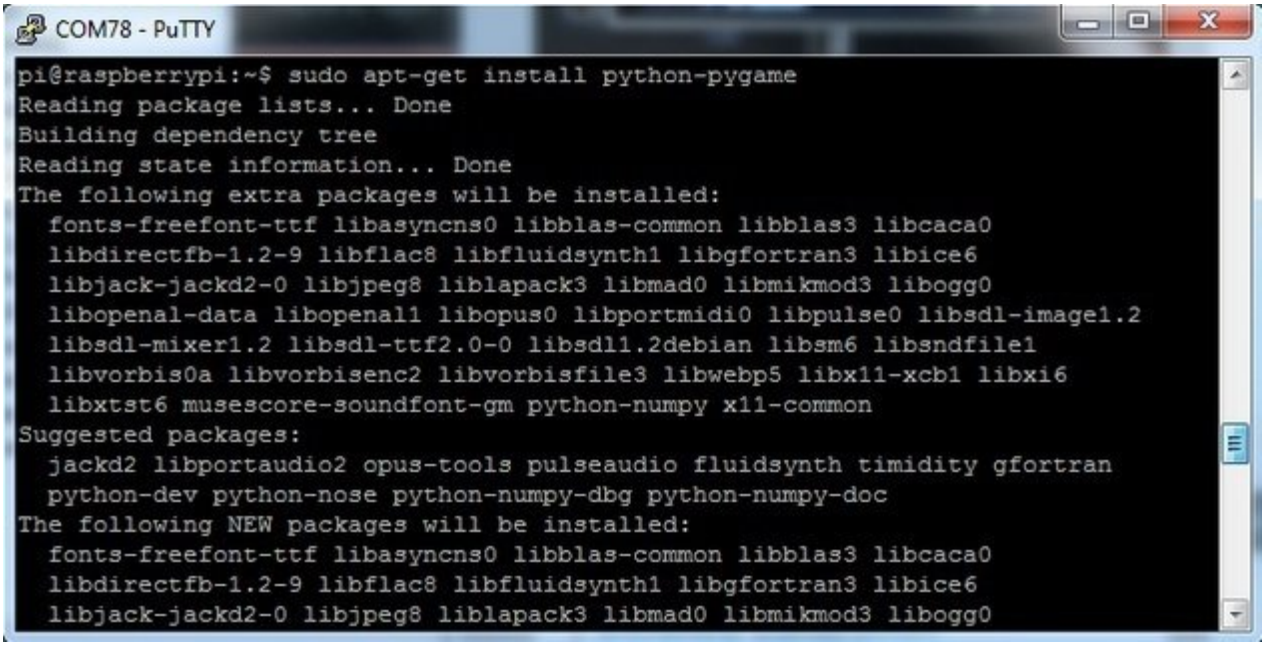
Install pip & pygame

Install Pip: **sudo apt-get install python-pip**



Install Pygame: **sudo apt-get install python-pygame**

(this will take a while)



Ensure you are running SDL 1.2

SDL 2.x and SDL 1.2.15-10 have some serious incompatibilities with touchscreen. You can force SDL 1.2 by running a script. (Thanks to heine in the forums!(http://adafru.it/sax))

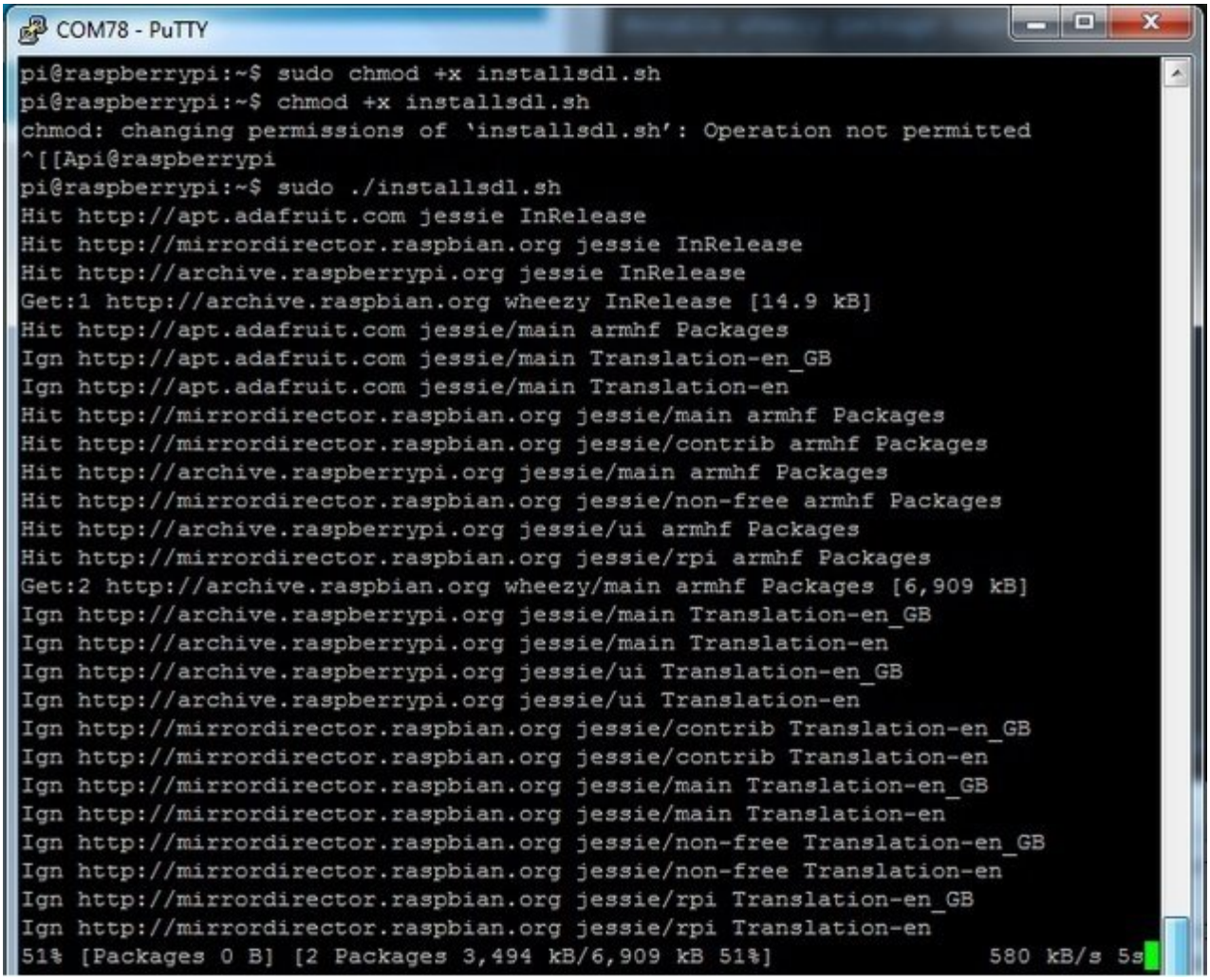
Edit a new file with **sudo nano installsdl.sh**
and paste in the following text:

```
#!/bin/bash

#enable wheezy package sources
echo "deb http://archive.raspbian.org/raspbian wheezy main
" > /etc/apt/sources.list.d/wheezy.list

#set stable as default package source (currently jessie)
echo "APT::Default-release \"stable\";
" > /etc/apt/apt.conf.d/10defaultRelease

#set the priority for libsdl from wheezy higher then the jessie package
echo "Package: libsdl1.2debian
Pin: release n=jessie
Pin-Priority: -10
Package: libsdl1.2debian
Pin: release n=wheezy
Pin-Priority: 900
" > /etc/apt/preferences.d/libsdl

#install
apt-get update
apt-get -y --force-yes install libsdl1.2debian/wheezy
```

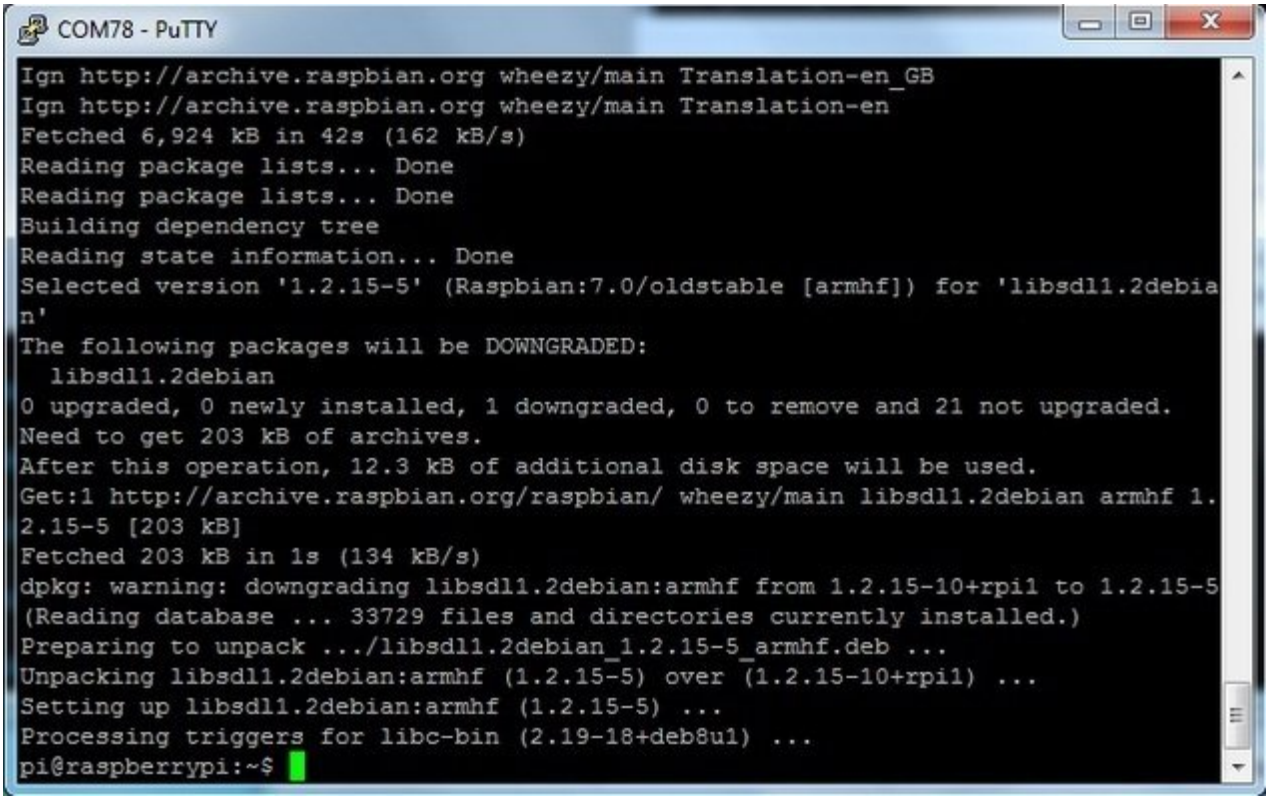
run

sudo chmod +x installsdl.sh

sudo ./installsdl.sh



it will force install SDL 1.2



OK **now** you can continue with pygame

HELP! (FAQ)

My PiTFT used to work, now it doesn't!

Did you do an apt-get upgradeor rpi-update? This command will blow away our PiTFT kernel which means that you will no longer have PiTFT support, you will have to redo the **easy-install** steps to reinstall the kernel.

If you had already made a working PiTFT setup, you may be able to reinstall the Adafruit kernel like so:

```
sudo apt-get install raspberrypi-kernel=1.20161027-1
```

If it tells you that the latest version is already installed, try this instead:

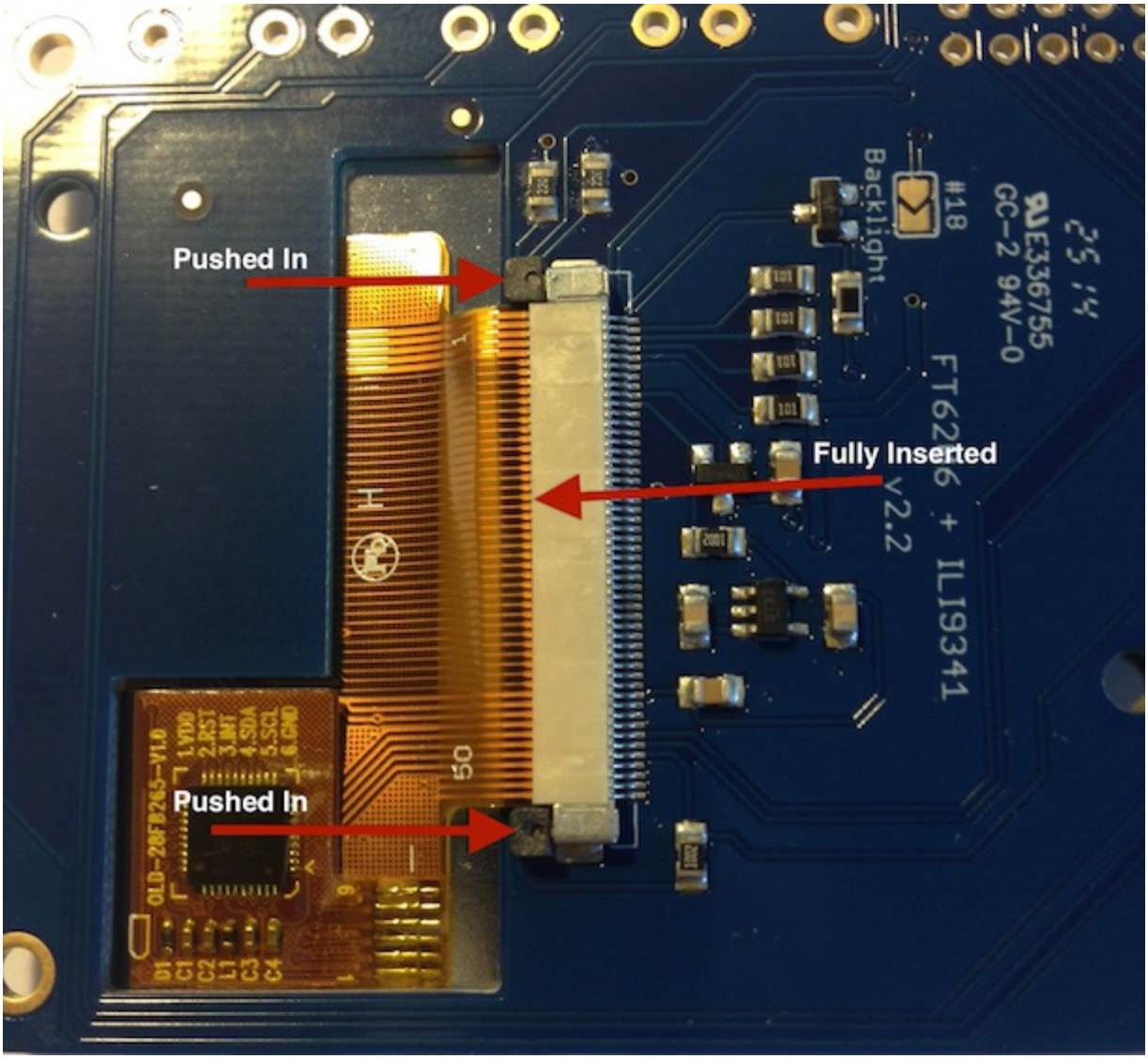
```
sudo apt-get install --reinstall raspberrypi-kernel=1.20161027-1
```

...you can check here (http://adafru.it/eUK) and substitute the most recent version you see in the =1.20161027-1 part.

I'm booting my Pi with the PiTFT and the HDMI output 'locks up' during boot!

It looks like the Pi is 'halting' or 'locking' up during boot but what is really happening is the console is switching from the HDMI output to the PiTFT output.

Check your PiTFT connections, particularly make sure you seated the PiTFT on the Pi properly, nothing is in the way, and the TFT flex connector is seated properly.



My PiTFT works for a bit and then I get a black screen with a short line of white pixels in one corner

Sounds like you tried to configure your Pi to 'boot straight to X', that is, start up the graphics interface on boot. This doesn't work by default because the Pi operating system is not expecting a PiTFT so it boots to the HDMI output. See below for how to set up your Pi to boot to X on the PiTFT

To 'fix' this, you can either connect an HDMI monitor, then in a terminal window run **sudo raspi-config** and configure the Pi to boot to the command line not X! If you do not have an HDMI monitor, you can also try a console cable

How can I force the Pi to bring up X on the HDMI/TV monitor?

There's two ways to do it. In older Pi installs, use the **fb0** framebuffer when you want to display stuff on the HDMI/TV display, for example:
FRAMEBUFFER=/dev/fb0 startx

will use the HDMI/TV framebuffer for X windows instead of the PiTFT

On Jessie Pi installs, run
sudo nano /usr/share/X11/xorg.conf.d/99-fbdev.conf
to edit the configuration file and make sure it contains:

```
Section "Device"
  Identifier "display"
  Driver "fbdev"
  Option "fbdev" "/dev/fb0"
EndSection
```

change the Option "fbdev" "/dev/fb0" line to Option "fbdev" "/dev/fb1" if you want the xdisplay on the PiTFT

I'm tring to run startx and I get FATAL: Module g2d_23 not found.
don't forget you have to remove the turbo file!
sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~
How come OMX-Player and Minecraft and other programs don't appear on the PiTFT display?

Some programs are graphics-optimized, particularly the video playback tools and some other programs like Minecraft. They write 'directly' to the HDMI output, and cannot write to the PiTFT so there is no way to directly make them work. However, you *can* have the output go to HDMI and then mirror the HDMI onto the PiTFT with **fbcp**. See this tutorial for more details (http://adafru.it/fbe)

Why doesn't the tactile button on GPIO #21 work?
On some older PiTFTs we had one of the buttons labeled #21 - that's the original RasPi name for that pin. If you're using a V2 (chance is, you are!) that is now called #27.
All the PiTFT's we ship now have the button labeled #21 and #27
I want better performance and faster updates!

You can change the SPI frequency (overclock the display) by editing **/boot/config.txt** and changing the **dtoverlay** options line to:

dtoverlay=pitft28r,rotate=90,speed=62000000,fps=25

Or whatever you like for speed, rotation, and frames-per-second. BUT, here's the thing, the Pi only supports a *fixed number* of SPI frequencies. So tweaking the number a little won't do anything. The kernel will round the number to the closest value. You will always get frequencies that are 250MHz divided by an even number. Here's the only SPI frequencies this kernel supports

- 15,625,000 (a.k.a 16000000 = 16 MHz)
- 17,857,142 (a.k.a. 18000000 = 18 MHz)
- 20,833,333 (a.k.a 21000000 = 21 MHz)

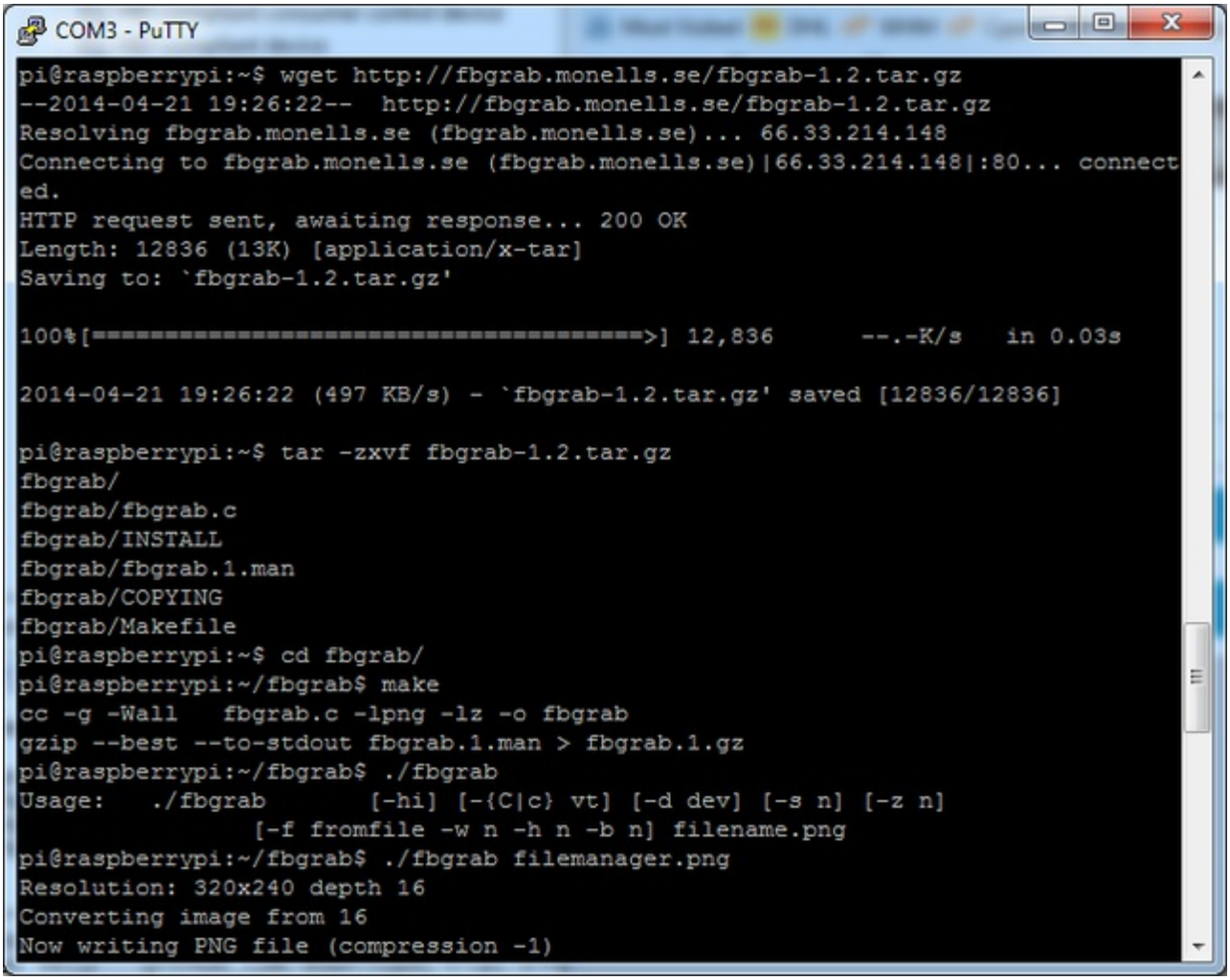
- 25,000,000 (= 25 MHz)
- 31,250,000 (a.k.a 32000000 = 32MHz)
- 41,666,666 (a.k.a 42000000 = 42MHz)
- 62,500,000 (a.k.a 62000000 = 62MHz)

So if you put in 48000000 for the speed, you won't actually get 48MHz, you'll actually only get about 42MHz because it gets rounded down. We tested this display nicely with 32MHz and we suggest that. But you can put in 42MHz or even try 62MHz and it will update faster

You can tweak fps (frames per second) from 20 to 60 and frequency up to 62MHz for tradeoffs in performance and speed. Reboot after each edit to make sure the settings are loaded properly. There's a trade off that if you ask for higher FPS you're going to load the kernel more because it's trying to keep the display updated.

How can I take screenshots of the little screen?
We took the screenshots for this tutorial with (http://adafru.it/diV)**fbgra (http://adafru.it/diV)b (http://adafru.it/diV)**

```
wget http://fbgrab.monells.se/fbgrab-1.2.tar.gz (http://adafru.it/diW)
tar -zxvf fbgrab*gz
cd fbgrab/
make
./fbgrab screenshot.png
```

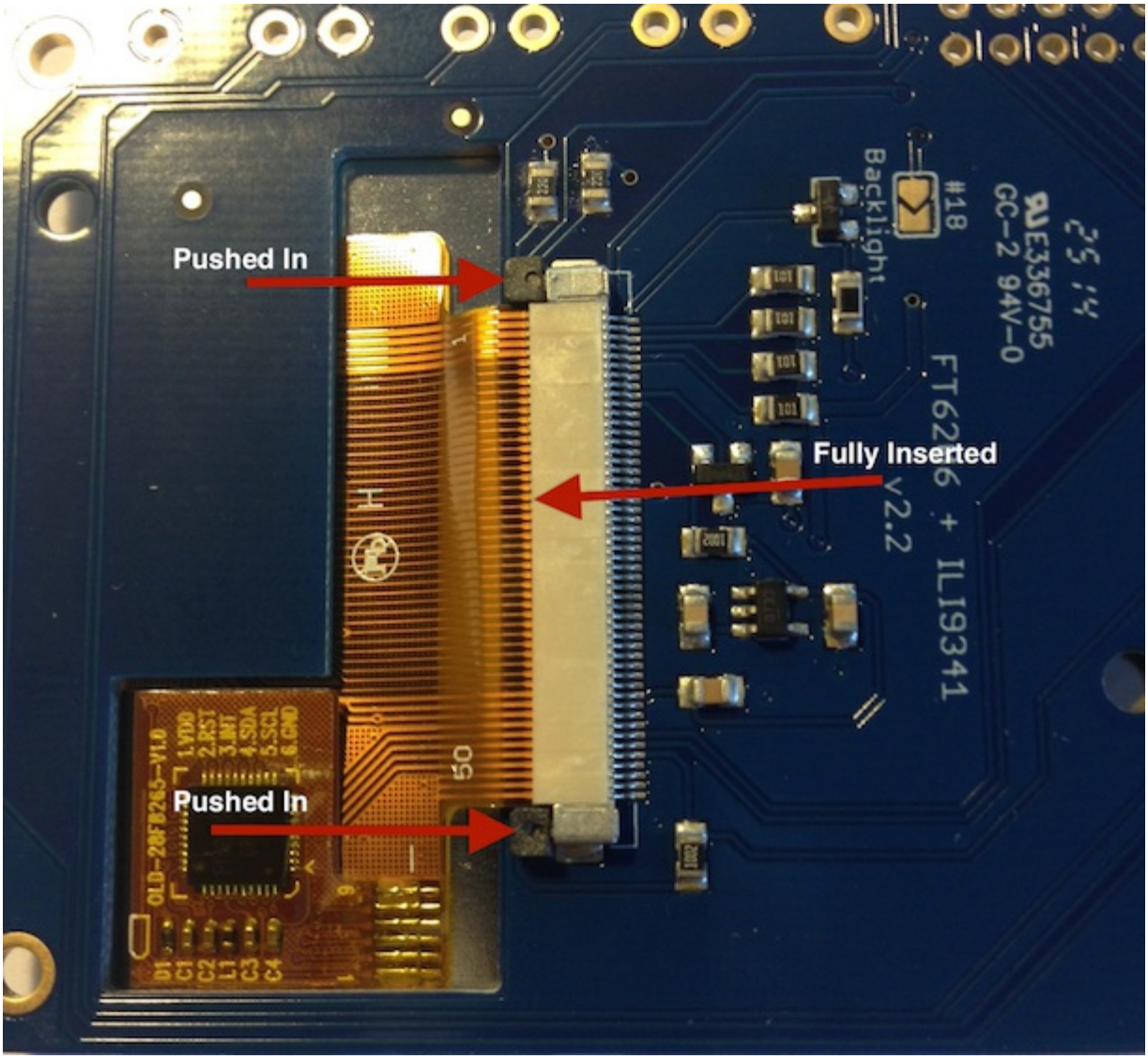


How do I automatically boot to X windows on the PiTFT?

Once you have a PiTFT installation setup you can add a custom X windows configuration to use the PiTFT by default. Then you can use the normal raspi-config boot to console/desktop options to control if the Pi boots to a console or desktop. See the detailed instructions on this page of the guide (http://adafru.it/ird) for more information.

My screen isn't working/works erratically/looks funny

Check to make syre that the flat flex cable is fully seated in the connetor and the 'ears' are pushed in to secure it. See the picture for what it should look like:



On my first run of startx I get a window saying "GDBus Error.org.Freedesktop Policy Kit1 Error: Failed Cannot determine user of subject"

This happens on the Raspberry Pi the first time you run startx, no matter what display. You can just re-start X and it wont appear again.

Can I get a right-click from the touch-screen?

Yes! Please see this post:

https://forums.adafruit.com/viewtopic.php?f=47&t=77528&p=393280#p393322 (http://adafru.it/fH3)

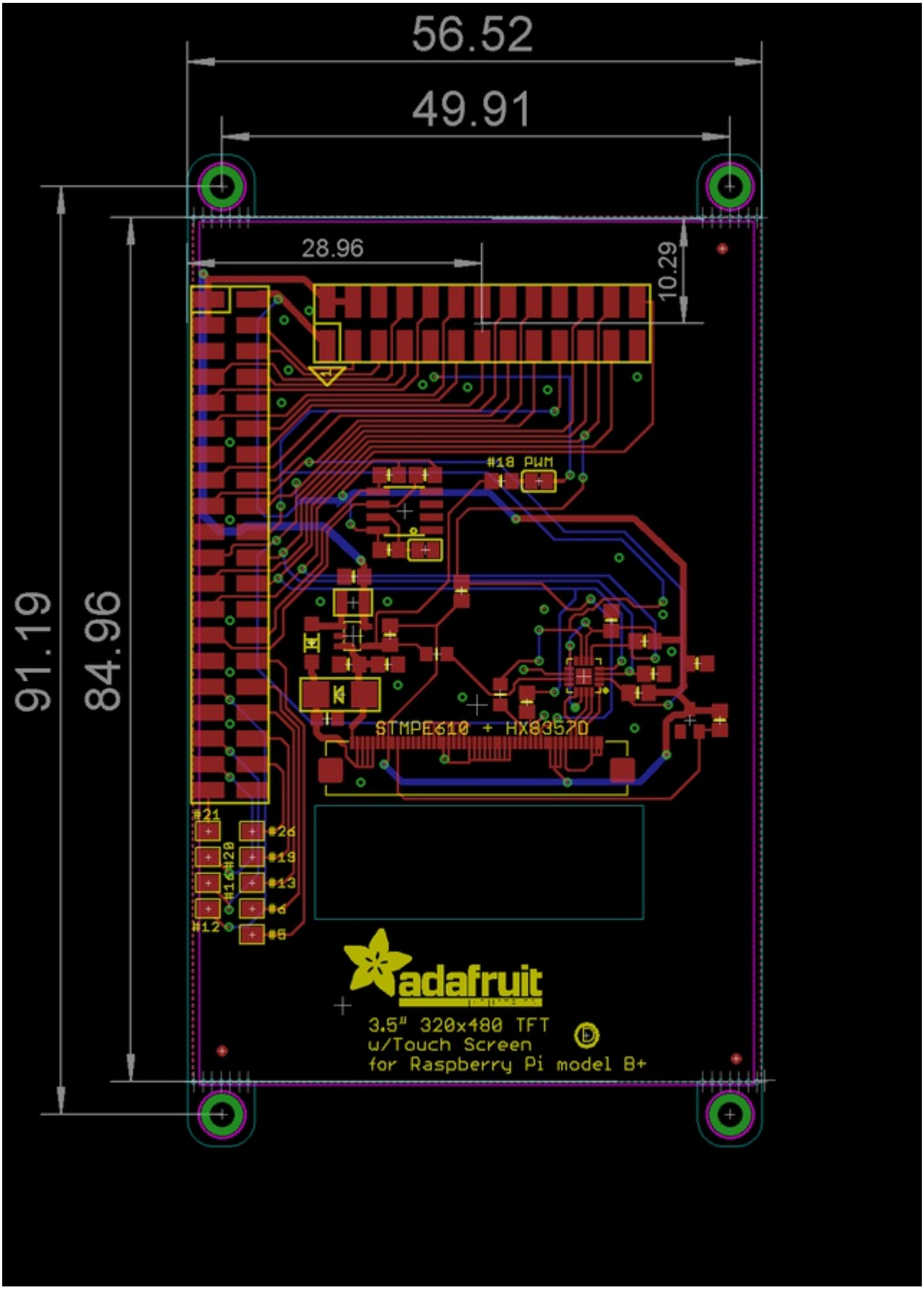
Downloads

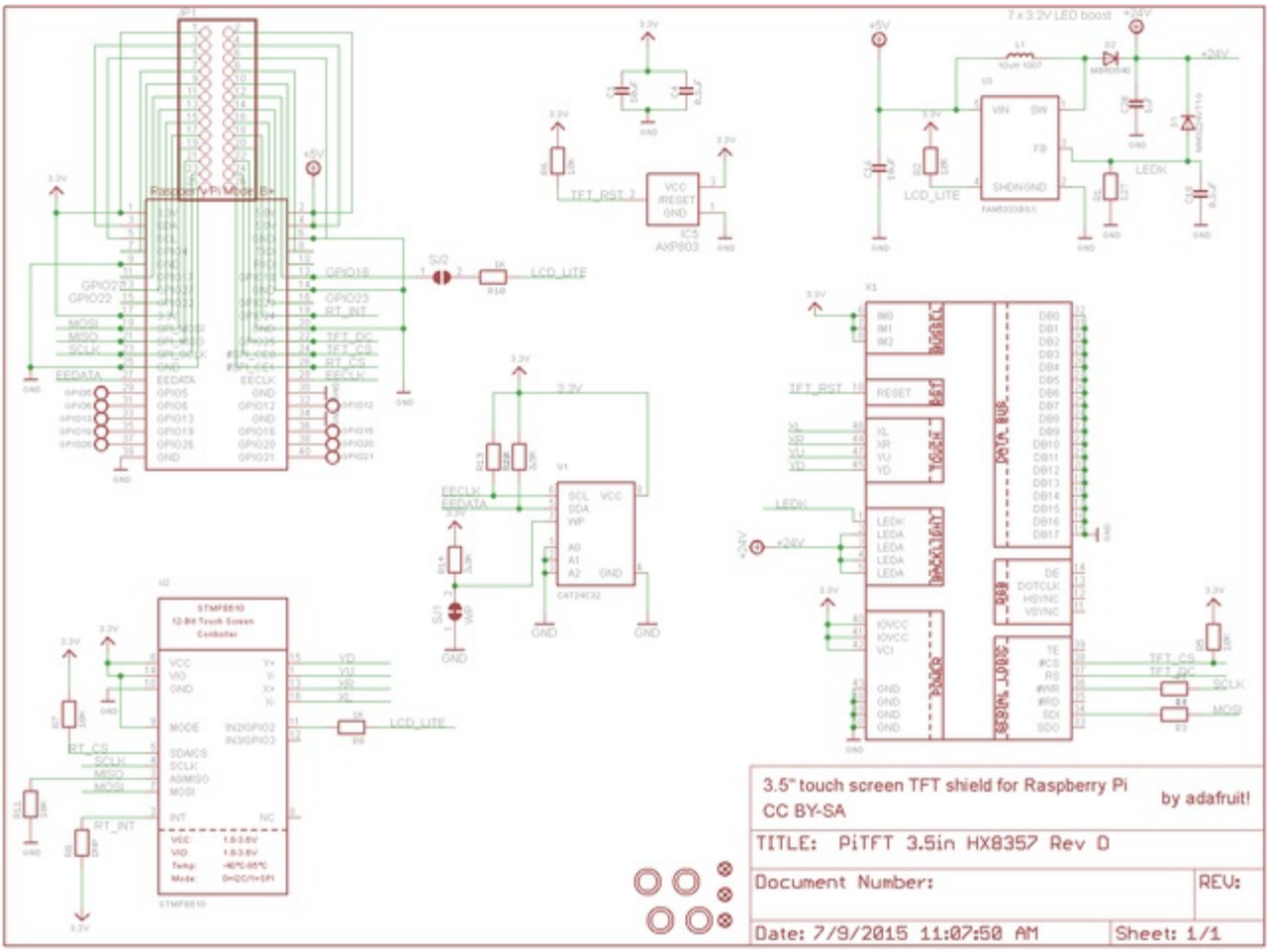
Files

- The latest kernel fork that adds all the TFT, touchscreen, and other addons is here on github (http://adafru.it/aPa)
- Datasheet for the controller chip (http://adafru.it/dQQ)
- Datasheet for the 'raw' 3.5" TFT display (http://adafru.it/dR4)
- PCB files on GitHub (http://adafru.it/rEC)
- Fritzing object in the Adafruit Fritzing library (http://adafru.it/aP3)

Layout and Schematic for PiTFT Plus 3.5"

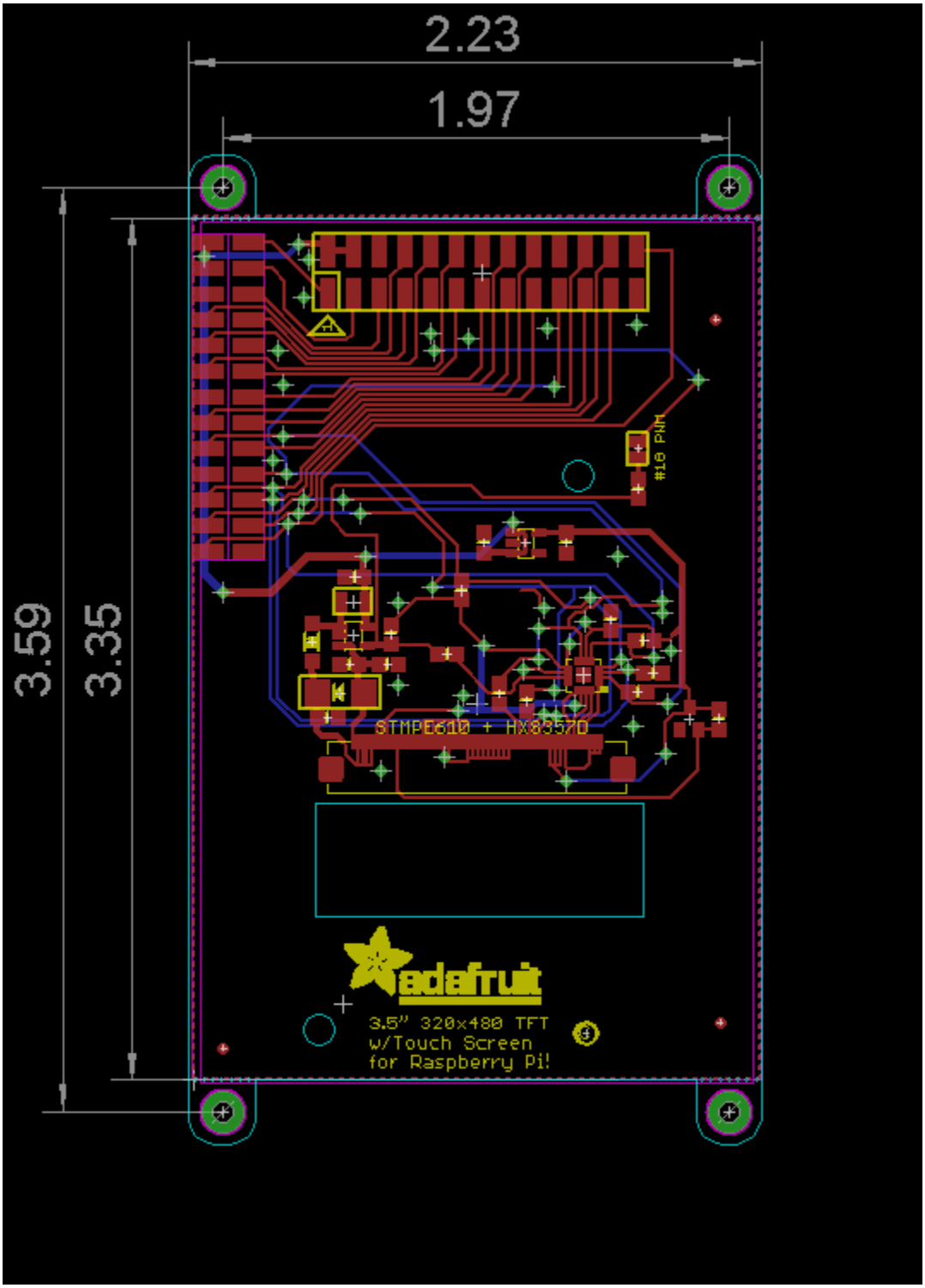
This is the newer PID 2441

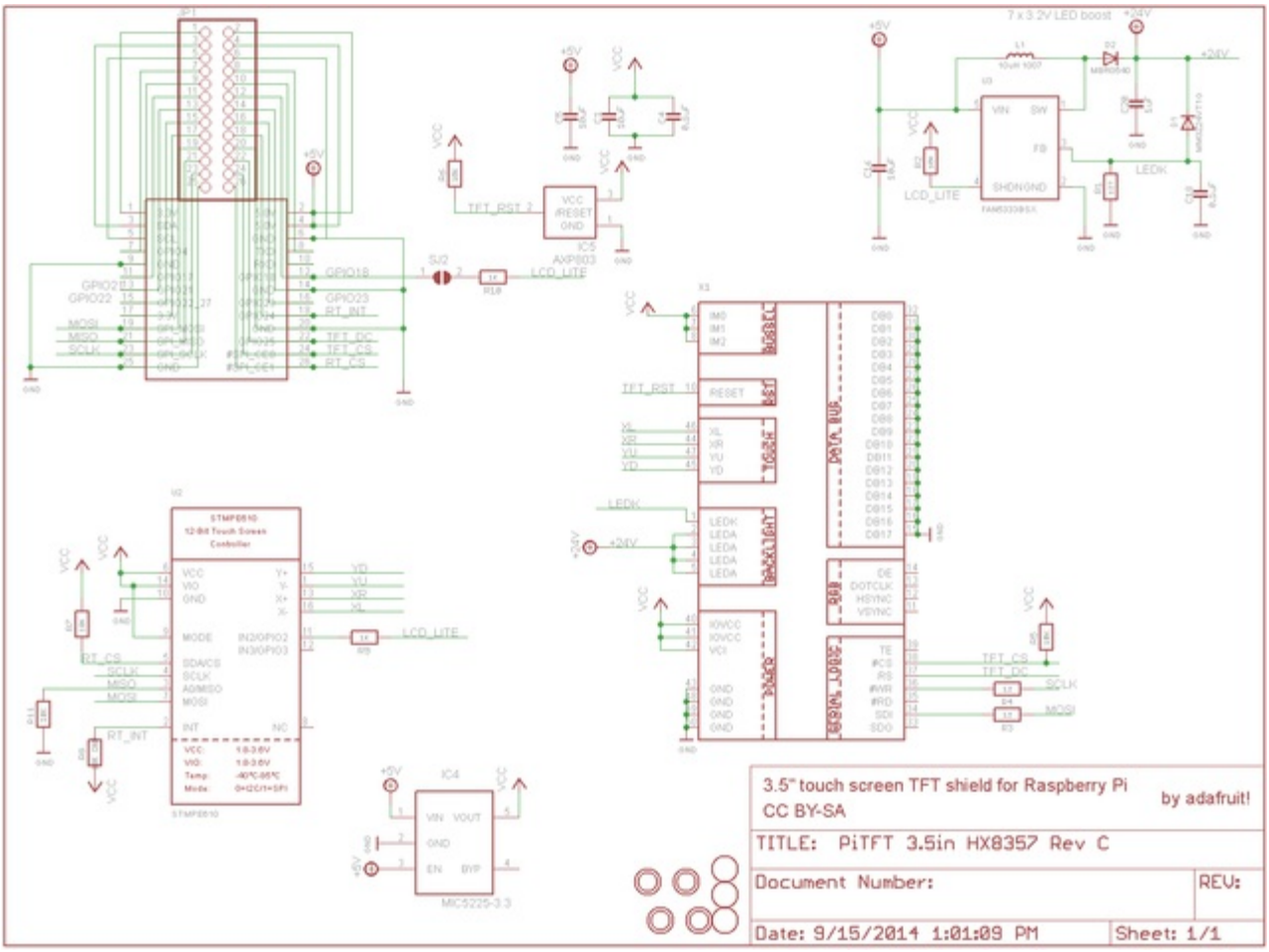

56.52
49.91
28.96
10.29
91.19
84.96
#18 PWM
STMPE610 + HX8357D
#21
#26
#19
#20
#16
#13
#6
#12
#5
adafruit
3.5" 320x480 TFT
w/Touch Screen
for Raspberry Pi model B+



Layout and Schematic for original PiTFT 3.5"

This is the original PID #2097 version


2.23
1.97
3.59
3.35
#16 PWM
STMPE610 + HX8357D
adafruit
3.5" 320x480 TFT
w/Touch Screen
for Raspberry Pi!


7 x 3.2V LED boost
3.5" touch screen TFT shield for Raspberry Pi
CC BY-SA
by adafruit!
TITLE: PiTFT 3.5in HX8357 Rev C
Document Number:
REV:
Date: 9/15/2014 1:01:09 PM
Sheet: 1/1