

□

Adafruit 2.2" PiTFT HAT - 320x240 Display

Created by lady ada



Last updated on 2016-08-01 08:10:32 PM UTC

Guide Contents

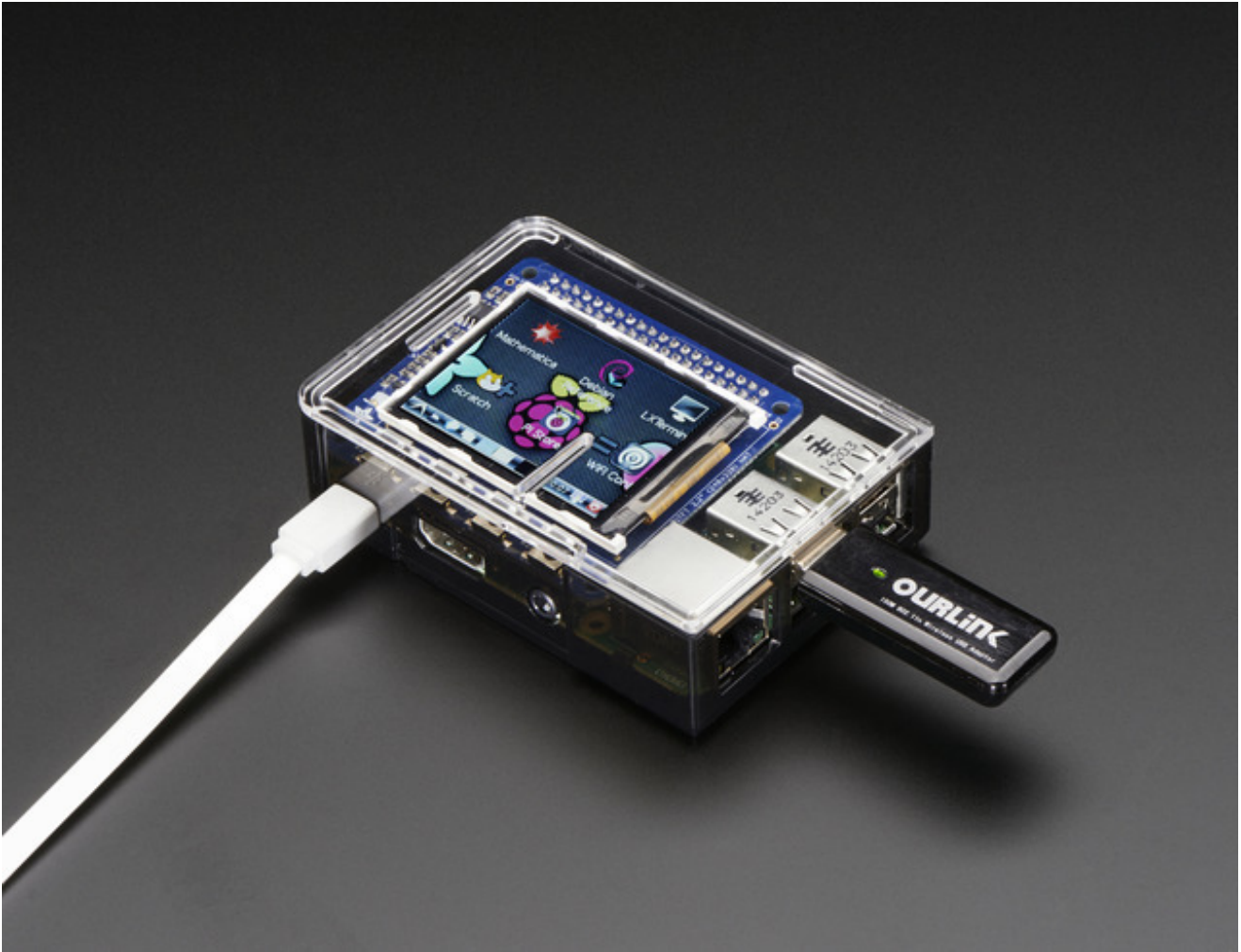
Guide Contents	2
Overview	3
Easy Install	7
Ready to go image	7
If you already have PiTFT / Notro Kernel	8
DIY Installer script	8
Detailed Installation	13
Before you start	14
Download & Install Kernel	14
Using the Console	21
Turn off Console Blanking	24
Playing Videos	25
Displaying Images	26
Using FBCP	27
Backlight Control	29
Extras!	30
Tactile switch as power button	30
Boot to X Windows on PiTFT	31
FAQ	33
Downloads	34
Schematic	34
Fabrication Print	34

Overview



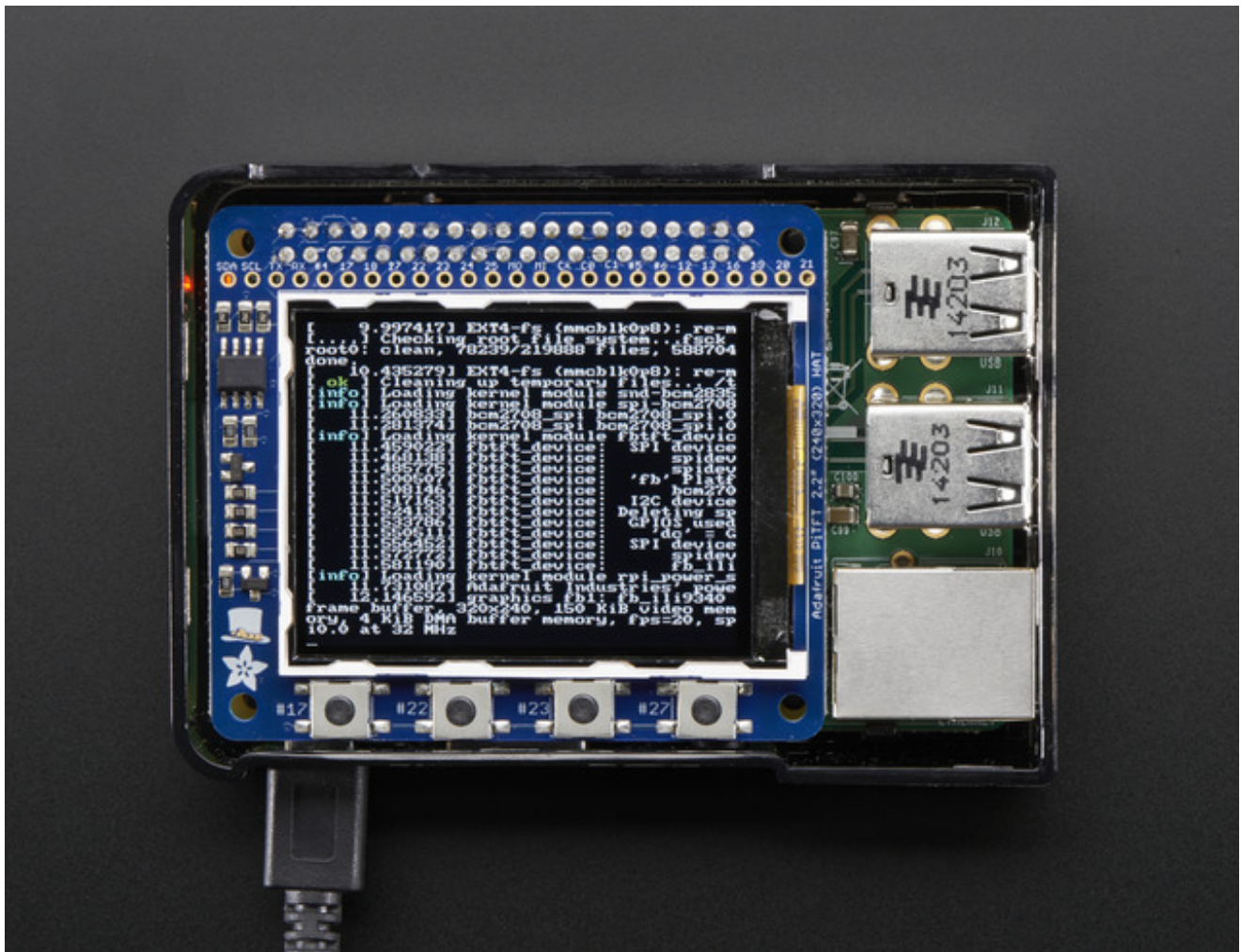
The cute PiTFT got even more adorable with this little primary display for Raspberry Pi in HAT form! It features a 2.2" display with 320x240 16-bit color pixels. The HAT uses the high speed SPI interface on the Pi and can use the mini display as a console, X window port, displaying images or video etc. Best of all it plugs right in on top of your Model A+ or B+ and fits into our case quite nicely.

It's designed to plug directly onto the Pi Model A+ or B+. While not specifically designed for Pi Model A or B, [you can use it with A/B if you solder in an extra-tall 2x13 header \(not included\)](http://adafru.it/eib) (<http://adafru.it/eib>) instead of the included 2x20 header

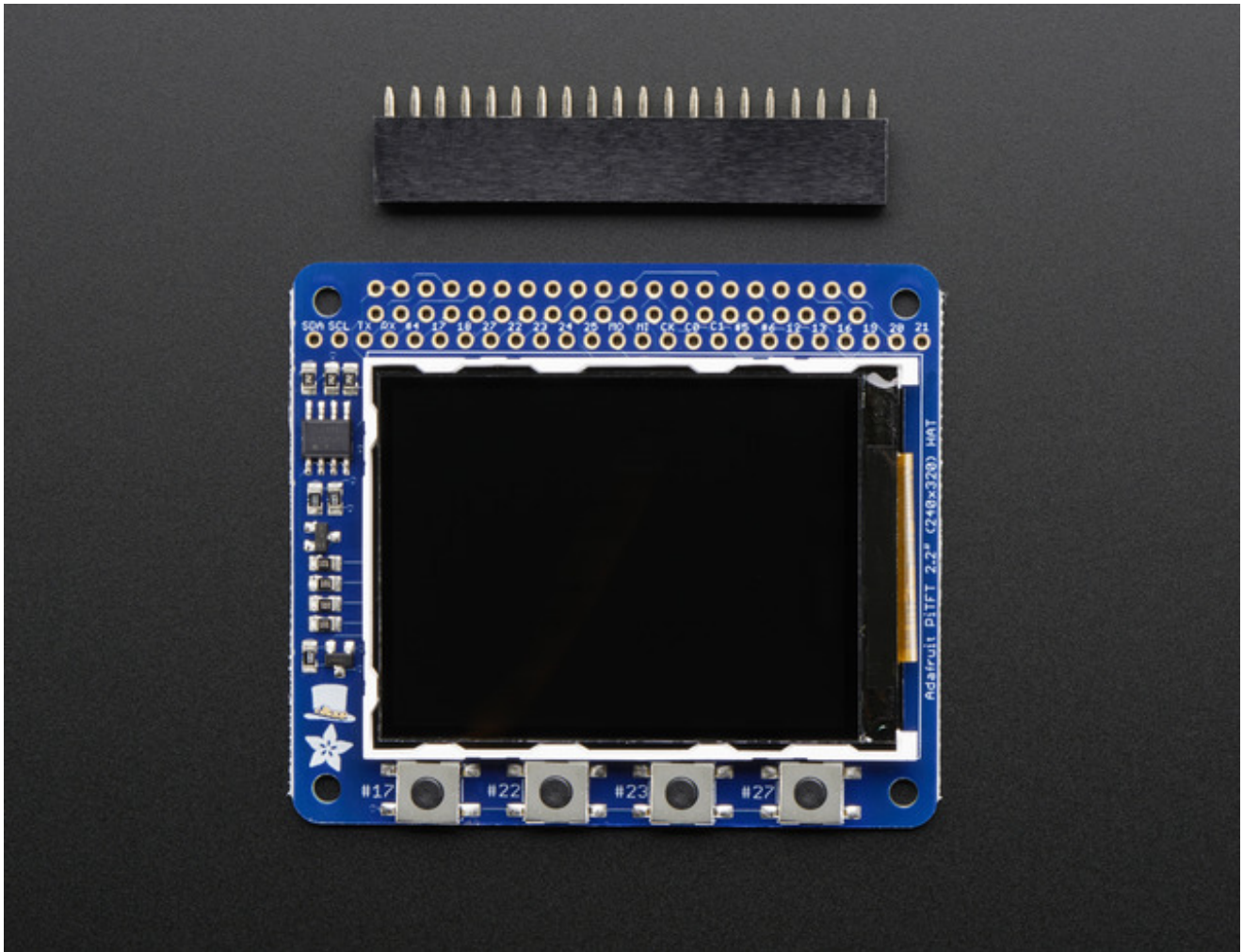


This design uses the hardware SPI pins (SCK, MOSI, MISO, CE0, CE1) as well as GPIO #25. All other GPIO are unused. Since we had a tiny bit of space, there's 4 flat tactile switches wired to four GPIOs, that you can use if you want to make a basic user interface. For example, you can use one as a power on/off button.

All the other pins are available on a 25-pin long breakout line.



To make it super easy for use: we've created a custom kernel package based off of Notro's awesome framebuffer work, so you can install it over your existing Raspbian (or derivative) images in just a few commands.



Comes as a fully assembled display PCB and an additional 2x20 GPIO header. Some light soldering is required to attach the 2x20 GPIO header to the HAT but it's fast and easy for anyone with a soldering iron and solder

This tutorial series shows you how to install the software, play small videos, or display images such as from your PiCam and more!

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

If 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 \(http://adafru.it/aMW\)](http://adafru.it/aMW)

[Download Jessie-based PiTFT 2.2" HAT Image for Pi 1, Pi 2 and Pi 3 \(March 25, 2016\) http://adafru.it/mAe](http://adafru.it/mAe)

[Download Jessie Lite-based PiTFT 2.2" HAT Image for Pi 1, Pi 2 and Pi 3 \(March 25, 2016\) http://adafru.it/mAf](http://adafru.it/mAf)

Previous Images:

- [Raspbian Jessie 2015/09/24-based image \(http://adafru.it/iDC\)](http://adafru.it/iDC)
- [Raspbian Wheezy 2015/09/09-based image \(http://adafru.it/idt\)](http://adafru.it/idt)

If you already have PiTFT / Notro Kernel

The 2.2" PiTFT is supported in all PiTFT images/kernels as well as Notro's images/kernels, use this in your **/etc/modprobe.d/adafruit.conf** file

```
options fbttft_device name=adafruit22a gpios=dc:25 rotate=270 frequency=32000000
```

Or if you have a device tree overlay-supporting kernel, put the following at the end of **/boot/config.txt**

```
[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=pitft22,rotate=270,speed=32000000,fps=20
```

The DIY Installer isn't working right now, please try the All-In-One image above - no ETA on why its not working, something to do with the latest Raspbian has changed. Thanks!

DIY Installer script

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


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

works

To download and run it, simply run the following commands:

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

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



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

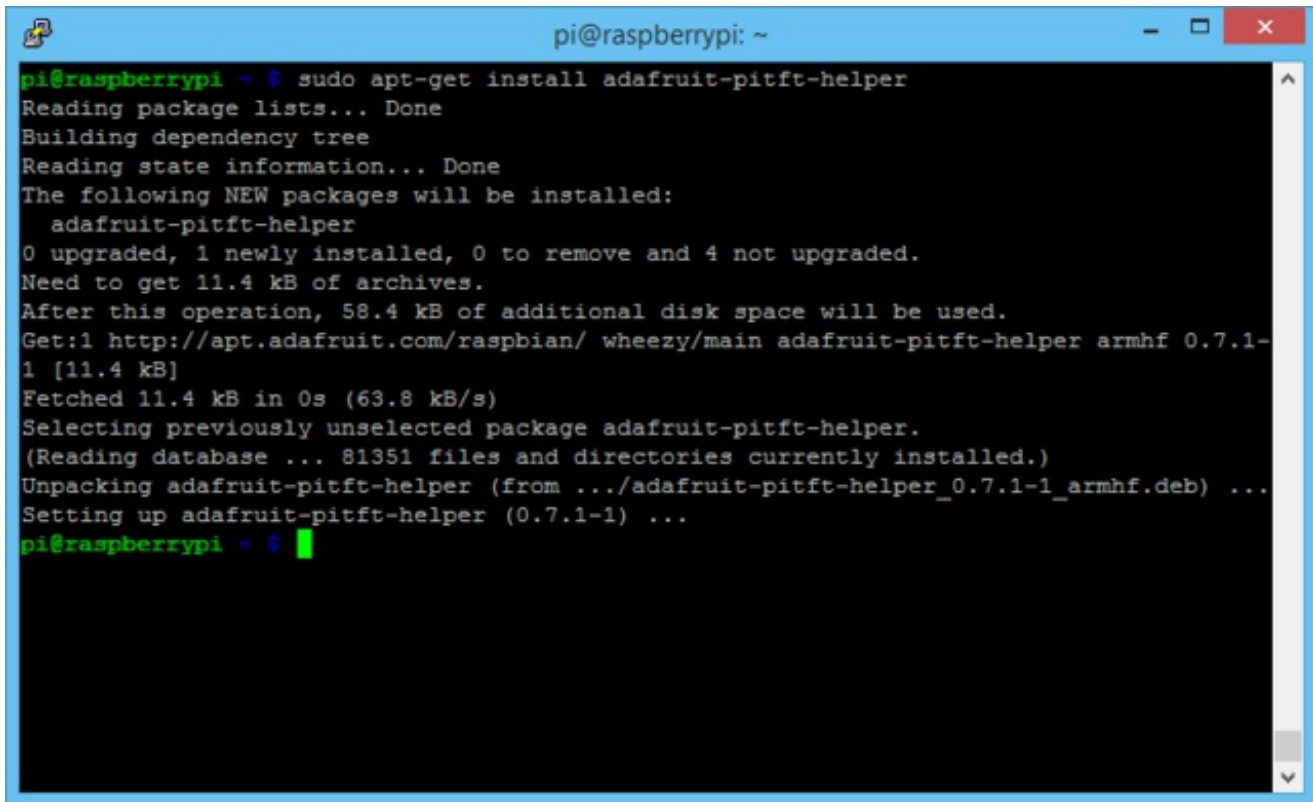
The next two do the actual download and installation, it'll take a while because there's a lot of software to replace for PiTFT support.

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

When prompted to continue, say **Yes**.

```
pi@raspberrypi ~/Adafruit-Occidentalis $ sudo apt-get install raspberrypi-bootloader
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
The following packages will be upgraded:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0 raspberrypi-bootloader
5 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
Need to get 61.5 MB of archives.
After this operation, 12.7 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

It's normal for the Pi to pause at this step for up to 20 minutes, there's a lot of kernel software to replace

A terminal window titled 'pi@raspberrypi: ~' with standard window controls. The terminal output shows the command 'sudo apt-get install adafruit-pitft-helper' being executed. The output includes package list reading, dependency tree building, and state information reading. It lists 'adafruit-pitft-helper' as a new package to be installed, showing disk space requirements (11.4 kB) and the source (http://apt.adafruit.com/raspbian/wheezy/main). The package is then fetched and installed, with the terminal showing the unpacking and setup process. The prompt returns to 'pi@raspberrypi ~\$' with a green cursor.

```
pi@raspberrypi ~$ sudo apt-get install adafruit-pitft-helper
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  adafruit-pitft-helper
0 upgraded, 1 newly installed, 0 to remove and 4 not upgraded.
Need to get 11.4 kB of archives.
After this operation, 58.4 kB of additional disk space will be used.
Get:1 http://apt.adafruit.com/raspbian/wheezy/main adafruit-pitft-helper armhf 0.7.1-1 [11.4 kB]
Fetched 11.4 kB in 0s (63.8 kB/s)
Selecting previously unselected package adafruit-pitft-helper.
(Reading database ... 81351 files and directories currently installed.)
Unpacking adafruit-pitft-helper (from .../adafruit-pitft-helper_0.7.1-1_armhf.deb) ...
Setting up adafruit-pitft-helper (0.7.1-1) ...
pi@raspberrypi ~$
```

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 22
```

This will install the 2.2" 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!

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo adafruit-pitft-helper -t 28r  
Type = 28r  
[PITFT] Updating X11 default calibration...  
[PITFT] Updating X11 setup tweaks...  
Moving /usr/share/X11/xorg.conf.d/99-fbturbo.conf to /home/pi/  
Adding 'export FRAMEBUFFER=/dev/fb1'  
[PITFT] Updating tslib default calibration...  
[PITFT] Updating SysFS rules for Touchscreen...  
Would you like the console to appear on the PiTFT display? [y/n] y
```

You will also be prompted on whether you want one of the tactile buttons to act as an 'on off' switch. Answer Y or N depending on your personal desires!

```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo adafruit-pitft-helper -t 28r  
Type = 28r  
[PITFT] Updating X11 default calibration...  
[PITFT] Updating X11 setup tweaks...  
Moving /usr/share/X11/xorg.conf.d/99-fbturbo.conf to /home/pi/  
Adding 'export FRAMEBUFFER=/dev/fb1'  
[PITFT] Updating tslib default calibration...  
[PITFT] Updating SysFS rules for Touchscreen...  
Would you like the console to appear on the PiTFT display? [y/n] y  
[PITFT] Updating console to PiTFT...  
[PITFT] Updating /etc/modules...  
Adding stmpe_ts  
Would you like GPIO #23 to act as a on/off button? [y/n] n
```

Thats it!

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

Detailed Installation

The DIY Installer isn't working right now, please try the All-In-One image above - no ETA on why its not working, something to do with the latest Raspbian has changed. Thanks! 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 2.2" TFT , 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 won't 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 patchfile \(http://adafru.it/cY2\)](http://adafru.it/cY2), 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 \(http://adafru.it/aWq\)](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.

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
```

This will take a up to 20 minutes so go make a sandwich or coffee. It takes longer than it used to because there's now 2 kernels (v6 and v7 arm) and 2 kernel module directories.

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

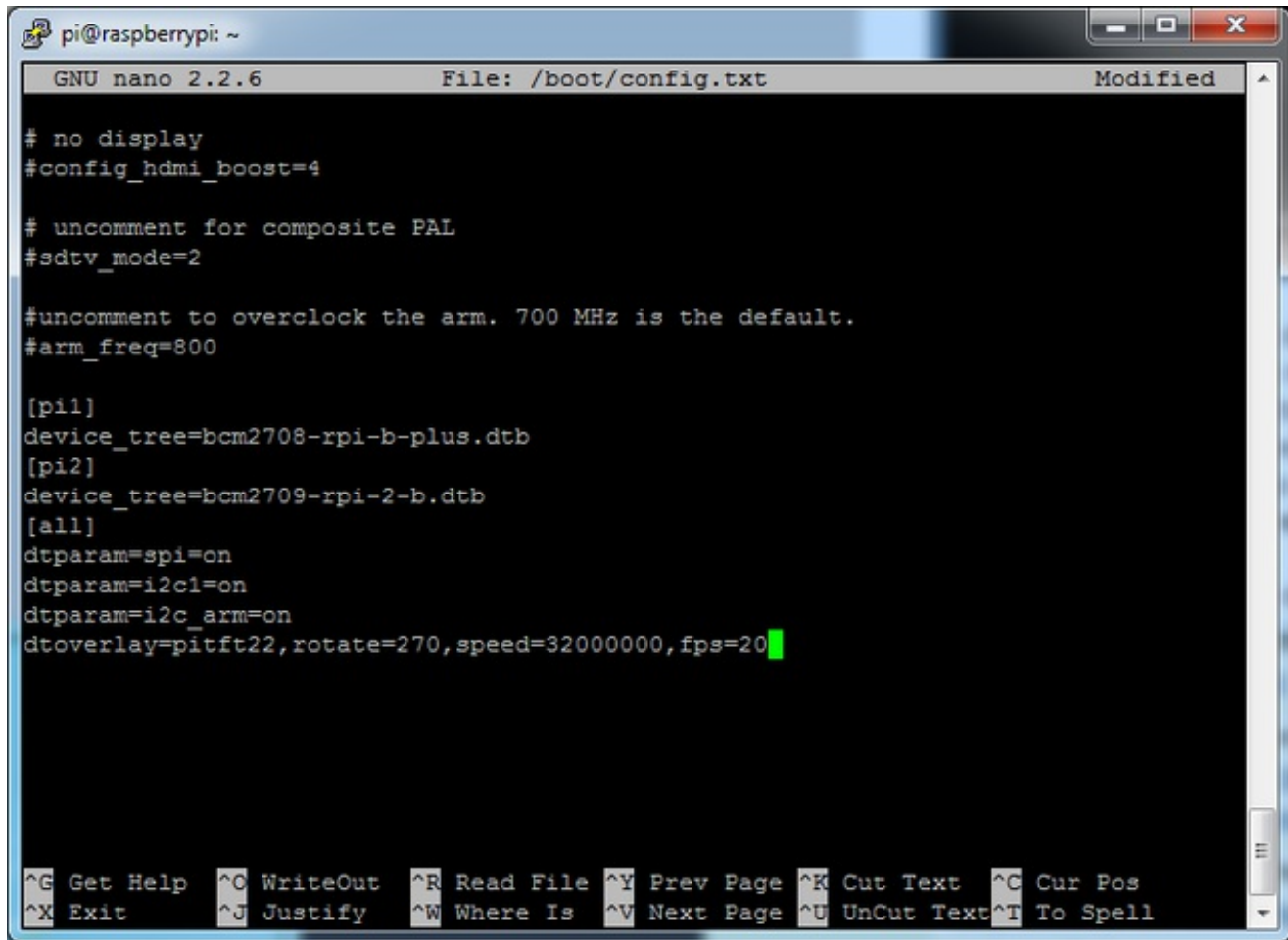
```
pi@raspberrypi ~/Adafruit-Occidentalis $ sudo apt-get install raspberrypi-bootloader
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
The following packages will be upgraded:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0 raspberrypi-bootloader
5 upgraded, 0 newly installed, 0 to remove and 32 not upgraded.
Need to get 61.5 MB of archives.
After this operation, 12.7 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

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=pitft22,rotate=270,speed=32000000,fps=20
```



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: /boot/config.txt Modified  
  
# no display  
#config_hdmi_boost=4  
  
# uncomment for composite PAL  
#sdtv_mode=2  
  
#uncomment to overclock the arm. 700 MHz is the default.  
#arm_freq=800  
  
[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=pitft22,rotate=270,speed=32000000,fps=20  
  
^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
```

The **rotate=** variable tells the driver to rotate the screen **0 90 180** or **270** degrees.

0 is portrait, with the bottom near the "Adafruit Logo"

90 is landscape, with the bottom of the screen near the buttons.

180 is portrait, with the top near the "Adafruit Logo"

270 is landscape, with the top of the screen near the buttons.

You can change this file with **nano** and reboot to make the change stick.

The **speed=** variable tells the driver how fast to drive the display. 32MHz (**32000000**) is a pretty nice 20 FPS rate but if your screen is acting funny, try taking it down to 16MHz (**16000000**)

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

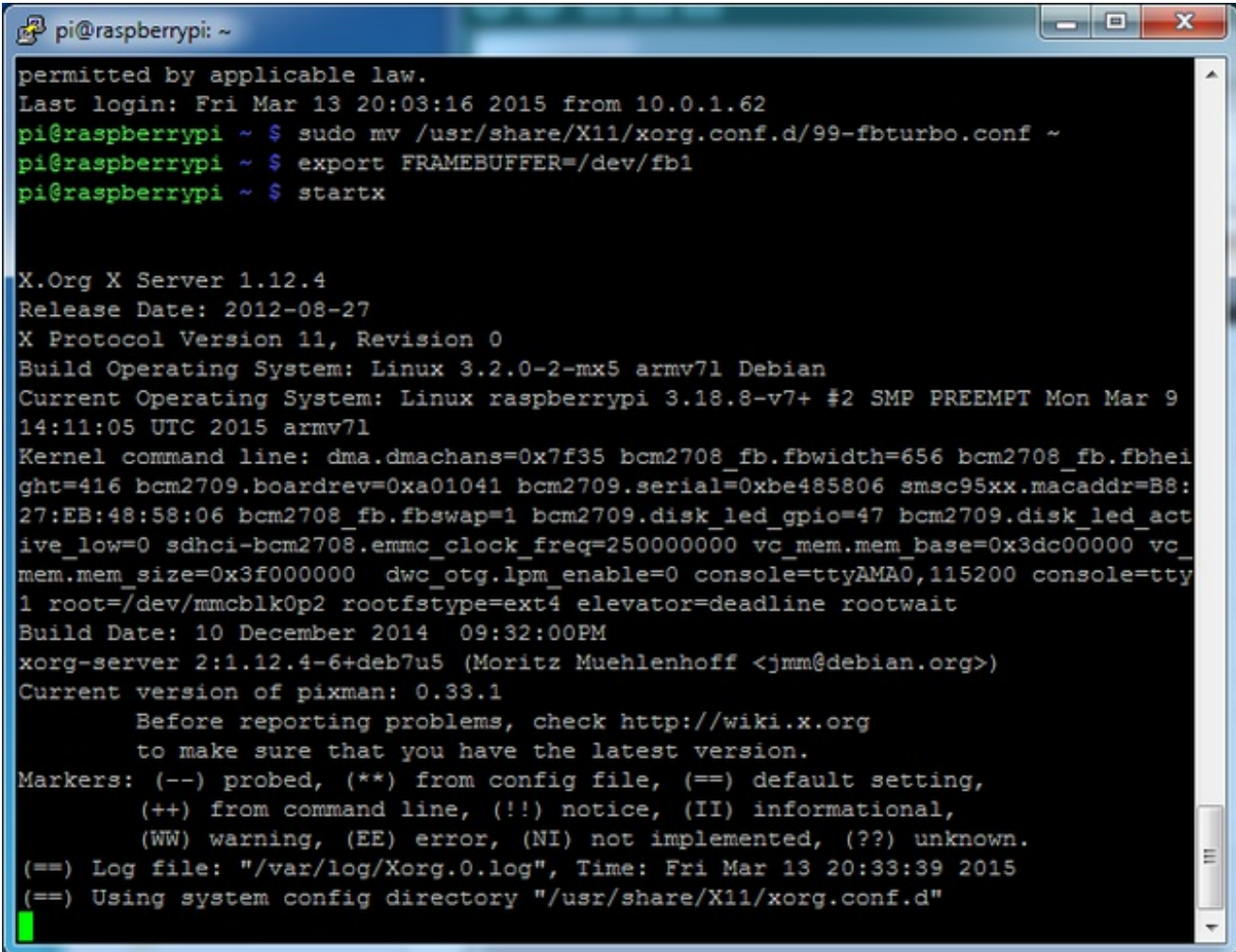
sudo reboot (if you have the TFT plate installed already)

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
```



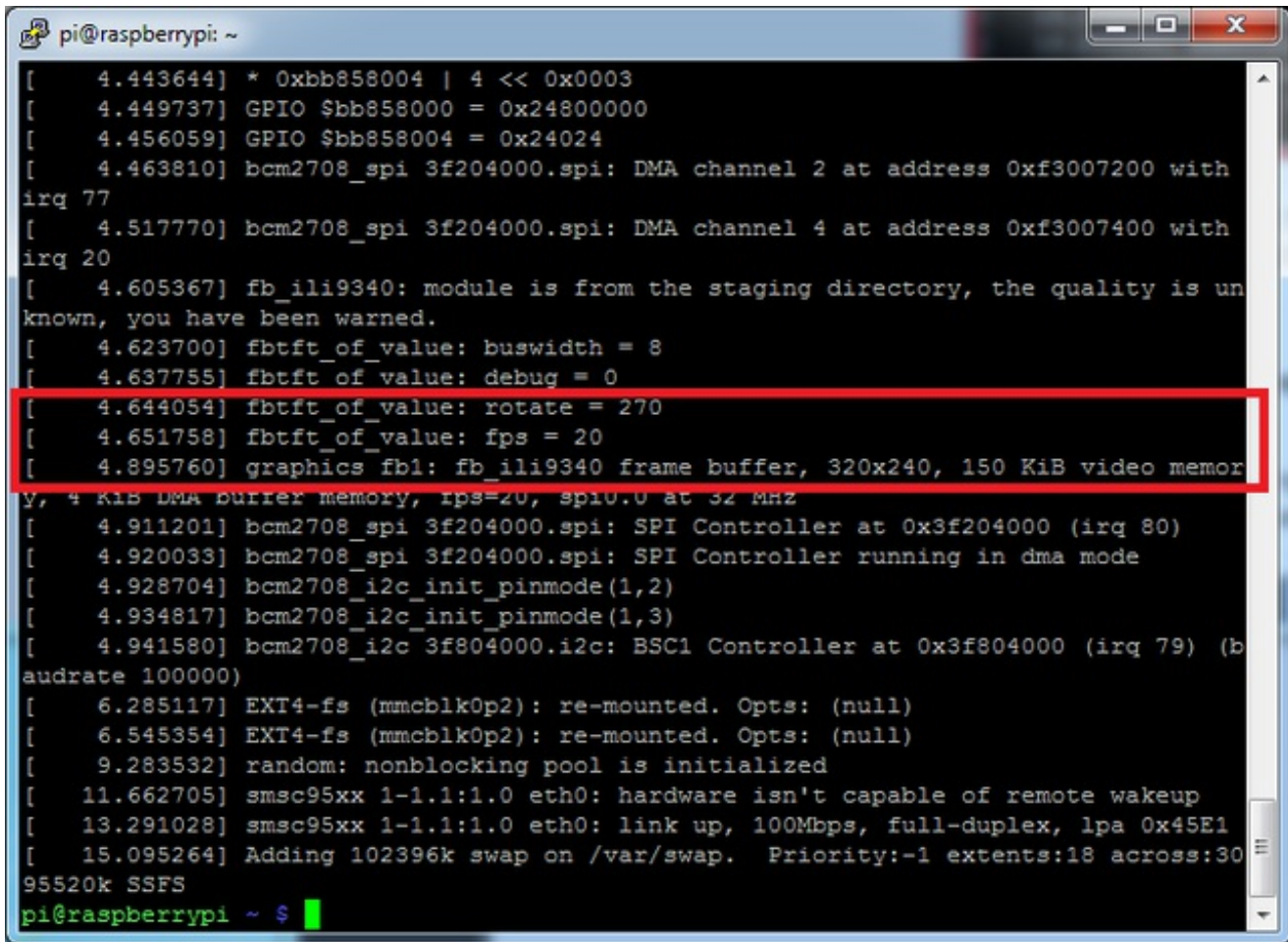
```
pi@raspberrypi: ~
permitted by applicable law.
Last login: Fri Mar 13 20:03:16 2015 from 10.0.1.62
pi@raspberrypi ~ $ sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~
pi@raspberrypi ~ $ export FRAMEBUFFER=/dev/fb1
pi@raspberrypi ~ $ startx

X.Org X Server 1.12.4
Release Date: 2012-08-27
X Protocol Version 11, Revision 0
Build Operating System: Linux 3.2.0-2-mx5 armv7l Debian
Current Operating System: Linux raspberrypi 3.18.8-v7+ #2 SMP PREEMPT Mon Mar 9
14:11:05 UTC 2015 armv7l
Kernel command line: dma.dmachans=0x7f35 bcm2708_fb.fbwidth=656 bcm2708_fb.fbhei
ght=416 bcm2709.boardrev=0xa01041 bcm2709.serial=0xbe485806 smsc95xx.macaddr=B8:
27:EB:48:58:06 bcm2708_fb.fbswap=1 bcm2709.disk_led_gpio=47 bcm2709.disk_led_act
ive_low=0 sdhci-bcm2708.emmc_clock_freq=250000000 vc_mem.mem_base=0x3dc00000 vc_
mem.mem_size=0x3f000000 dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty
1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
Build Date: 10 December 2014 09:32:00PM
xorg-server 2:1.12.4-6+deb7u5 (Moritz Muehlenhoff <jmm@debian.org>)
Current version of pixman: 0.33.1
    Before reporting problems, check http://wiki.x.org
    to make sure that you have the latest version.
Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.0.log", Time: Fri Mar 13 20:33:39 2015
(==) Using system config directory "/usr/share/X11/xorg.conf.d"
```

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

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 ILI9340 screen

frequency as highlighted here



```
pi@raspberrypi: ~  
[ 4.443644] * 0xbb858004 | 4 << 0x0003  
[ 4.449737] GPIO $bb858000 = 0x24800000  
[ 4.456059] GPIO $bb858004 = 0x24024  
[ 4.463810] bcm2708_spi 3f204000.spi: DMA channel 2 at address 0xf3007200 with  
irq 77  
[ 4.517770] bcm2708_spi 3f204000.spi: DMA channel 4 at address 0xf3007400 with  
irq 20  
[ 4.605367] fb_ili9340: module is from the staging directory, the quality is un  
known, you have been warned.  
[ 4.623700] fbtft_of_value: buswidth = 8  
[ 4.637755] fbtft_of_value: debug = 0  
[ 4.644054] fbtft_of_value: rotate = 270  
[ 4.651758] fbtft_of_value: fps = 20  
[ 4.895760] graphics fb1: fb_ili9340 frame buffer, 320x240, 150 KiB video memor  
y, 4 KiB DMA buffer memory, fps=20, spi0.0 at 32 MHz  
[ 4.911201] bcm2708_spi 3f204000.spi: SPI Controller at 0x3f204000 (irq 80)  
[ 4.920033] bcm2708_spi 3f204000.spi: SPI Controller running in dma mode  
[ 4.928704] bcm2708_i2c_init_pinmode(1,2)  
[ 4.934817] bcm2708_i2c_init_pinmode(1,3)  
[ 4.941580] bcm2708_i2c 3f804000.i2c: BSC1 Controller at 0x3f804000 (irq 79) (b  
audrate 100000)  
[ 6.285117] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)  
[ 6.545354] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)  
[ 9.283532] random: nonblocking pool is initialized  
[ 11.662705] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup  
[ 13.291028] smsc95xx 1-1.1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0x45E1  
[ 15.095264] Adding 102396k swap on /var/swap. Priority:-1 extents:18 across:30  
95520k SSFS  
pi@raspberrypi ~ $
```

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
```

The image shows a Raspberry Pi Zero W board with a custom blue PCB and a black display. The display shows a terminal window with boot logs. The board is connected to a USB drive and a USB cable.

Boot logs on the display:

```

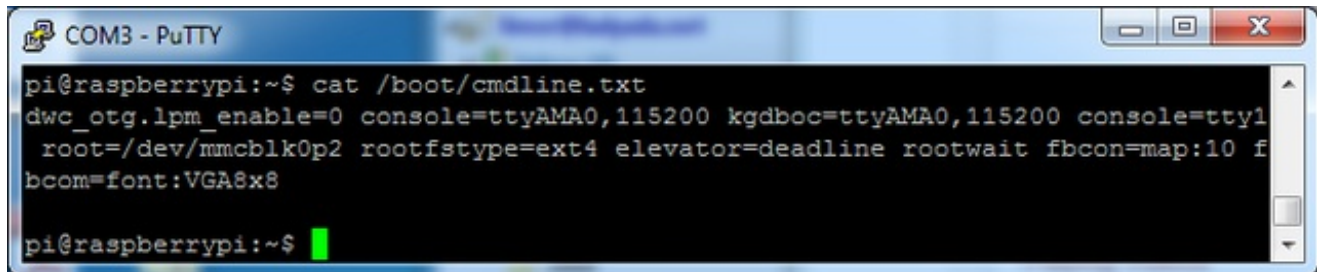
9.2974171 EK74-Fs (mchblk0p8): re-m
[...]: Checking root file system...fsck
root0: clean, 78239/219888 files, 588704
done.
10.4352791 EK74-Fs (mchblk0p8): re-m
ok: Cleaning up temporary files.../t
info: Loading kernel module snd-bcm2835
info: Loading kernel module spi-bcm2708
11.2608331 bcm2708_spi bcm2708 spi.0
11.2813741 bcm2708_spi bcm2708 spi.0
info: Loading kernel module fbtt device
11.4510221 fbtt device: SPI device
11.4601381 fbtt device: spidev
11.4857751 fbtt device: spidev
11.5005071 fbtt device: 'fb' Platf
11.5081461 fbtt device: bcm270
11.5171631 fbtt device: I2C device
11.5241821 fbtt device: Deleting sp
11.5295111 fbtt device: GP I/Os used
11.5354521 fbtt device: dc' = C
11.5511961 fbtt device: SPI device
11.5511961 fbtt device: spidev
11.5511961 fbtt device: fb. 111
info: Loading kernel module rpi_power_s
12.3108771 adafruit industries' powe
graphics fb1: fb 1119940
frame buffer, 320x240, 150 Kib video mem
oru, 4 Kib DMA buffer memory, fps=20, sp
10.0 at 32 MHz
  
```

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.

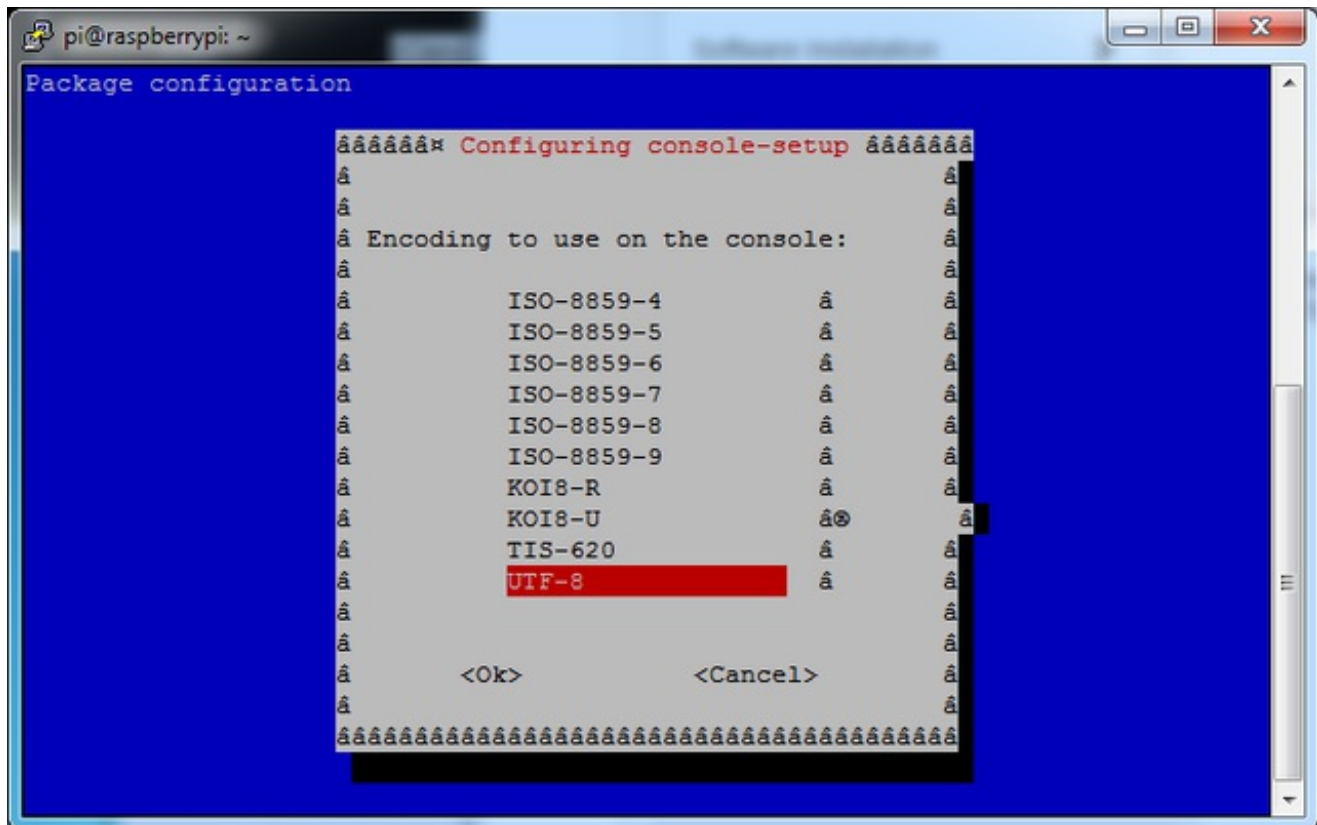
On the next boot, it will bring up the console.

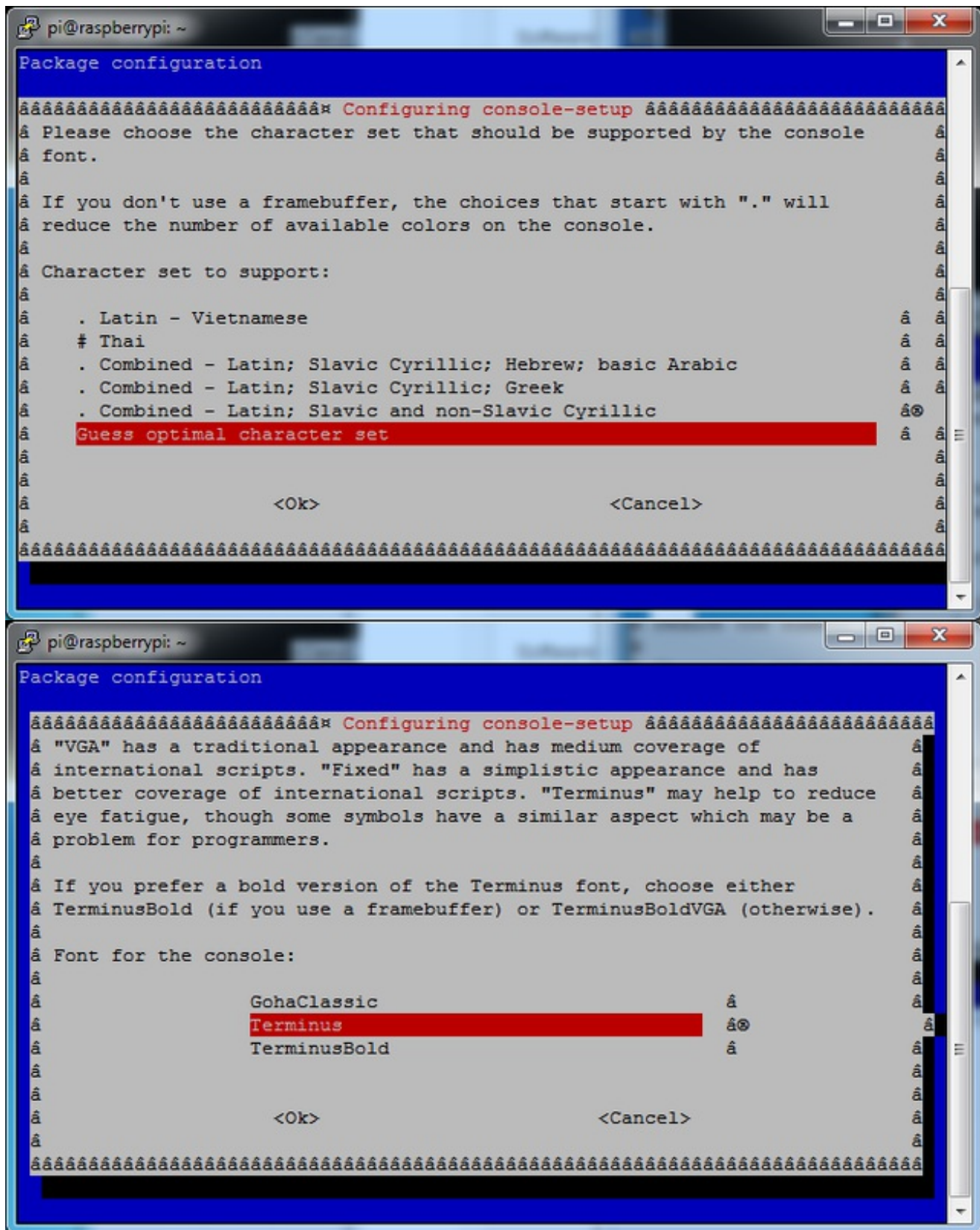
Note that the kernel has to load up the display driver module before it can display anything on it so you won't get the rainbow screen, a NooBs prompt, or a big chunk of the kernel details since the module is loaded fairly late in the boot process.

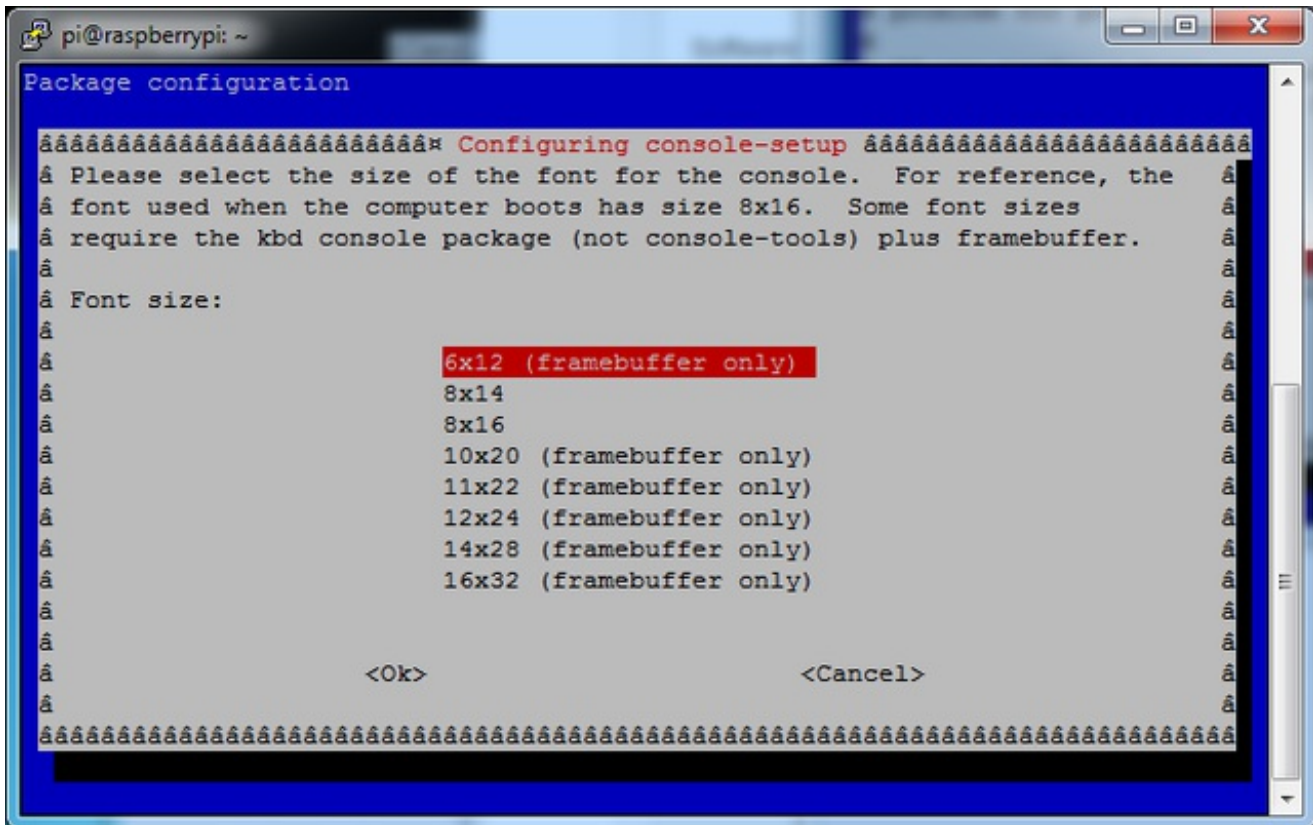


```
pi@raspberrypi:~$ cat /boot/cmdline.txt
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait fbcon=map:10 f
bcom=font:VGA8x8
pi@raspberrypi:~$
```

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







Turn off Console Blanking

You may notice the console goes black after 30 minutes, this is a sort of 'power saving' or 'screensaver' feature. You can disable this by editing `/etc/kbd/config` and setting the blank time to 0 (which turns it off)

```
BLANK_TIME=30
```



Playing Videos

[This tutorial page is the same as the one for the 2.8" PiTFT so please go read all about it there! \(http://adafru.it/dIY\)](http://adafru.it/dIY)



Displaying Images

[This tutorial page is the same as the one for the 2.8" PiTFT so please go read all about it there! \(http://adafru.it/dJ0\)](http://adafru.it/dJ0)

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://adafruit.com/blog/2015/08/28/fbcp-framebuffer-copy/)

<http://adafruit.com/blog/2015/08/28/fbcp-framebuffer-copy/>

Backlight Control

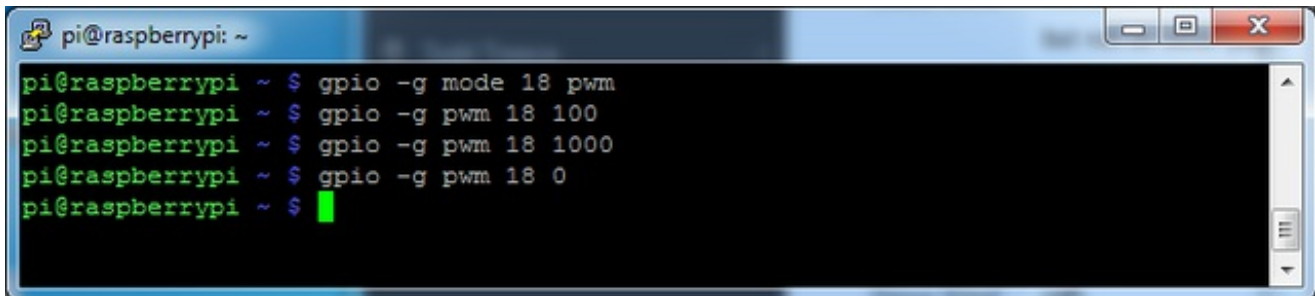
Unlike the resistive 2.8" PiTFT, this little PiTFT does not have a resistive touch controller chip that we can take advantage of as an extra backlight control pin. Instead, you can set use GPIO #18 as an on/off or PWM control.

Note that if you are playing audio out the headphone jack, you can't use the PWM capabilities of GPIO #18 at the same time, the PWM function is reassigned to do audio. However, you can use it as a simple on/off pin

There's python code available for controlling the PWM on #18 but you can also just use the kernel module and shell commands.

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

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

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The terminal shows four lines of commands being executed: 'gpio -g mode 18 pwm', 'gpio -g pwm 18 100', 'gpio -g pwm 18 1000', and 'gpio -g pwm 18 0'. Each line is preceded by the prompt 'pi@raspberrypi ~ \$'. The cursor is at the end of the last line.

```
pi@raspberrypi ~ $ gpio -g mode 18 pwm
pi@raspberrypi ~ $ gpio -g pwm 18 100
pi@raspberrypi ~ $ gpio -g pwm 18 1000
pi@raspberrypi ~ $ gpio -g pwm 18 0
pi@raspberrypi ~ $
```

Try other numbers, from 0 (off) to 1023 (all the way on)!

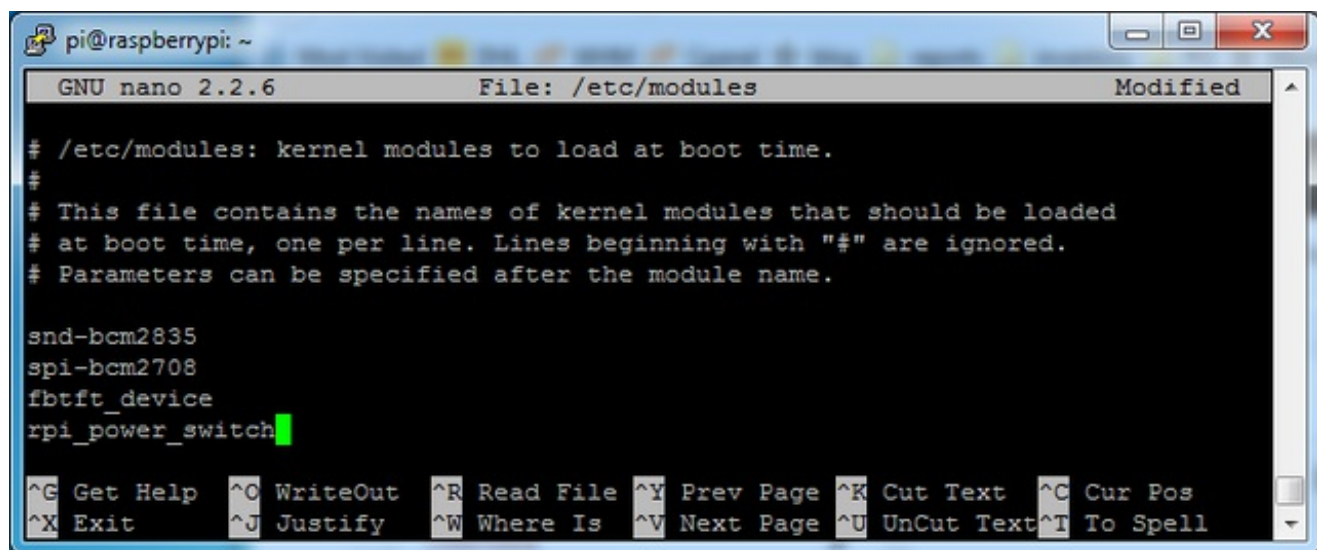
Extras!

Tactile switch as power button

It's a good idea to safely turn off your Pi with a good **sudo shutdown -h now** but that often means pulling out a keyboard or connecting to the console. With our kernel we added a cool module that will let you turn any GPIO into a power button. Since there's a couple of tactile switches right there on the front, let's turn one into a power button. Press once to properly turn off the pi, press again to start it up. Isn't that nice?

We'll be using GPIO #23, the third button. You can use any of them or other GPIO but #23's our favorite number anyways.

Add **rpi_power_switch** to **/etc/modules** and save



```
pi@raspberrypi: ~  
GNU nano 2.2.6 File: /etc/modules Modified  
# /etc/modules: kernel modules to load at boot time.  
#  
# This file contains the names of kernel modules that should be loaded  
# at boot time, one per line. Lines beginning with "#" are ignored.  
# Parameters can be specified after the module name.  
  
snd-bcm2835  
spi-bcm2708  
fbttft_device  
rpi_power_switch  
  
^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
```

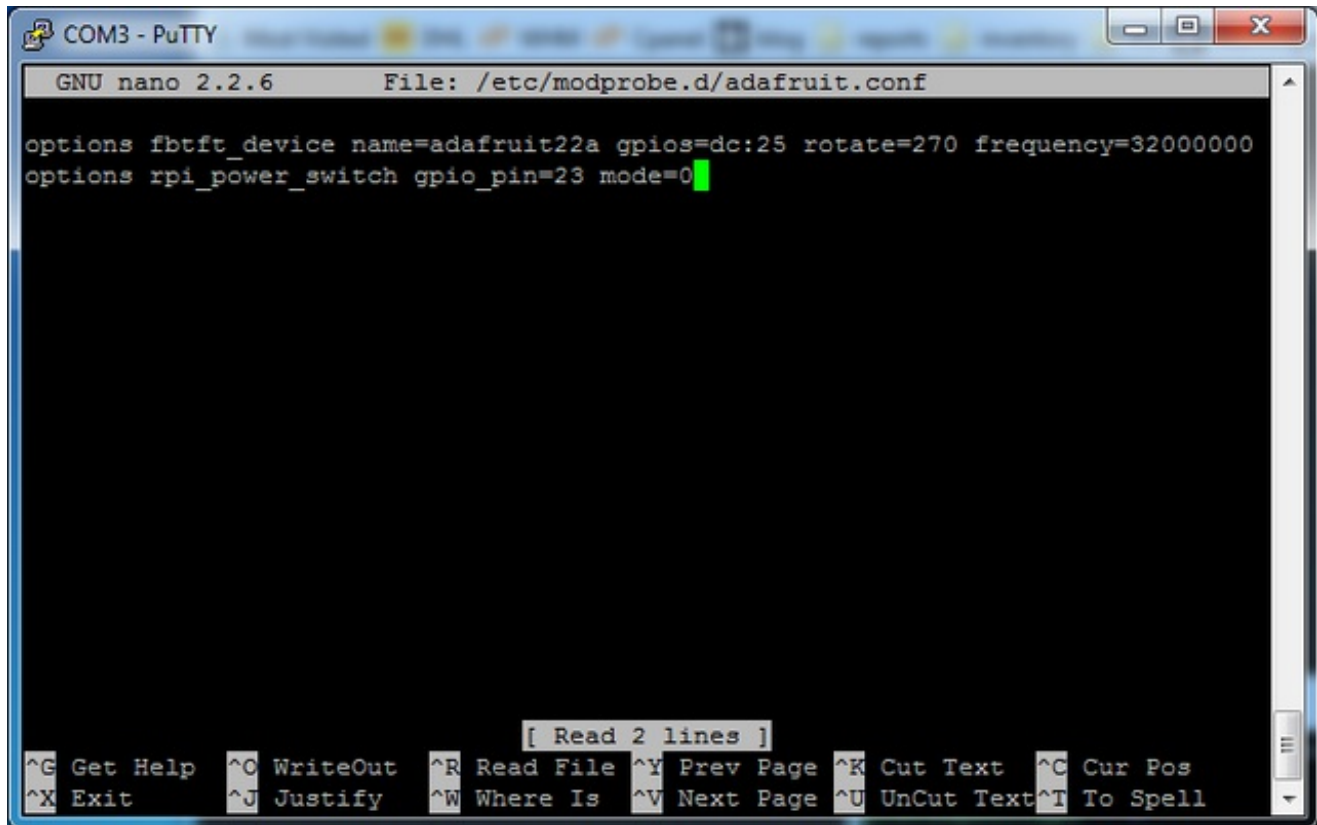
Now create a new conf file or edit our existing one with

sudo nano /etc/modprobe.d/adafruit.conf

and enter in the line

options rpi_power_switch gpio_pin=23 mode=0

Of course, change the **gpio_pin** setting to some other # if you wish. **mode=0** means it's a pushbutton *not* a switch. If you happen to install an on/off switch, use **mode=1**

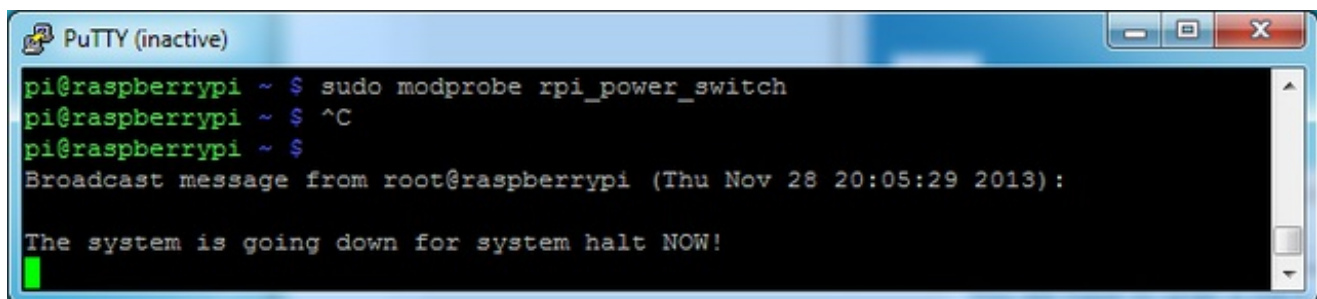


```
COM3 - PuTTY
GNU nano 2.2.6 File: /etc/modprobe.d/adafruit.conf

options fbtft_device name=adafruit22a gpios=dc:25 rotate=270 frequency=32000000
options rpi_power_switch gpio_pin=23 mode=0

[ Read 2 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
```

To make it active immediately run **sudo modprobe rpi_power_switch**



```
PuTTY (inactive)
pi@raspberrypi ~ $ sudo modprobe rpi_power_switch
pi@raspberrypi ~ $ ^C
pi@raspberrypi ~ $
Broadcast message from root@raspberrypi (Thu Nov 28 20:05:29 2013):

The system is going down for system halt NOW!
```

Boot to X Windows on PiTFT

To enable booting straight to X windows on the PiTFT follow the steps below. First make sure a display configuration which would conflict is **not** present by executing in a terminal on the Pi:

```
sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~
```

Don't worry if the command fails with an error that the file doesn't exist. This failure is normal and should happen on a good PiTFT install. You can ignore it and move on.

Next run the command below to open the nano text editor as root and create the

file **/usr/share/X11/xorg.conf.d/99-pitft.conf**:

```
sudo nano /usr/share/X11/xorg.conf.d/99-pitft.conf
```

When the editor loads to a blank file, copy in the text below:

```
Section "Device"
  Identifier "Adafruit PiTFT"
  Driver "fbdev"
  Option "fbdev" "/dev/fb1"
EndSection
```

Then save the file by pressing **Ctrl-O** and then **enter**, and finally exit by pressing **Ctrl-X**.

The step above will create a configuration file which tells X windows that it should use the PiTFT framebuffer (located at /dev/fb1) by default when it runs.

At this point you can use the raspi-config tool to enable booting to desktop just like normal on the Pi. Run the following command:

```
sudo raspi-config
```

Then pick the **Enable Boot to Desktop/Scratch** option and choose if you want to boot to the console, desktop, or scratch environment. After exiting the tool and rebooting you should see the Pi load X windows on the PiTFT after (be patient it can take around 30 seconds to load).

If you want to disable booting to X, just use the raspi-config command again to choose the console boot option

FAQ

We'll fill these in as we get more questions,[you can also check out the F.A.Qs for the 2.8" PiTFT, as many of the same questions may apply!](http://adafru.it/dJ2) (<http://adafru.it/dJ2>)

- ## Schematic

The image shows a detailed PCB layout for the Adafruit 2.2in TFT HAT rev A. The board is populated with a Raspberry Pi Model B+, an ICS AXP803 DC-DC converter, a 2.2in TFT display, and a backlight. The layout includes a top layer with components and a bottom layer with a 1x25 header. A 3D isometric view of the assembled board is shown in the bottom right corner. The PCB is labeled with dimensions and component values.

Fabrication Print



