



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



□

Adafruit 2.8" PiTFT - Capacitive Touch

Created by lady ada



Last updated on 2016-11-26 04:43:44 AM UTC

Guide Contents

Guide Contents	2
Overview	4
Assembly	9
Easy Install	10
Ready to go image	10
DIY Installer script	10
Step 1. Expand Filesystem	11
Step 2. Install new Kernel	11
Step 3. Enable & Configure the PiTFT	13
Detailed Installation	15
Before you start	15
Download & Install Kernel	15
Capacitive Touchscreen Configuration	25
If you are running FT6236 driver	26
If you are running FT6X06 driver	26
Event Testing	27
TSLIB calibration	28
X11 Calibration	30
Calibration for other rotations	30
Console Configuration	32
Turn off Console Blanking	35
Raspbian Jessie	35
Raspbian Wheezy	35
Playing Videos	37
How To Play Videos	37
Converting/Resizing Videos	39
Displaying Images	43
Using FBCP	45
Backlight Control	47
Extras!	49
Tactile switch as power button	49

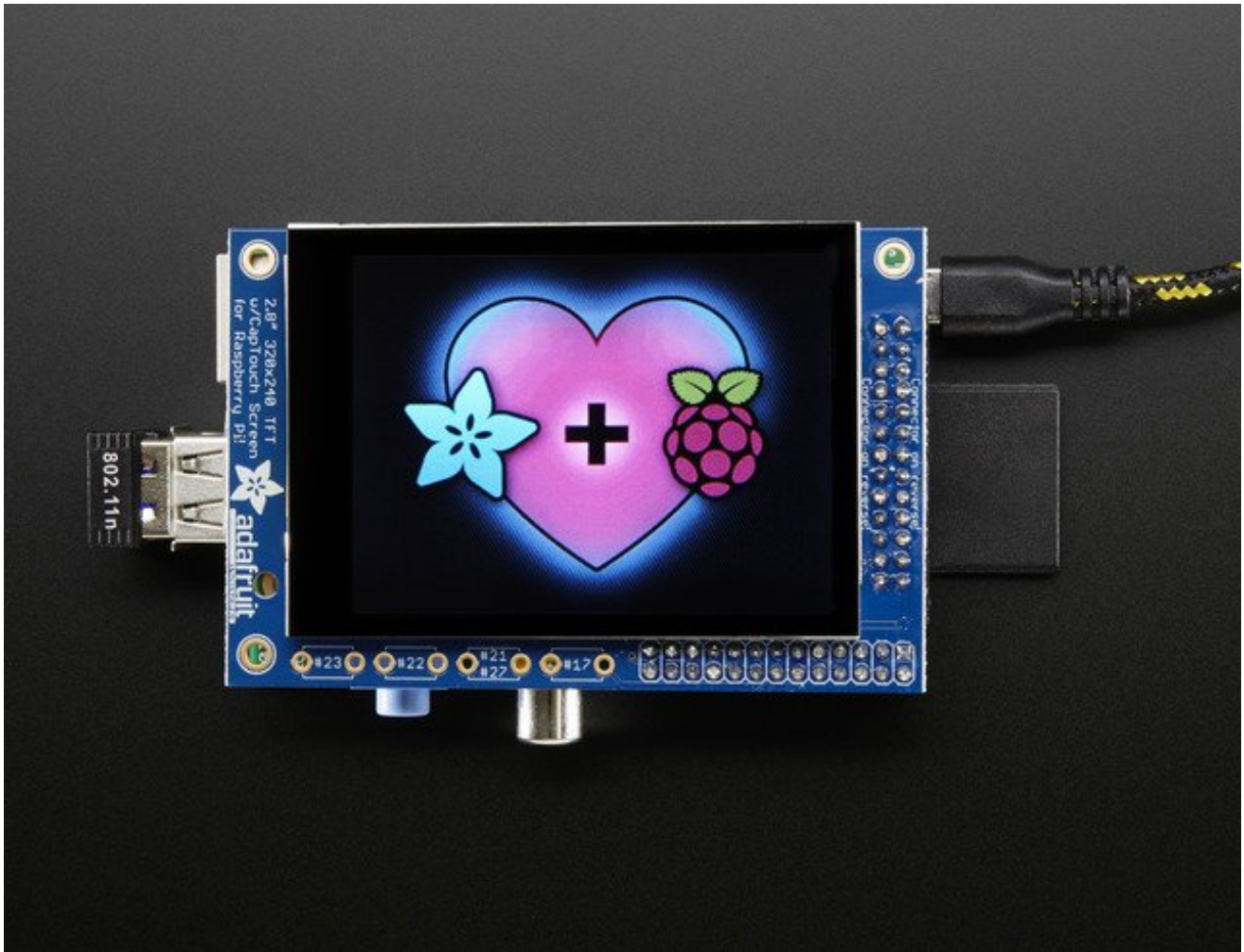
Making it easier to click icons in X	50
Boot to X Windows on PiTFT	51
Right-click on a touchscreen	52
Gesture Input	54
Installation	54
Usage	54
PiTFT PyGame Tips	58
Install pip & pygame	58
Ensure you are running SDL 1.2	59
F.A.Q.	62
Downloads	63
Files	63
Schematic for Pi 1 Version	63
Schematic for PiTFT Plus (B+/Pi 2 shape)	64
Fabrication Print (Pi 1 Version)	64
Fabrication Print (B+/Pi 2 Version)	66

Overview

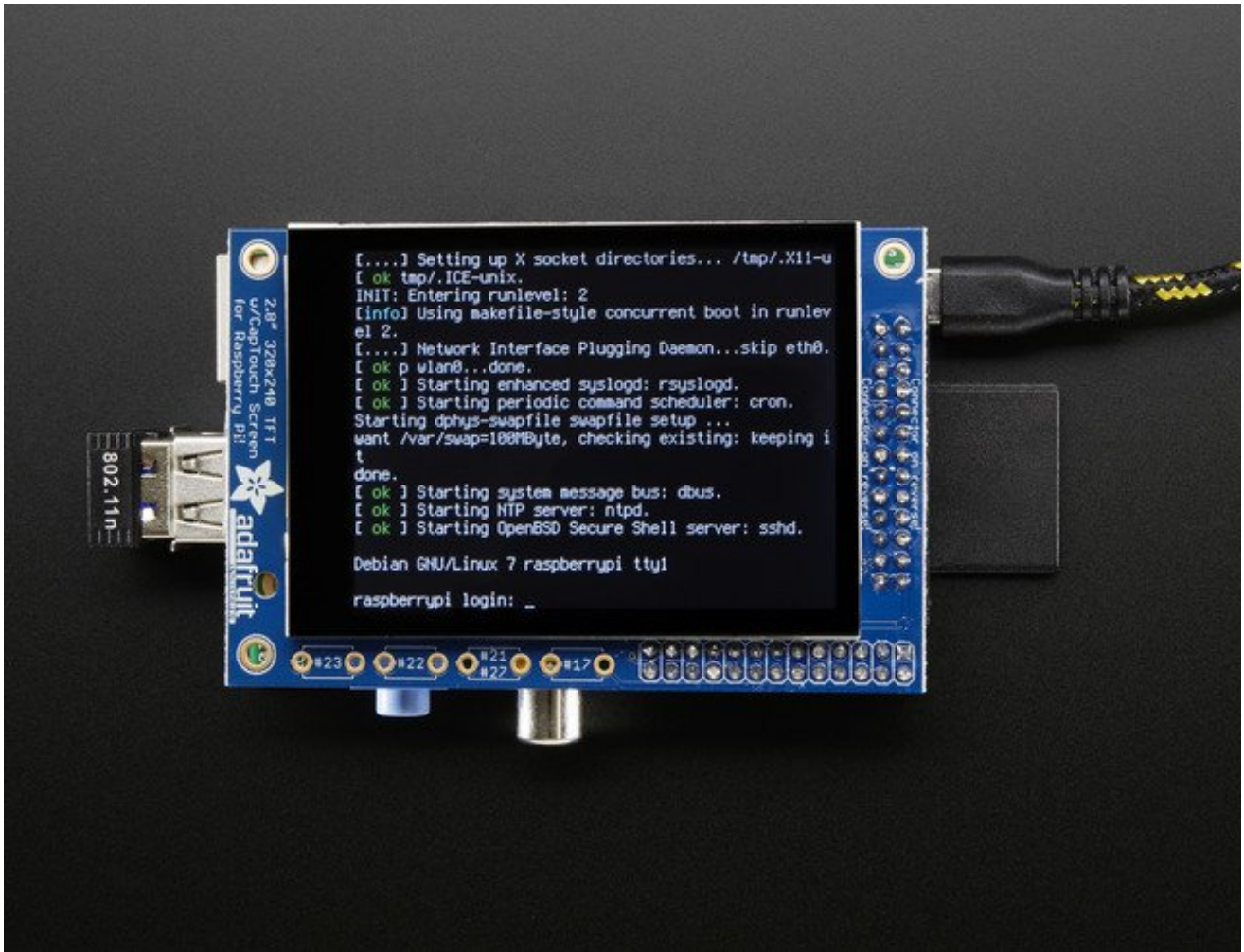


Our best-selling PiTFT just got a fancy upgrade, now we have a version with a **capacitive touchscreen!** That's right, instead of a resistive touchscreen, which requires a fingernail or stylus, you can now use a fingerpad. The screen looks much nicer, with a black bezel and glass overlay.

Featuring a 2.8" display with 320x240 16-bit color pixels and a capacitive touch overlay. 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!



Uses the hardware I2C Pins (SDA & SCL), SPI pins (SCK, MOSI, MISO, CE0) as well as GPIO #25 and #24. All other GPIO are unused. Since we had a tiny bit of space, there's 4 spots for optional slim 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. See below for the link to get the optional tact switches, they're not included.

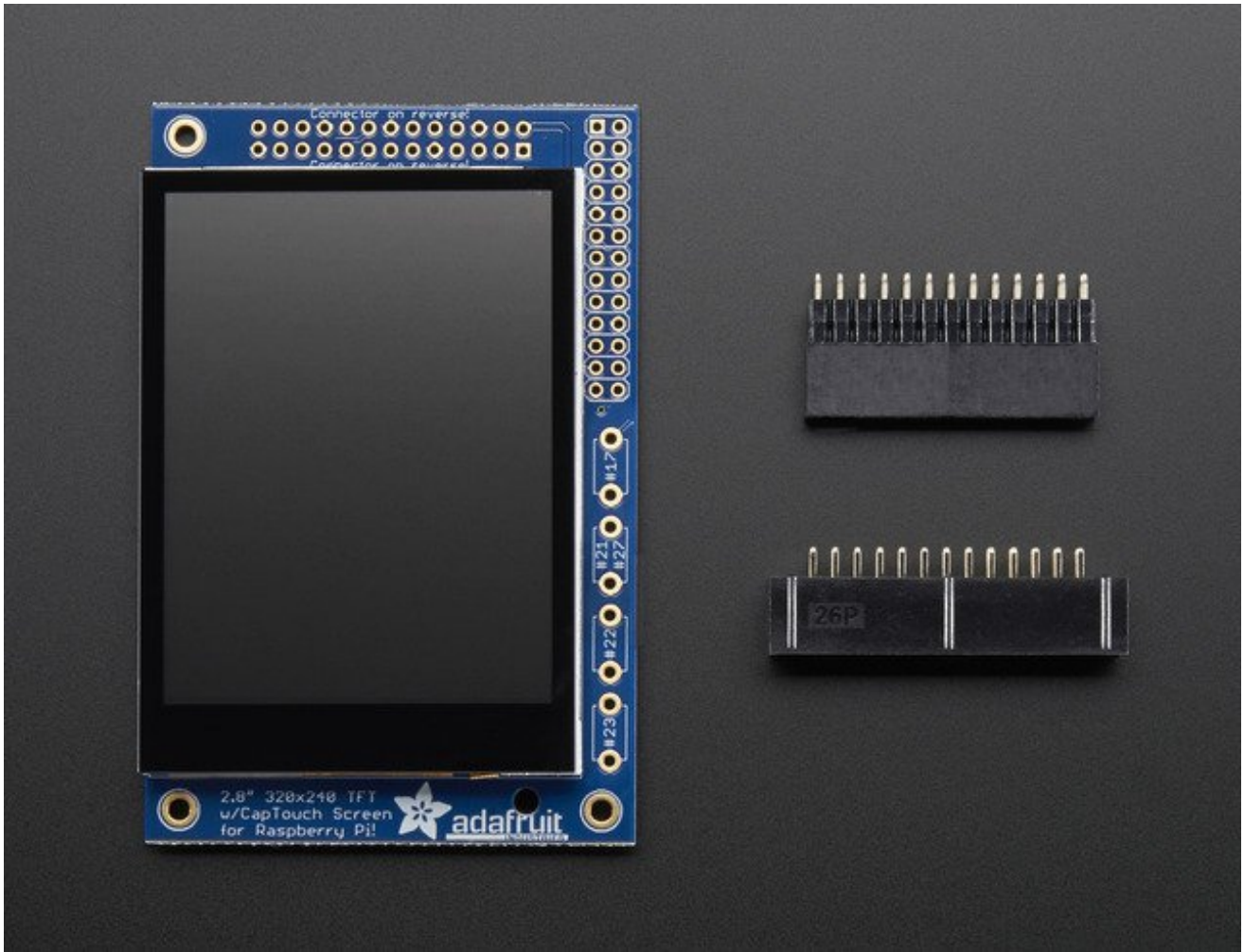


The screen is the same size as the resistive type so you can use this with the PiTFT PiBow or any other enclosure you may already have. We also use the same SDL device and signals so PyGame and X11 based programs can be swapped in with no changes in code.



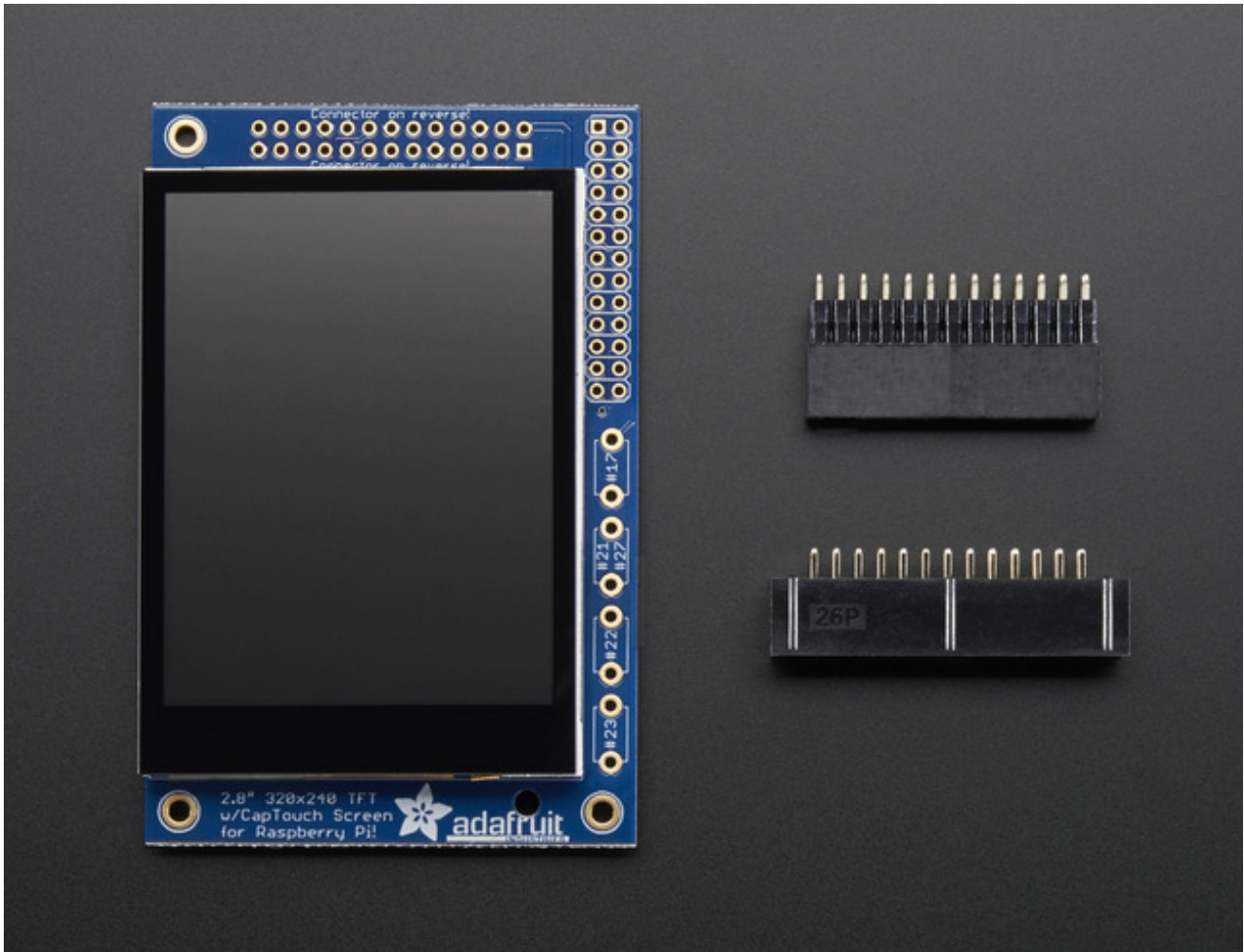
It's designed to fit nicely onto the Pi Model A or B rev 2 but also works perfectly fine with the Model B+ as long as you don't mind the PCB overhangs the USB ports by 5mm, see the photos above. Model B rev 1 have an older layout for the I2C pins and won't be able to use the touch screen

This version comes as a mini-kit, with a 2x13 extra-tall female header (to connect the plate to the Pi) and a 2x13 male header that can be used to connect an IDC cable or cobbler from the side. The photos above also show the optional installed slim tactile buttons. [The tactile buttons are not included, but you can pick up a pack of 20 here. \(http://adafru.it/1489\)](http://adafru.it/1489) Some basic soldering is required to install the headers. [You can also pick up an extra-long Pi stacking header if you want to install that instead of the 2x13 female header installed. \(http://adafru.it/1112\)](http://adafru.it/1112)



Assembly

We are now selling these displays pre-assembled - skip this step if your PiTFT is not a mini-kit



This section is identical to the PiTFT Resistive 2.8" so please visit that page to complete assembly of this Pi Plate

[Visit the 2.8" Resistive PiTFT Assembly Page](http://adafru.it/DDQ)

<http://adafru.it/DDQ>



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)

This image is customized for the CAPACITIVE touch 2.8" TFT, also known as PID #1983!
Not for PID #1601

[Download Jessie-based PiTFT 2.8" Capacitive Image for Pi 1, Pi 2 and Pi 3 \(Sept 23, 2016\)](http://adafru.it/saM)
<http://adafru.it/saM>

[Download Jessie Lite-based PiTFT 2.8" Capacitive Image for Pi 1, Pi 2 and Pi 3 \(Sept 23, 2016\)](http://adafru.it/saN)
<http://adafru.it/saN>

Older images:

- [Raspbian Jessie 2016-03-25-based image \(http://adafru.it/mAc\)](http://adafru.it/mAc)
- [Raspbian Jessie Lite 2016-03-25-based image \(http://adafru.it/mAd\)](http://adafru.it/mAd)
- [Raspbian Jessie 2015/09/24-based image \(http://adafru.it/iDy\)](http://adafru.it/iDy)
- [Raspbian Wheezy 2015/09/24-based image \(http://adafru.it/idz\)](http://adafru.it/idz)
- [Raspbian 2014/09/18-based image \(http://adafru.it/e11\)](http://adafru.it/e11)
- [Raspbian 2014/06/20-based image \(http://adafru.it/dSO\)](http://adafru.it/dSO)
- [Raspbian image from 2015/03/03 \(http://adafru.it/eUI\)](http://adafru.it/eUI)

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

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 filesystem is expanded, download and install the 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, and tells apt that it should give a very high priority to packages installed there.

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

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.

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

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

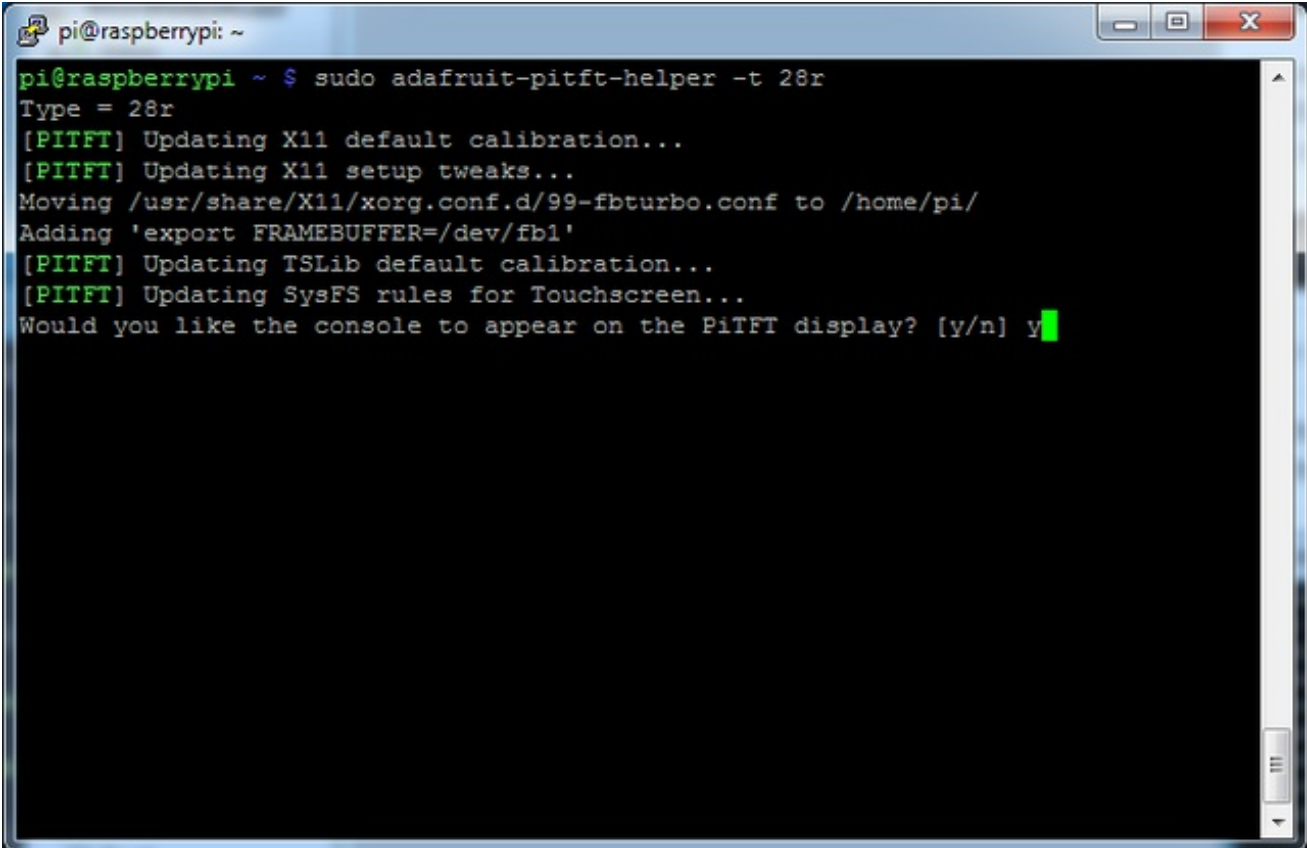
Step 3. Enable & Configure the PiTFT

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 28c
```

This will install the "2.8 Capacitive" 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 ~ $ 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
```

That's it!

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



Detailed Installation

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 2.8" TFT and capacitive 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 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.

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 and make it the default source for packages it hosts



```
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

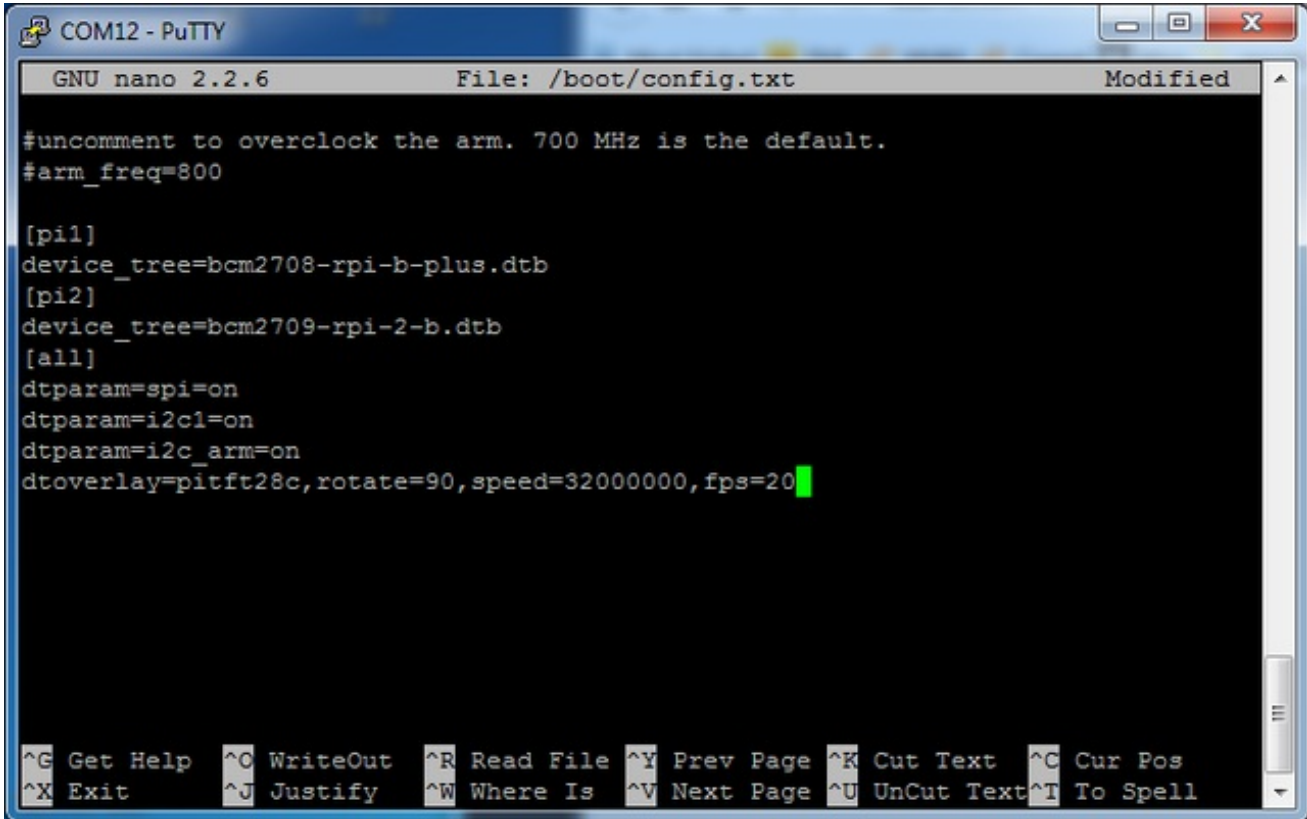
pi@raspberrypi ~/Adafruit-Occidentalis $ sudo apt-get install raspberrypi-bootloader
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
The 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=pitft28c,rotate=90,speed=32000000,fps=20
```



```
COM12 - PuTTY
GNU nano 2.2.6 File: /boot/config.txt Modified
#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=pitft28c,rotate=90,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

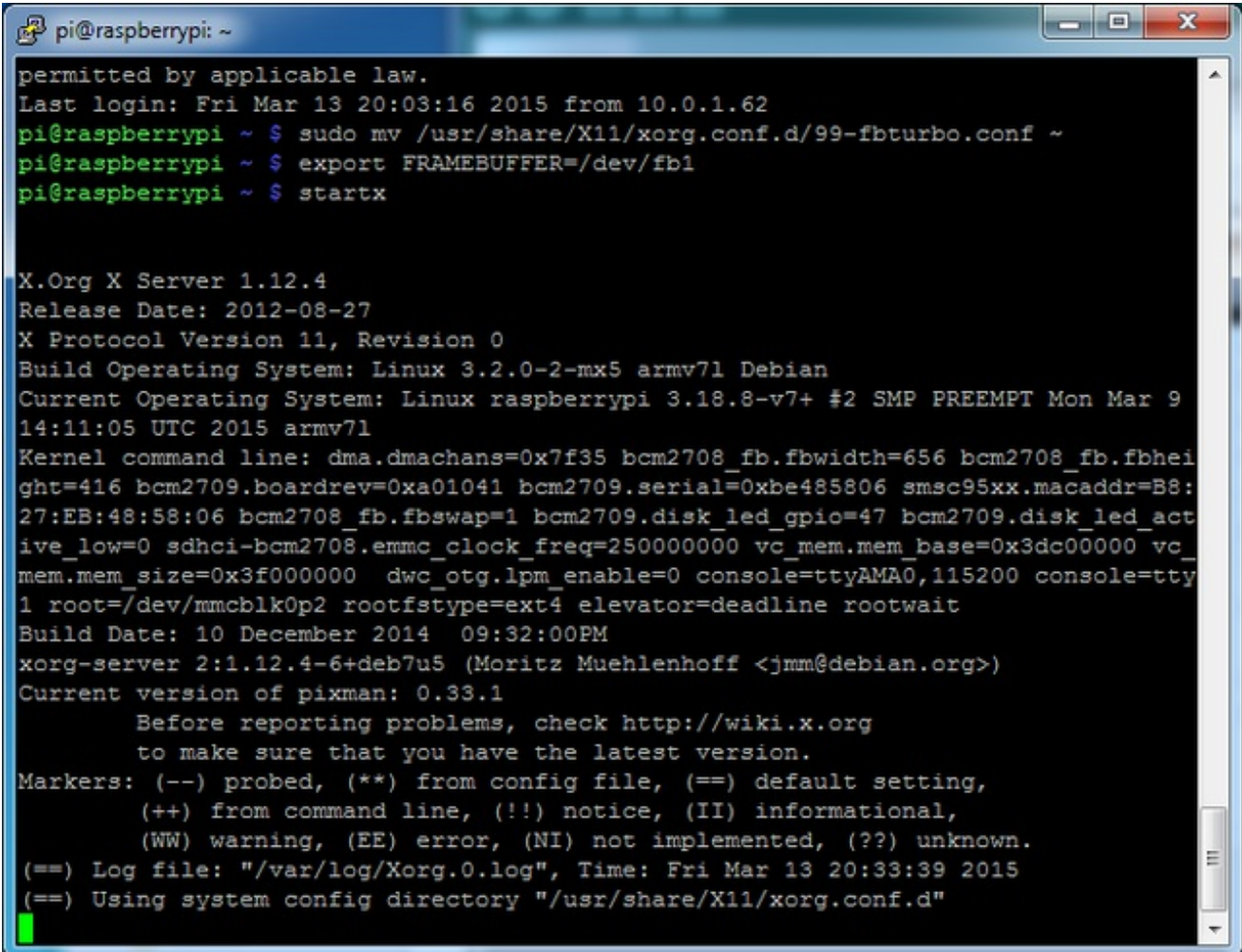
sudo reboot (if you have the TFT plate installed already)

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"
```

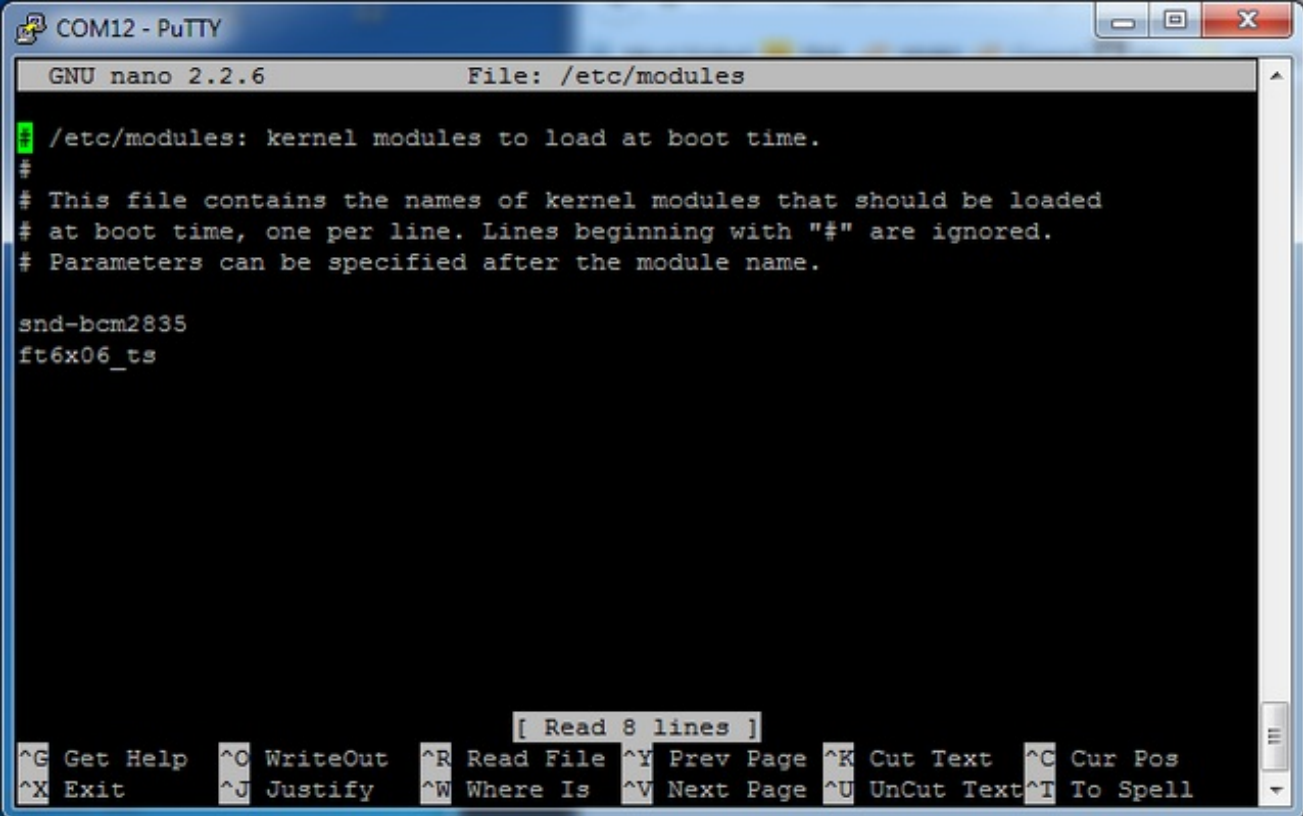
You should see the Pi desktop show up on the TFT! Congrats, you've completed the first test perfectly.

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 **ft6x06_ts** on a line at the end



```
COM12 - PuTTY
GNU nano 2.2.6 File: /etc/modules
/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
ft6x06_ts

[ Read 8 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
```

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 FT6206 (a.k.a. FT6x06) detection and the ILI9340 screen frequency as highlighted here

```
COM12 - PuTTY
[ 4.543920] fbtft_of_value: buswidth = 8
[ 4.572892] fbtft_of_value: debug = 0
[ 4.590365] fbtft_of_value: rotate = 90
[ 4.597653] fbtft_of_value: fps = 20
[ 4.867695] graphics fb1: fb_ili9340 frame buffer, 320x240, 150 KiB video mem
ory, 4 KiB DMA buffer memory, fps=20, spi0.0 at 32 MHz
[ 4.883122] bcm2708_spi 3f204000.spi: SPI Controller at 0x3f204000 (irq 80)
[ 4.891941] bcm2708_spi 3f204000.spi: SPI Controller running in dma mode
[ 4.900748] bcm2708_i2c_init_pinmode(1,2)
[ 4.906728] bcm2708_i2c_init_pinmode(1,3)
[ 4.922822] bcm2708_i2c 3f804000.i2c: BSC1 Controller at 0x3f804000 (irq 79)
(baudrate 100000)
[ 6.282948] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[ 6.521625] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[ 7.064131] input: ft6x06_ts as /devices/virtual/input/input0
[ 7.229885] i2c-core: driver [ft6x06_ts] using legacy suspend method
[ 7.238487] i2c-core: driver [ft6x06_ts] using legacy resume method
[ 7.850318] random: nonblocking pool is initialized
[ 11.612592] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
[ 13.256120] smsc95xx 1-1.1:1.0 eth0: link up, 100Mbps, full-duplex, lpa 0x45E
1
[ 14.426639] Adding 102396k swap on /var/swap. Priority:-1 extents:22 across:
2974180k SSFS
pi@raspberrypi:~$
```

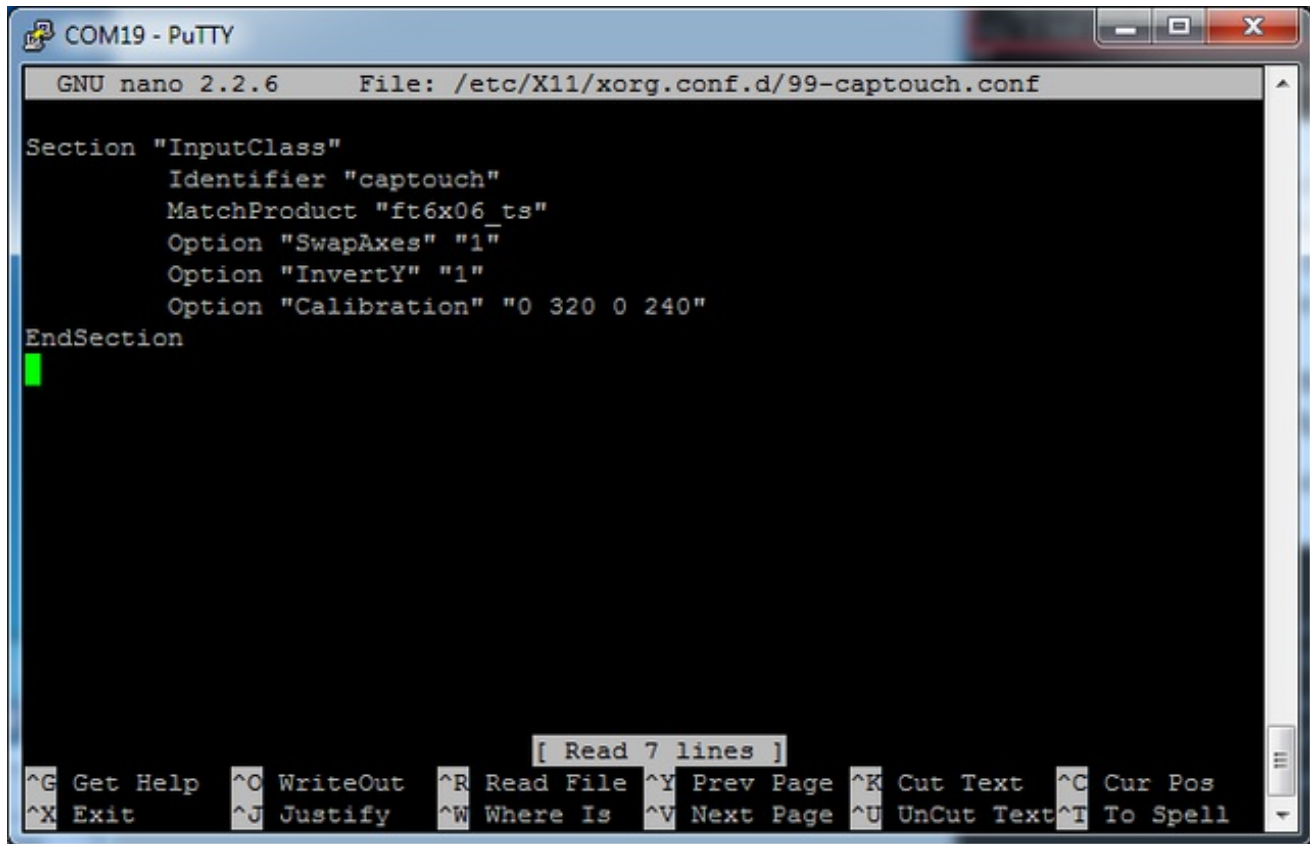
We can set up the touchscreen for **rotate=90** configuration by doing the following (for more delicate calibration or for other rotate=XX values, see the next section)

Create the directory and new calibration configuration file:

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

and enter in the following lines, then save.

```
Section "InputClass"  
    Identifier "captouch"  
    MatchProduct "ft6x06_ts"  
    Option "SwapAxes" "1"  
    Option "InvertY" "1"  
    Option "Calibration" "0 320 0 240"  
EndSection
```



```
COM19 - PuTTY
GNU nano 2.2.6 File: /etc/X11/xorg.conf.d/99-captouch.conf

Section "InputClass"
    Identifier "captouch"
    MatchProduct "ft6x06_ts"
    Option "SwapAxes" "1"
    Option "InvertY" "1"
    Option "Calibration" "0 320 0 240"
EndSection
█

[ Read 7 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
```

You can now try to run X again with

FRAMEBUFFER=/dev/fb1 startx

The touchscreen now works, you can try it out!



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
```

Capacitive Touchscreen 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 touchscreen



The nifty thing about capacitive touch screens is that they **do not require calibration!** The calibration is done 'in chip' on the screen itself. However, we still do need to tell the Pi how to read the capacitive chip.

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.

First up figure out if you have the FT62X6 driver or FT6236 driver by running **dmesg | grep ft6**