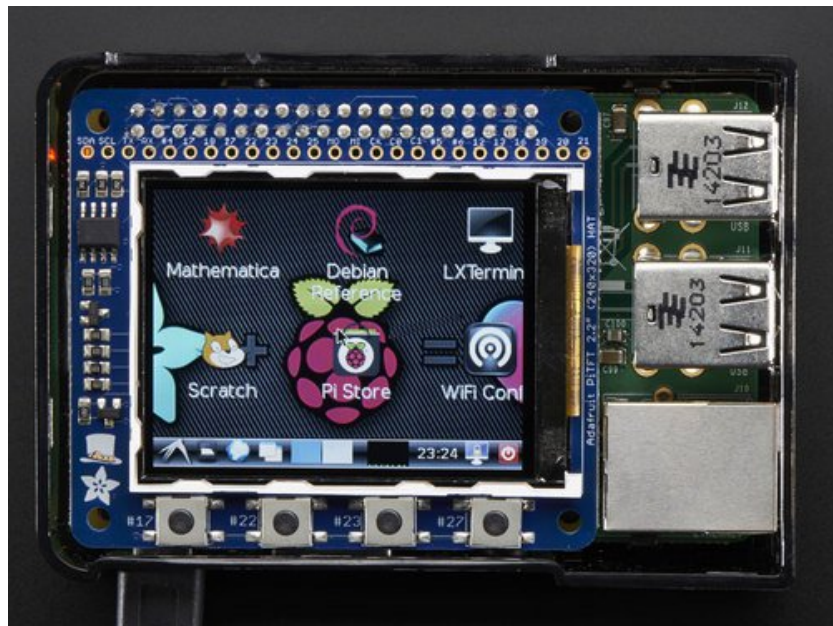


□

## Adafruit 2.2" PiTFT HAT - 320x240 Display

Created by lady ada



Last updated on 2016-10-23 03:00:09 PM UTC

## Guide Contents

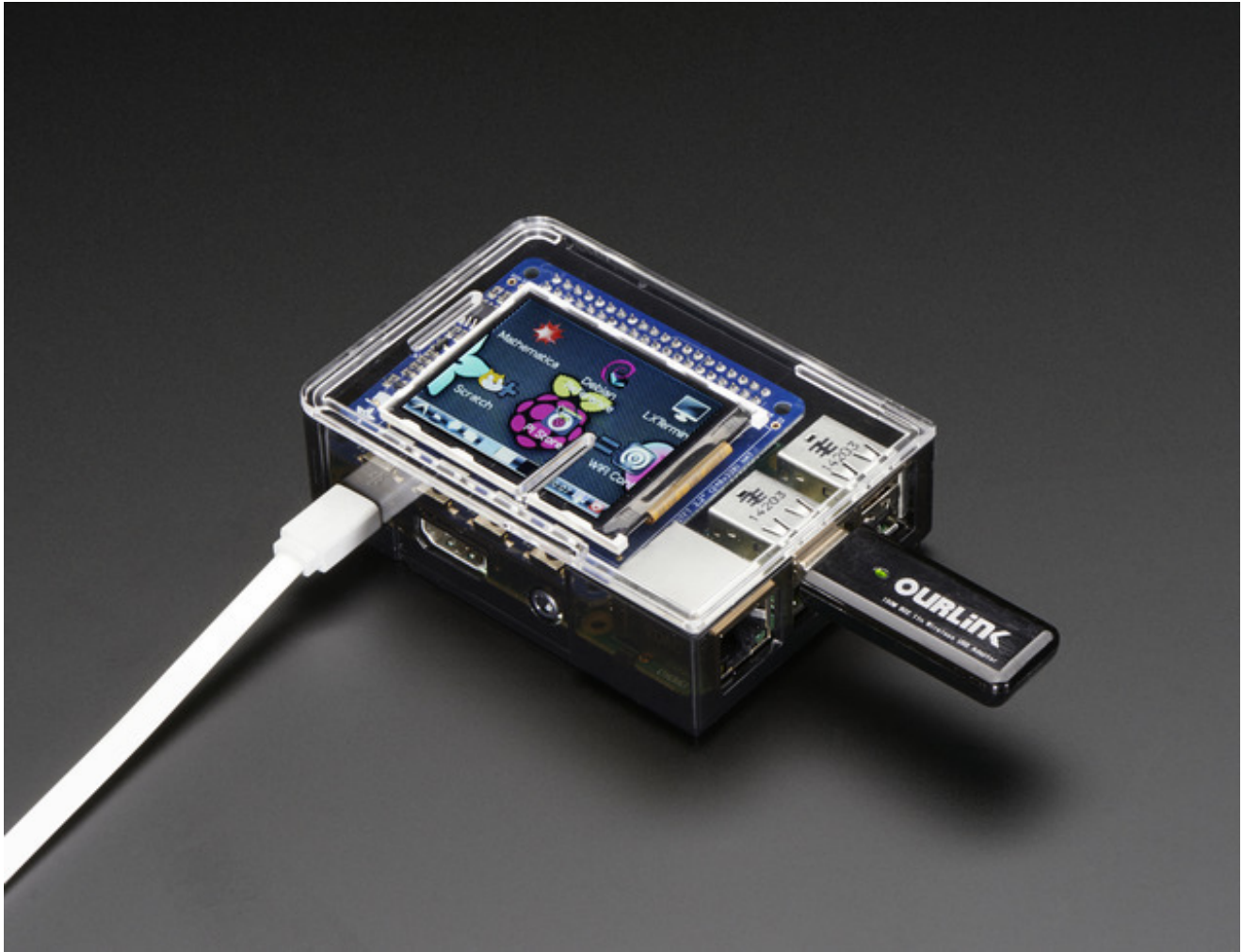
Guide Contents	2
Overview	3
Easy Install	7
Ready to go image	7
If you already have PiTFT / Notro Kernel	8
DIY Installer script	8
Step 1. Expand Filesystem	9
Step 2. Install new Kernel	9
Step 3. Enable & Configure the PiTFT	10
Detailed Installation	12
Before you start	13
Download & Install Kernel	13
Console Configuration	20
Turn off Console Blanking	23
Raspbian Jessie	23
Raspbian Wheezy	23
Playing Videos	25
How To Play Videos	25
Converting/Resizing Videos	27
Displaying Images	31
Using FBCP	33
Backlight Control	35
Extras!	36
Tactile switch as power button	36
Boot to X Windows on PiTFT	37
HELP! (FAQ)	39
Downloads	45
Schematic	45
Fabrication Print	45

# Overview



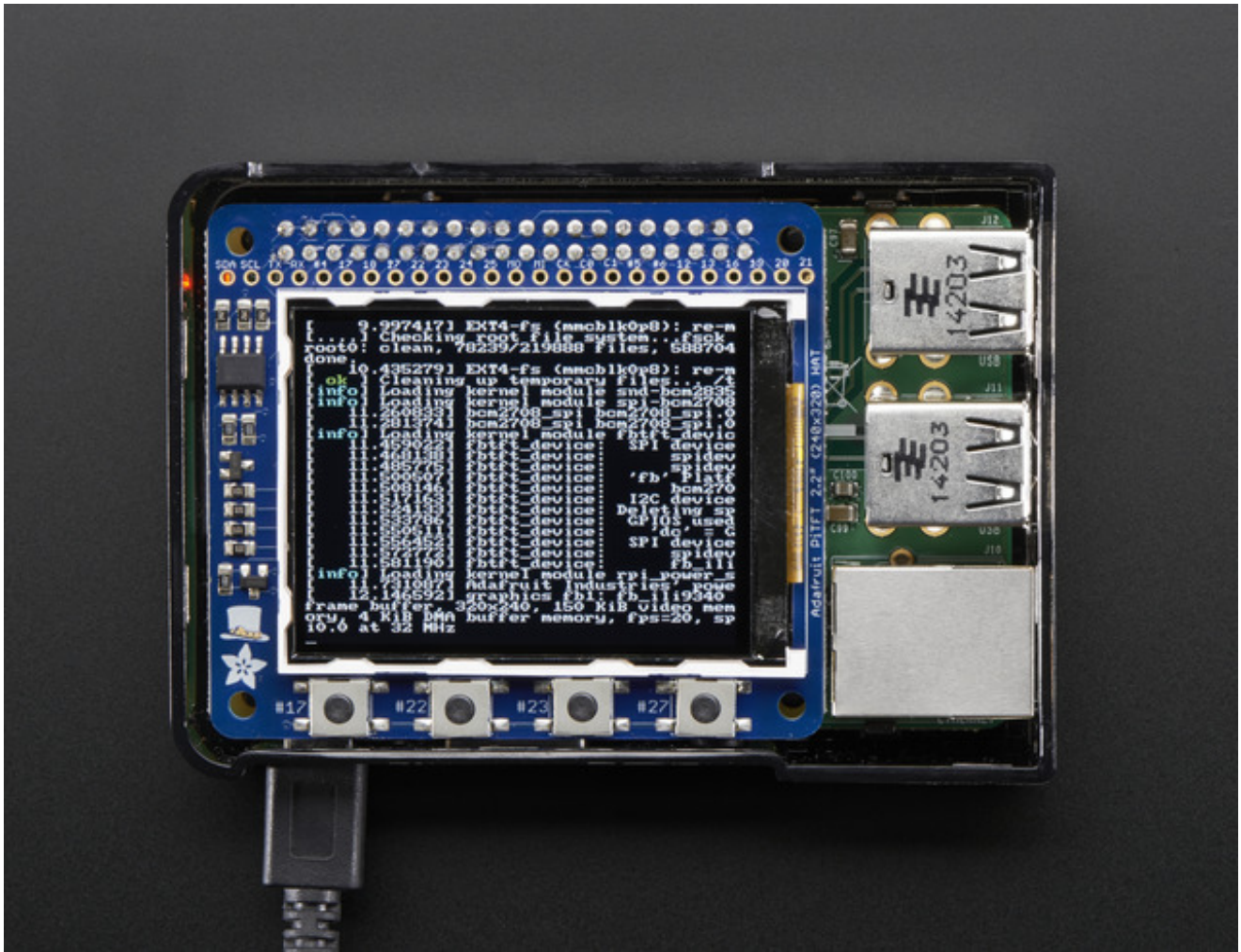
The cute PiTFT got even more adorable with this little primary display for Raspberry Pi in HAT form! It features a 2.2" display with 320x240 16-bit color pixels. The HAT uses the high speed SPI interface on the Pi and can use the mini display as a console, X window port, displaying images or video etc. 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

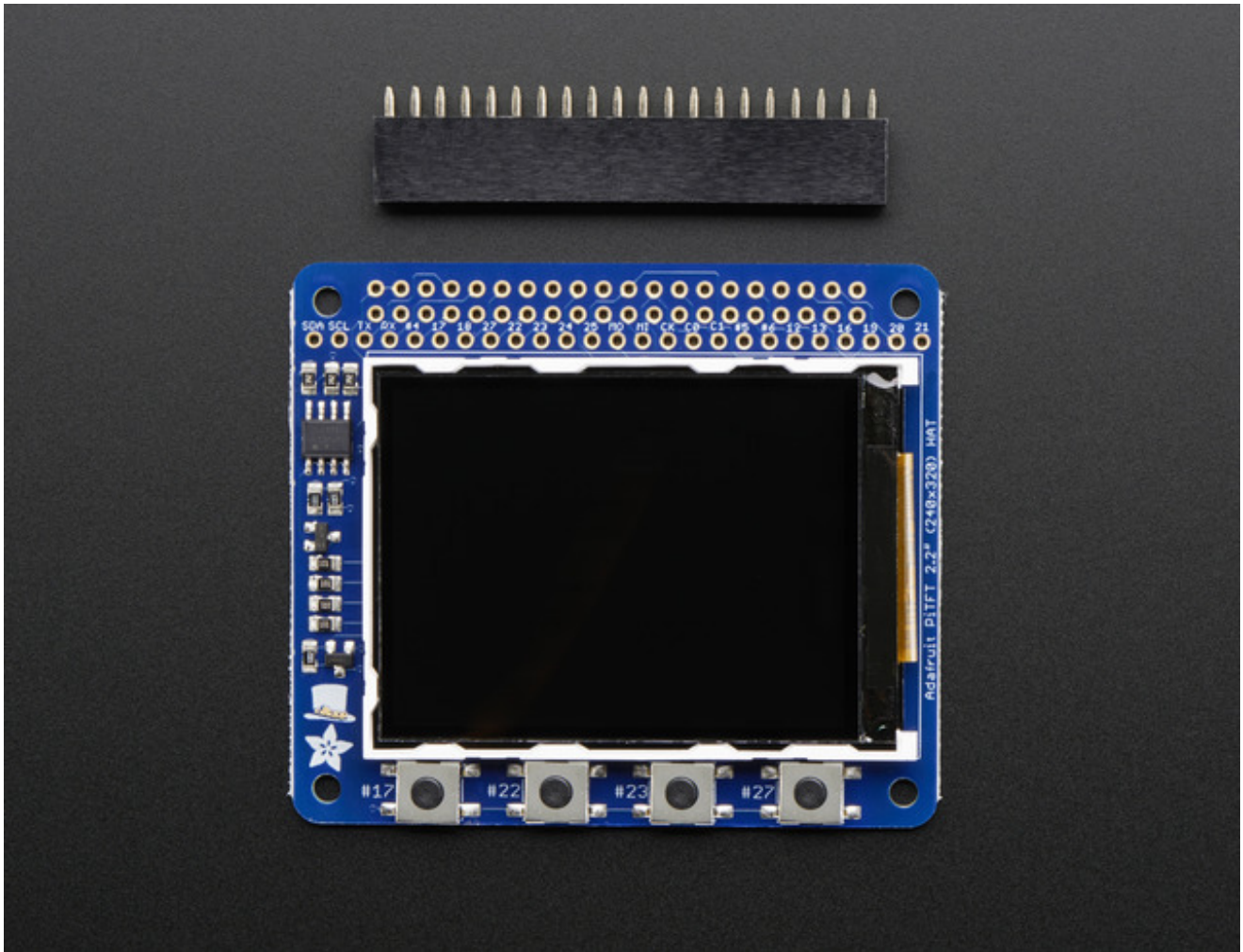


This design uses the hardware SPI pins (SCK, MOSI, MISO, CE0, CE1) as well as GPIO #25. All other GPIO are unused. Since we had a tiny bit of space, there's 4 flat tactile switches wired to four GPIOs, that you can use if you want to make a basic user interface. For example, you can use one as a power 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, 2, 3, Zero \(Sept 23, 2016\) http://adafru.it/sbg](http://adafru.it/sbg)

[Download Jessie Lite-based PiTFT 2.2" HAT Image for Pi 1, 2, 3, Zero \(Sept 23, 2016\) http://adafru.it/sbh](http://adafru.it/sbh)

Previous Images:

- [Raspbian Jessie 2016/03/25-based image \(http://adafru.it/mAe\)](http://adafru.it/mAe)
- [Raspbian Jessie Lite 2016/03/25-based image \(http://adafru.it/mAf\)](http://adafru.it/mAf)
- [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 fbttf_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
```

## 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!!!**

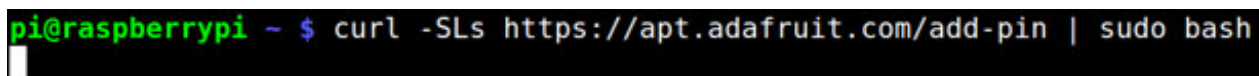
```
sudo raspi-config  
(expand filesystem)  
sudo reboot
```

# Step 2. Install new Kernel

Then, once the filesys 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



```
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.

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

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.

## 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 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
```

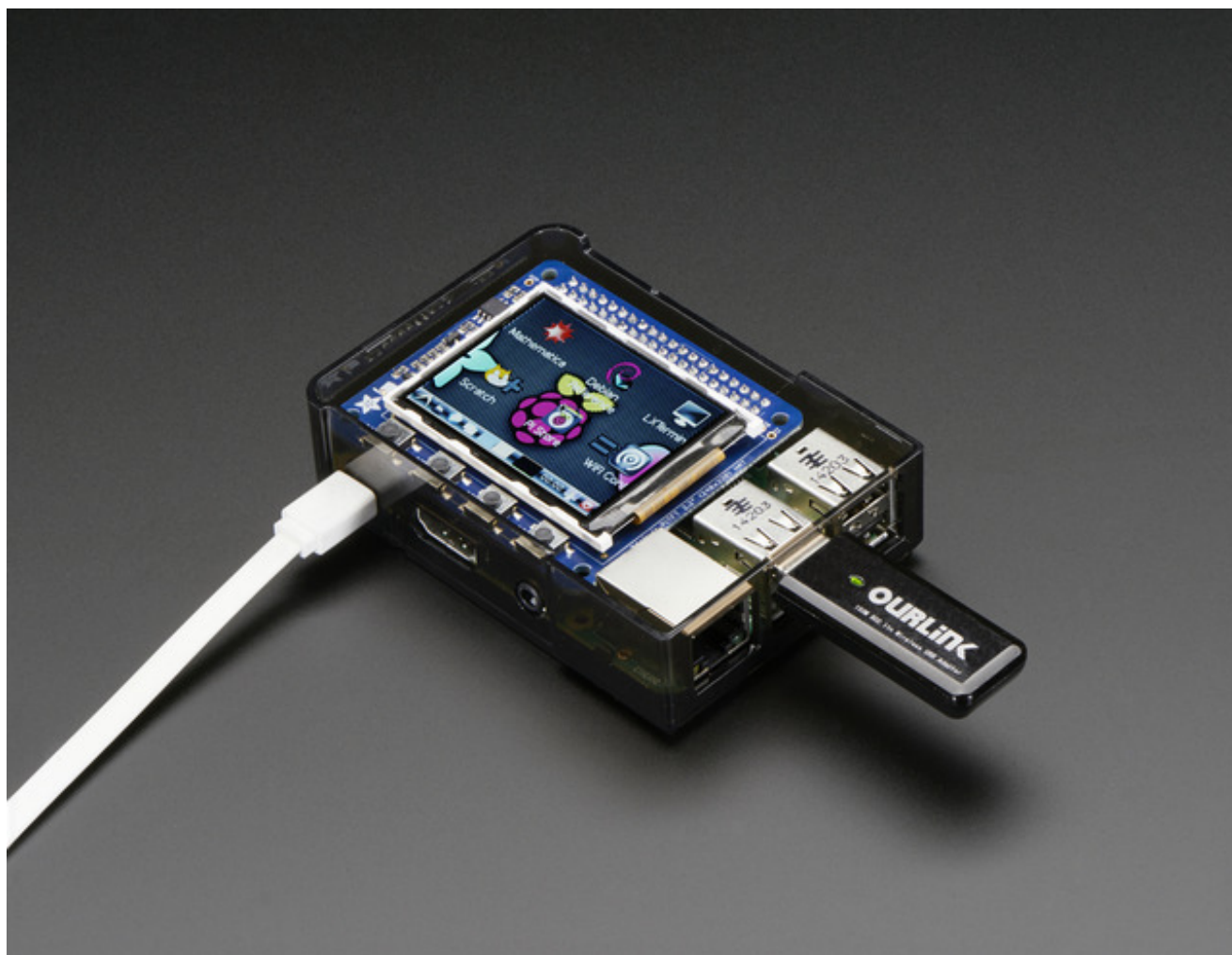
That's it!

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 2.2" 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 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 \(http://adafru.it/cY2\)](#), 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.

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)

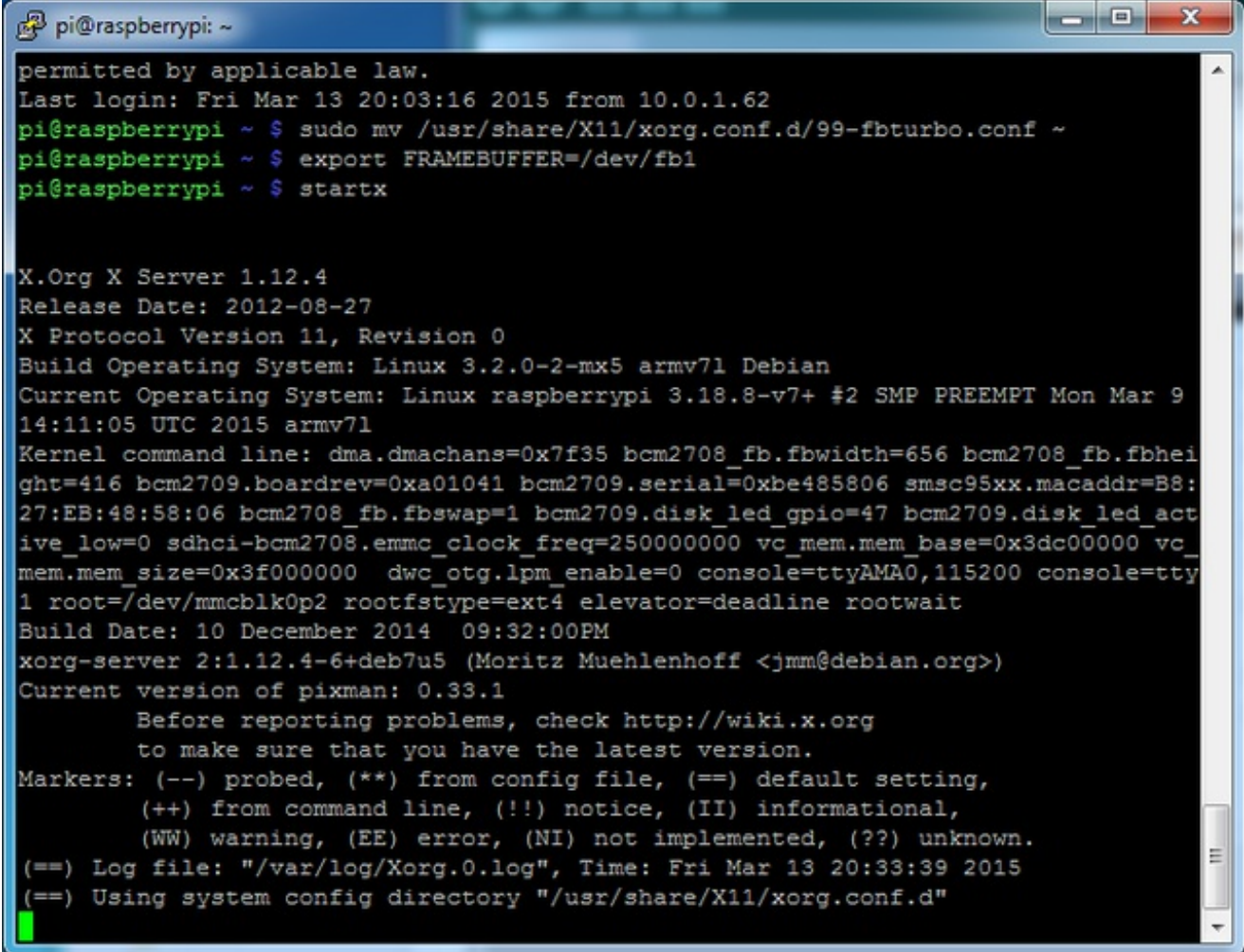


When the Pi restarts, the attached PiTFT should start out all white and then turn black. That means the kernel found the display and cleared the screen. If the screen did not turn black, that means that likely there's something up with your connection or kernel install. Solder anything that needs resoldering!

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
```



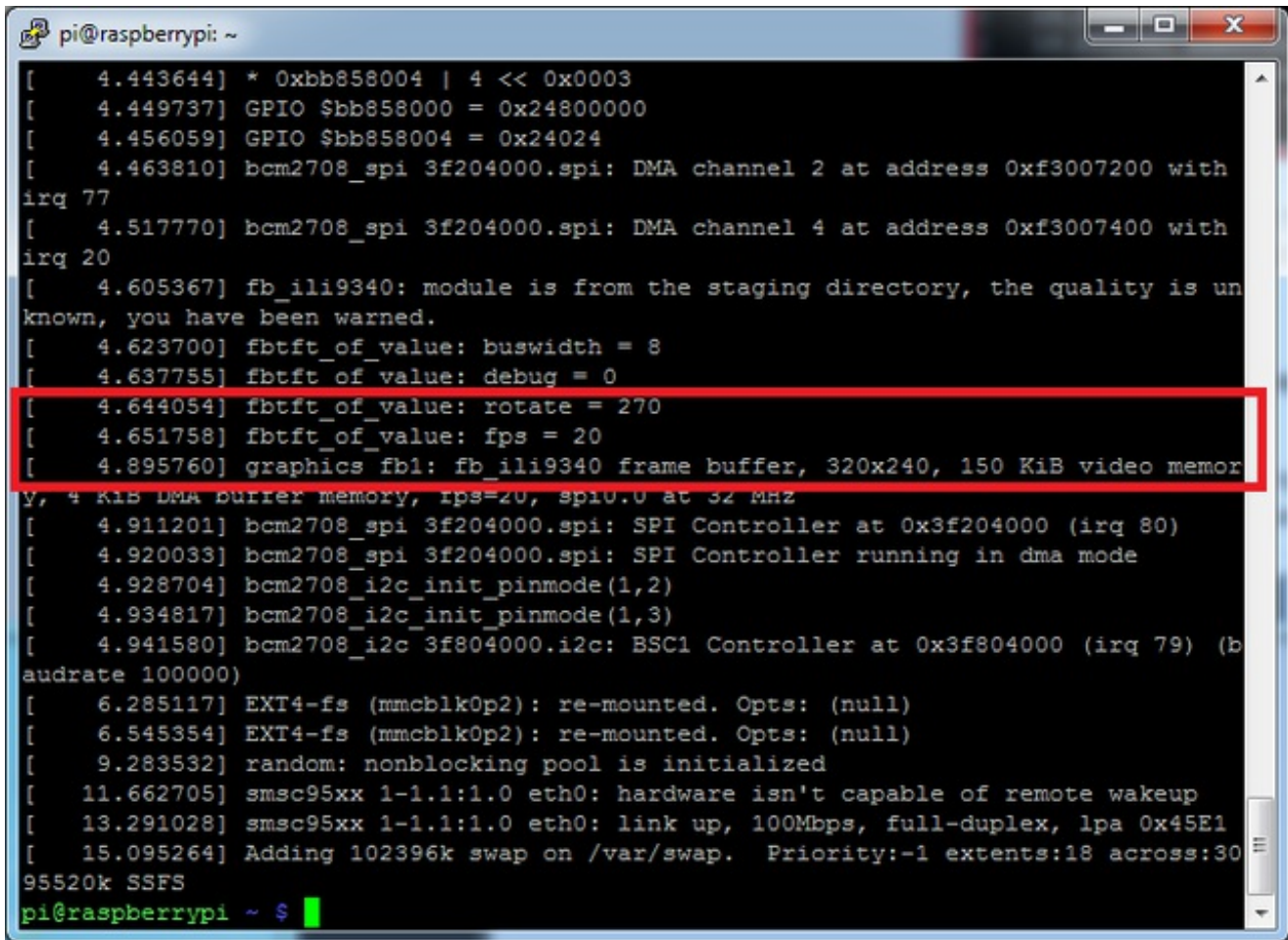
```
pi@raspberrypi: ~
permitted by applicable law.
Last login: Fri Mar 13 20:03:16 2015 from 10.0.1.62
pi@raspberrypi ~ $ sudo mv /usr/share/X11/xorg.conf.d/99-fbturbo.conf ~
pi@raspberrypi ~ $ export FRAMEBUFFER=/dev/fb1
pi@raspberrypi ~ $ startx

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 burrer 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

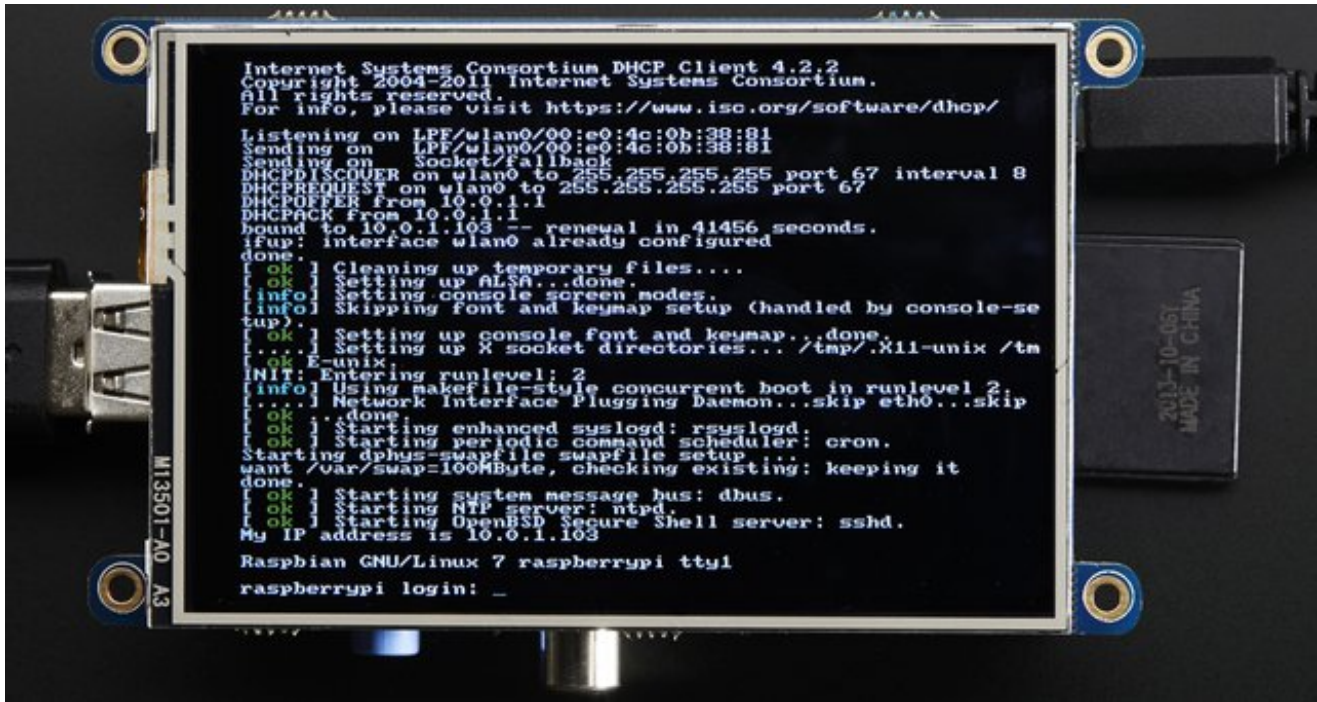
export FRAMEBUFFER=/dev/fb1

# 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
```

# 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/fbtt/wiki/Boot-console> (<http://adafruit.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
```

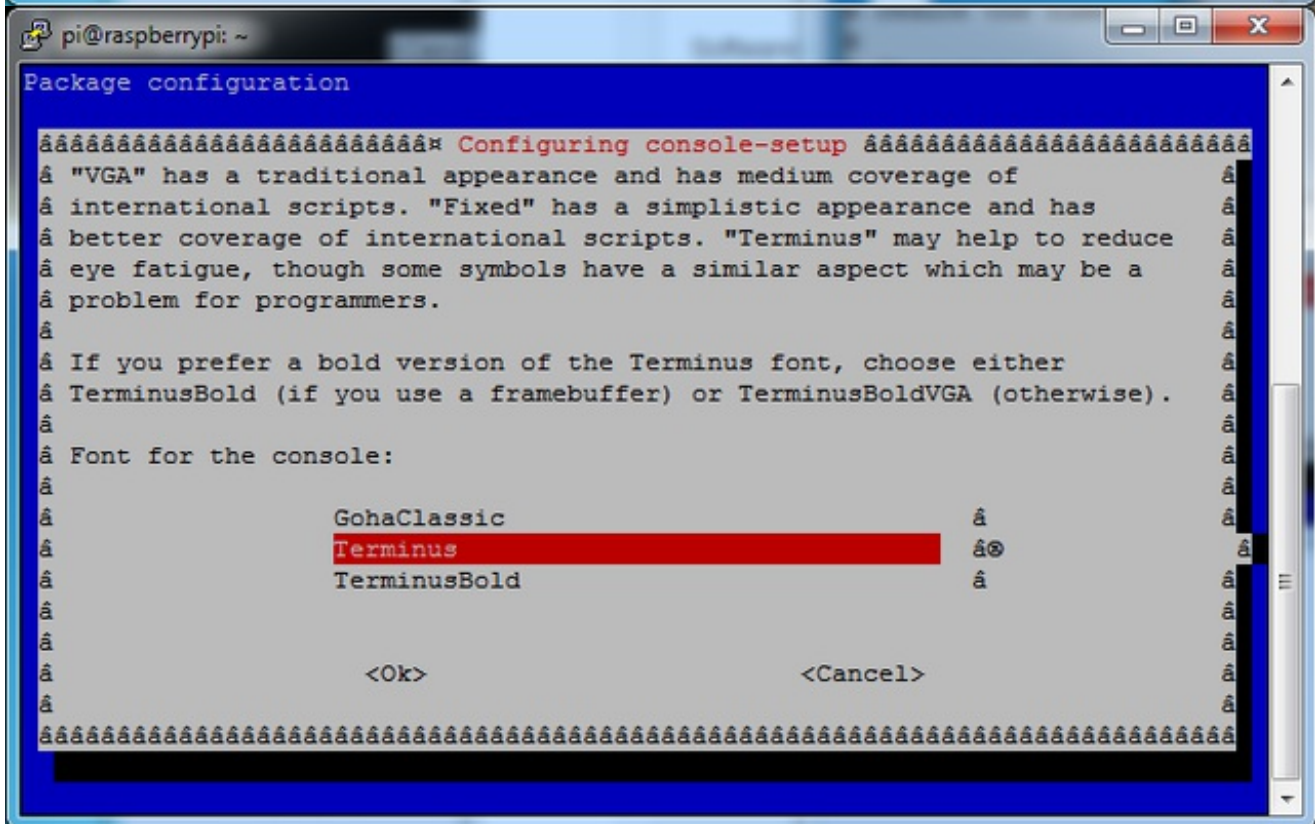
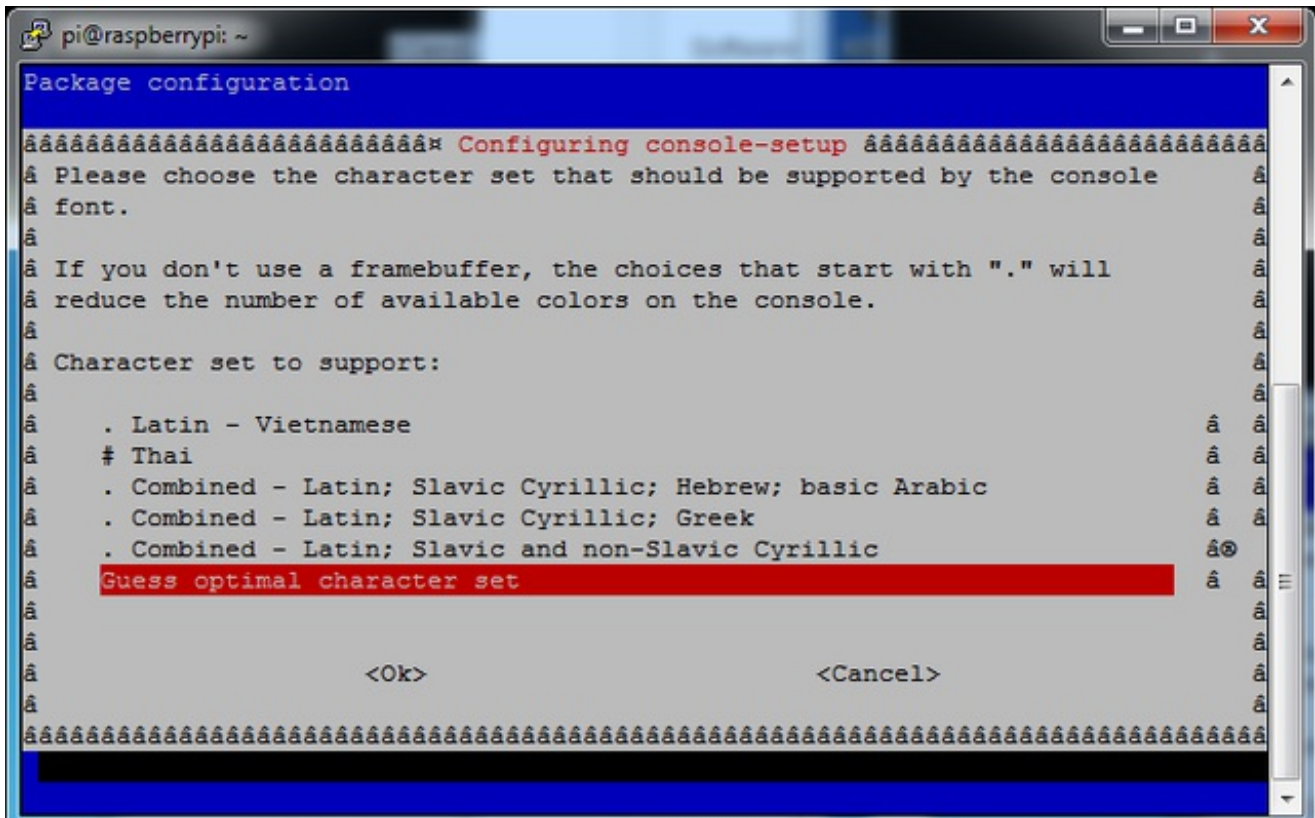
you can also edit it by putting the SD card into a computer and opening the same file.

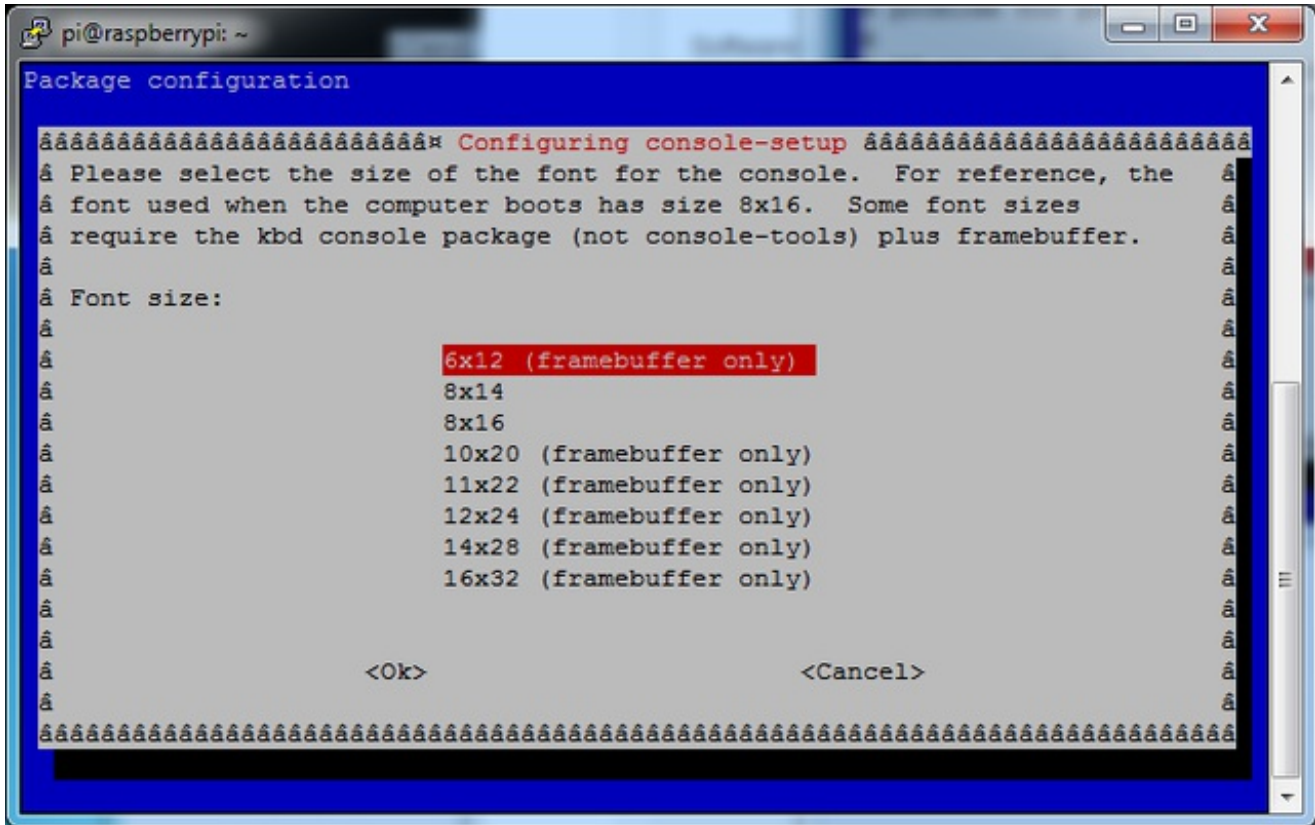
At the end of the line, find the text that says `rootwait` and right after that, enter in:  
`fbcon=map:10 fbcon=font:VGA8x8` then save the file.

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**







## 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 `finalexit 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





# 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

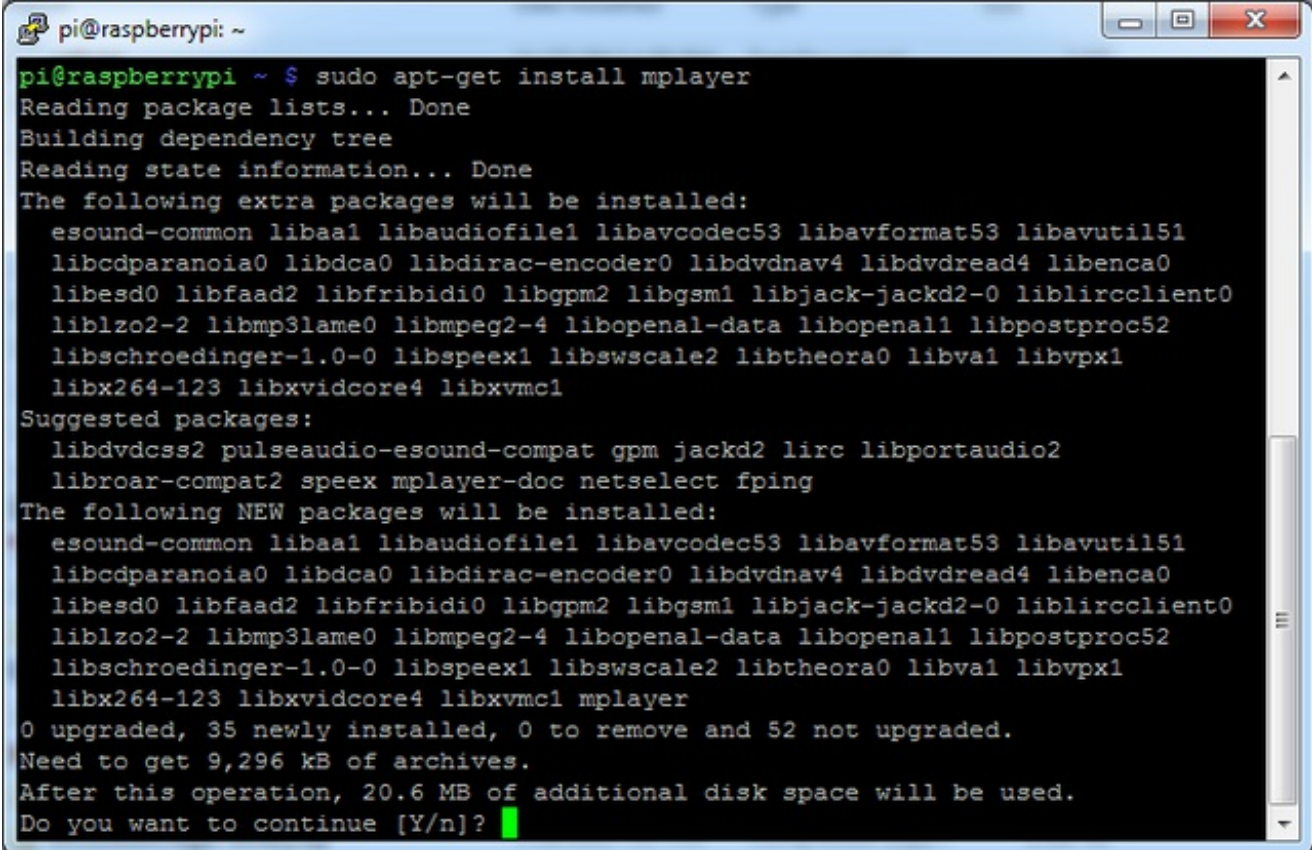
this, run `sudo raspi-config` to expand the SD card so you don't run out of space!

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

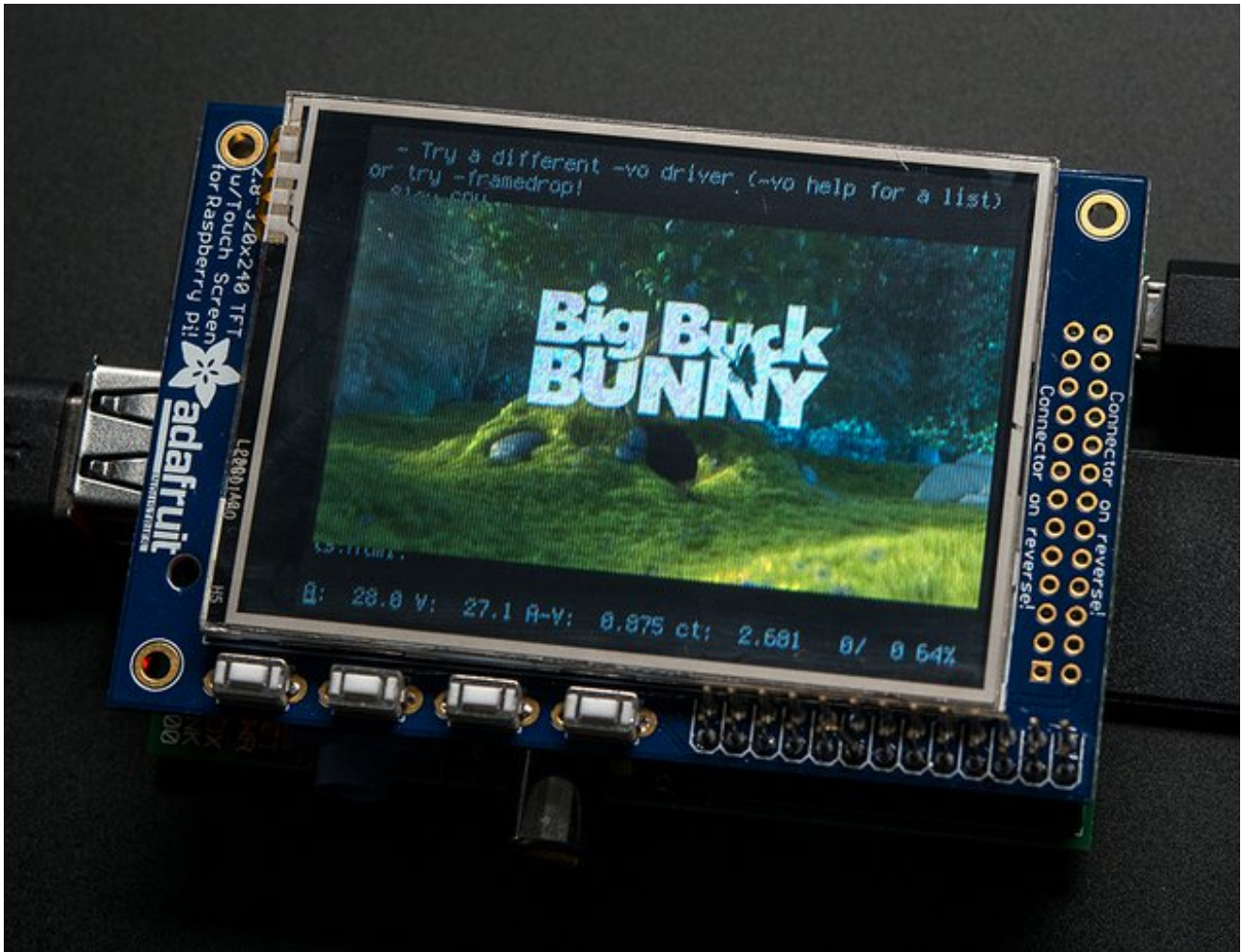


```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo apt-get install mplayer  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  esound-common libaa1 libaudiofile1 libavcodec53 libavformat53 libavutil51  
  libcdparanoia0 libdca0 libdirac-encoder0 libdvdnav4 libdvdread4 libenca0  
  libesd0 libfaad2 libfribidi0 libgpm2 libgsm1 libjack-jackd2-0 liblircclient0  
  liblzo2-2 libmp3lame0 libmpeg2-4 libopenal-data libopenal1 libpostproc52  
  libschroedinger-1.0-0 libspeex1 libswscale2 libtheora0 libva1 libvpx1  
  libx264-123 libxvidcore4 libxvmc1  
Suggested packages:  
  libdvdcss2 pulseaudio-esound-compat gpm jackd2 lirc libportaudio2  
  libroar-compat speex mplayer-doc netselect fping  
The following NEW packages will be installed:  
  esound-common libaa1 libaudiofile1 libavcodec53 libavformat53 libavutil51  
  libcdparanoia0 libdca0 libdirac-encoder0 libdvdnav4 libdvdread4 libenca0  
  libesd0 libfaad2 libfribidi0 libgpm2 libgsm1 libjack-jackd2-0 liblircclient0  
  liblzo2-2 libmp3lame0 libmpeg2-4 libopenal-data libopenal1 libpostproc52  
  libschroedinger-1.0-0 libspeex1 libswscale2 libtheora0 libva1 libvpx1  
  libx264-123 libxvidcore4 libxvmc1 mplayer  
0 upgraded, 35 newly installed, 0 to remove and 52 not upgraded.  
Need to get 9,296 kB of archives.  
After this operation, 20.6 MB of additional disk space will be used.  
Do you want to continue [Y/n]? █
```

OK now you just have to run:

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

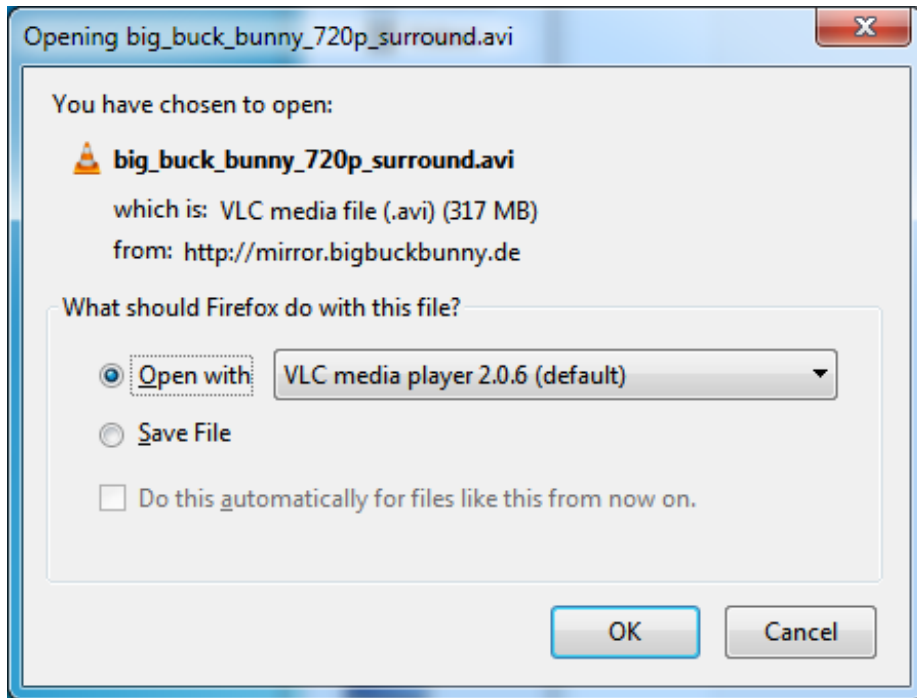
If your video is not sized for 320 wide, you may need to add `a-zoom` after `-framedrop` so that it will resize - note that this is quite taxing for the Pi, so it may result in a choppy or mis-synced video!



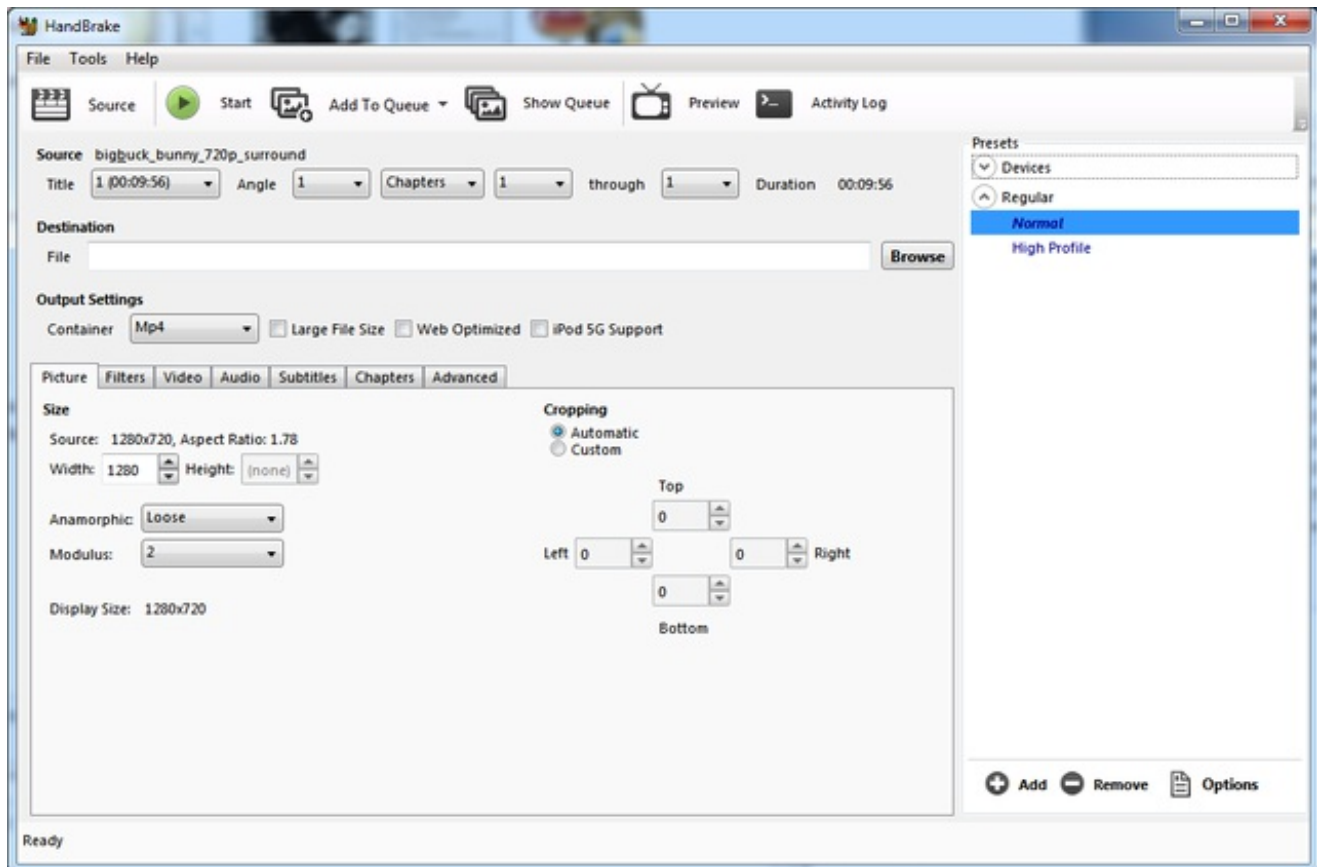
## Converting/Resizing Videos

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) (<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.

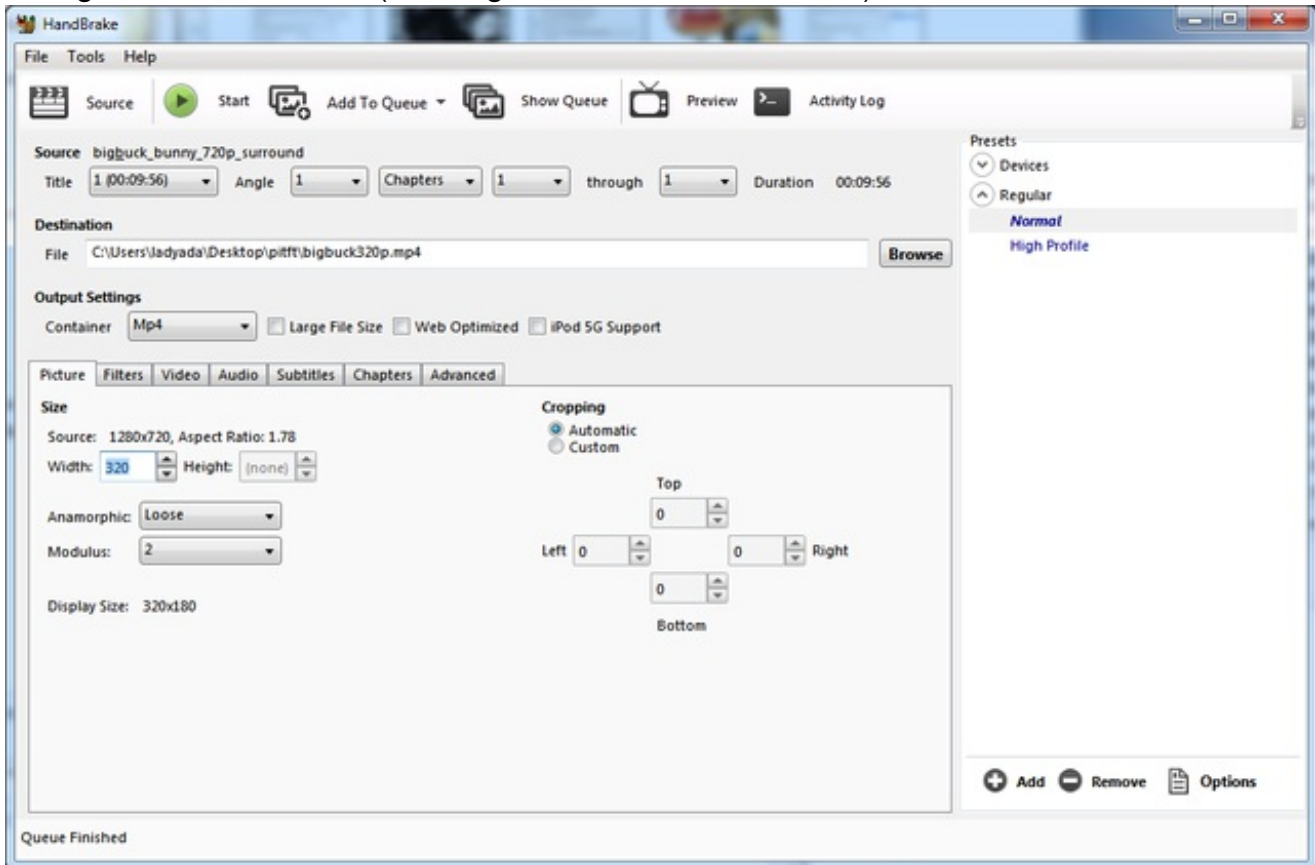


To do the conversion itself, we suggest [HandBrake](http://adafru.it/cXT) (<http://adafru.it/cXT>) which works great and is open source so it runs on all operating systems! Download and install from the link. Then run the installed application and open up the AVI file from before. The app will pre-fill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

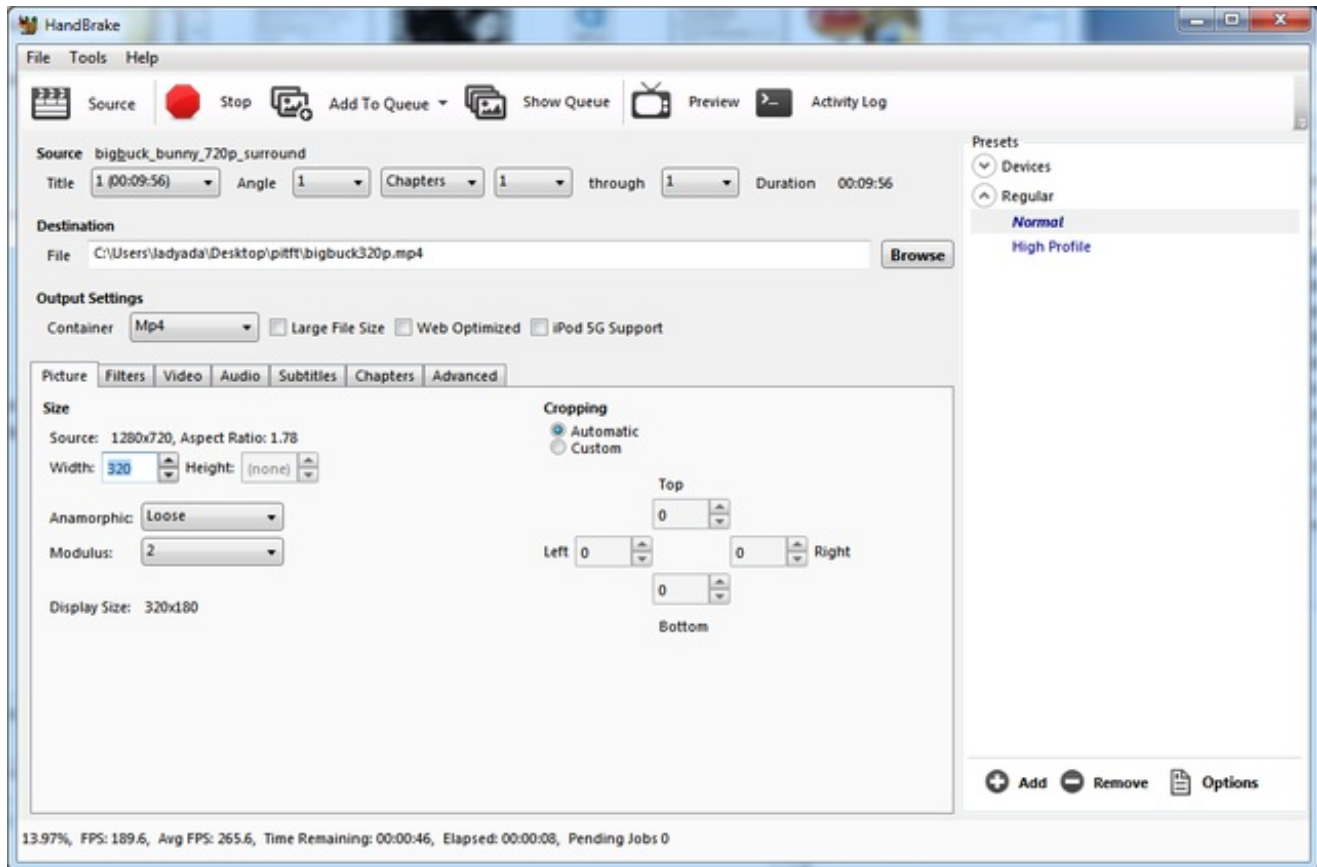


Under **Destination** click **Browse...** to select a new MP4 file to save. Then under **Picture**

change the **Width** to 320 (the height will be auto-calculated)



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

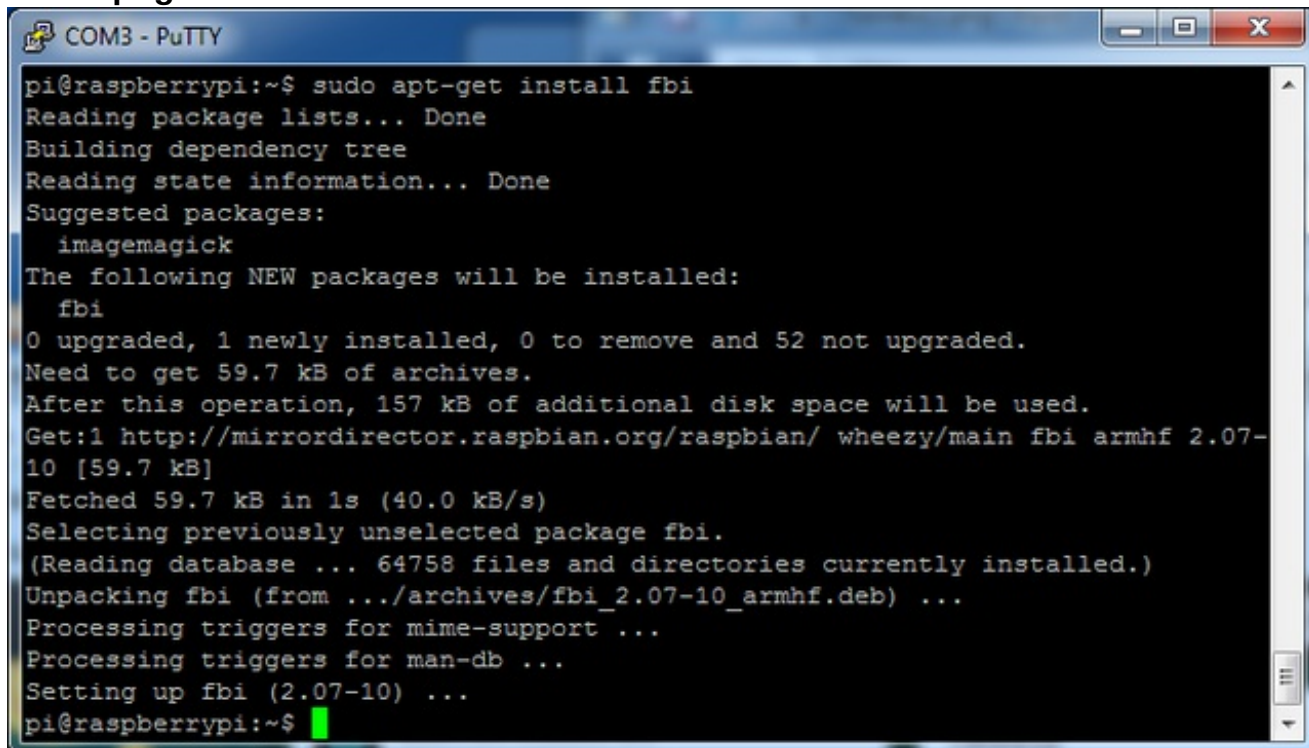


That's it! You now have a smaller file. Don't forget to play it on your computer to make sure it plays right before copying it to your Pi

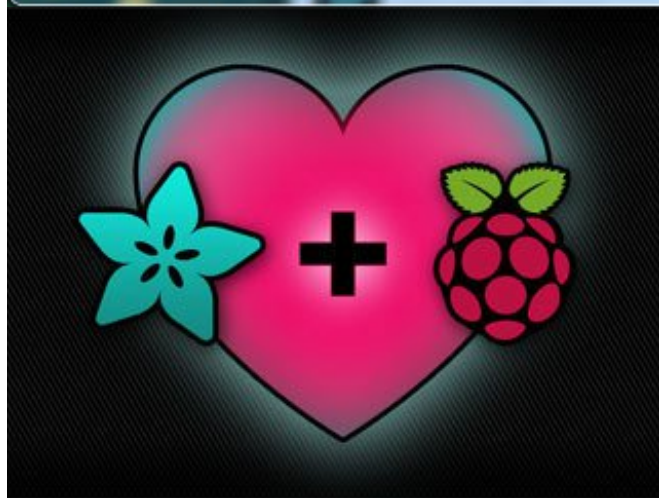
# Displaying Images

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



```
COM3 - PuTTY
pi@raspberrypi:~$ sudo apt-get install fbi
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  imagemagick
The following NEW packages will be installed:
  fbi
0 upgraded, 1 newly installed, 0 to remove and 52 not upgraded.
Need to get 59.7 kB of archives.
After this operation, 157 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main fbi armhf 2.07-10 [59.7 kB]
Fetched 59.7 kB in 1s (40.0 kB/s)
Selecting previously unselected package fbi.
(Reading database ... 64758 files and directories currently installed.)
Unpacking fbi (from ../archives/fbi_2.07-10_armhf.deb) ...
Processing triggers for mime-support ...
Processing triggers for man-db ...
Setting up fbi (2.07-10) ...
pi@raspberrypi:~$
```



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)
```

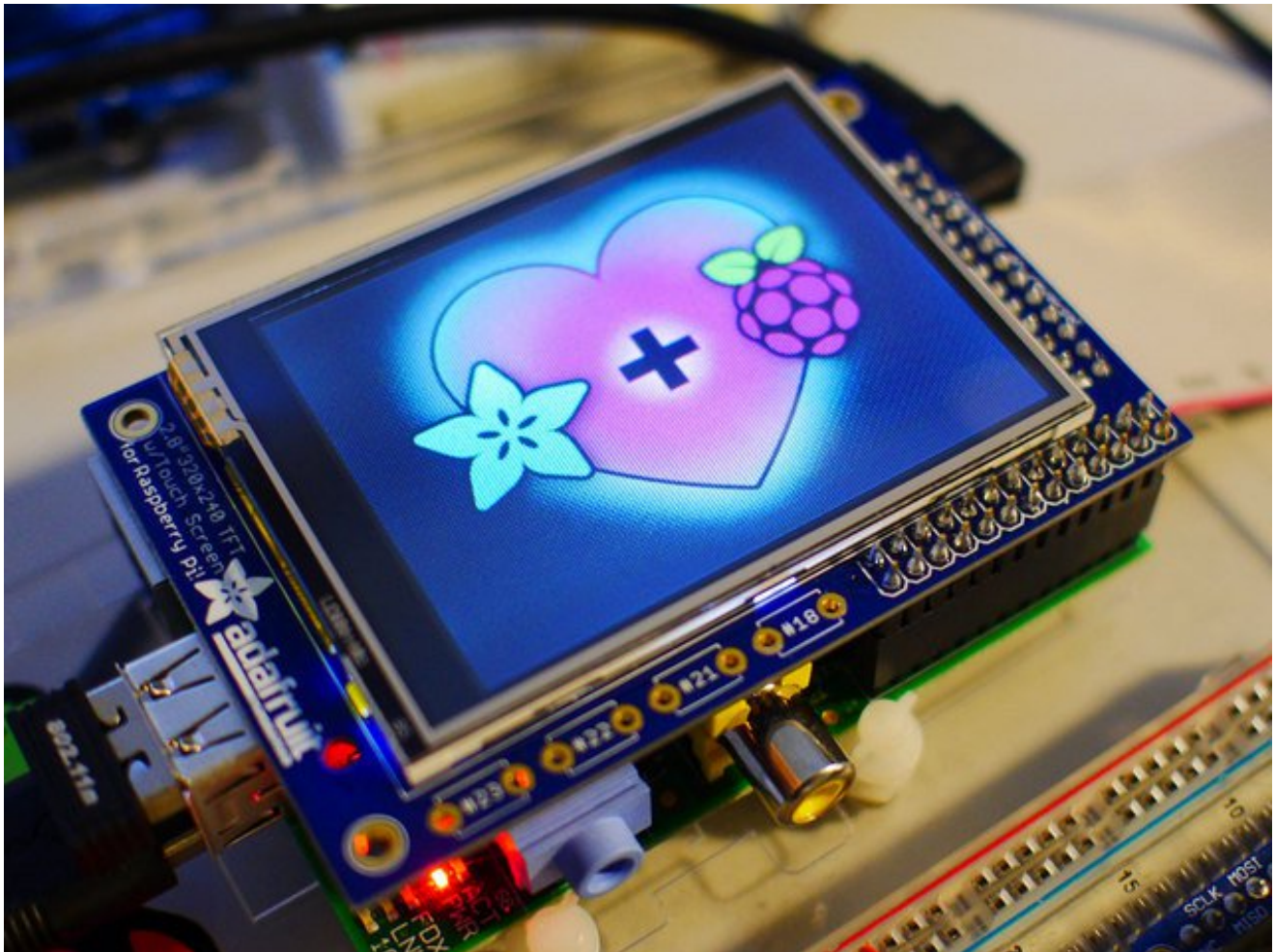
For 320x240 PiTFTs (2.2", 2.4", 2.8" or 3.2") view it with

```
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!





# 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/fbc)

<http://adafru.it/fbc>





# Backlight Control

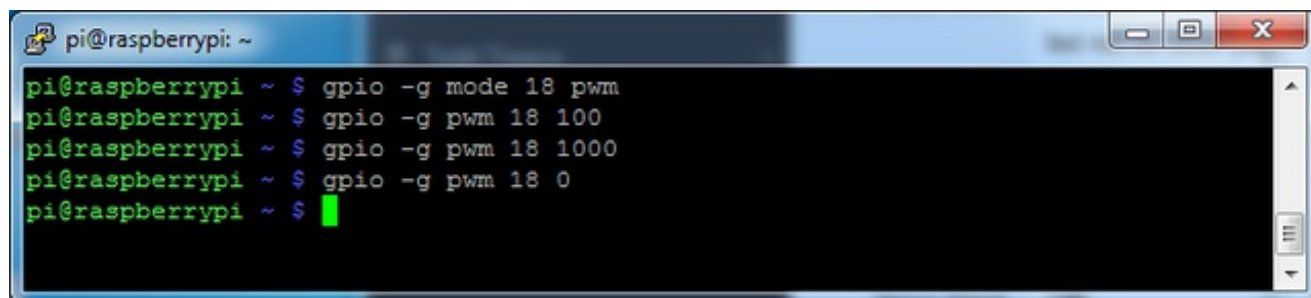
Unlike the resistive 2.8" PiTFT, this little PiTFT does not have a resistive touch controller chip that we can take advantage of as an extra backlight control pin. Instead, you can set use GPIO #18 as an on/off or PWM control.

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
```



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

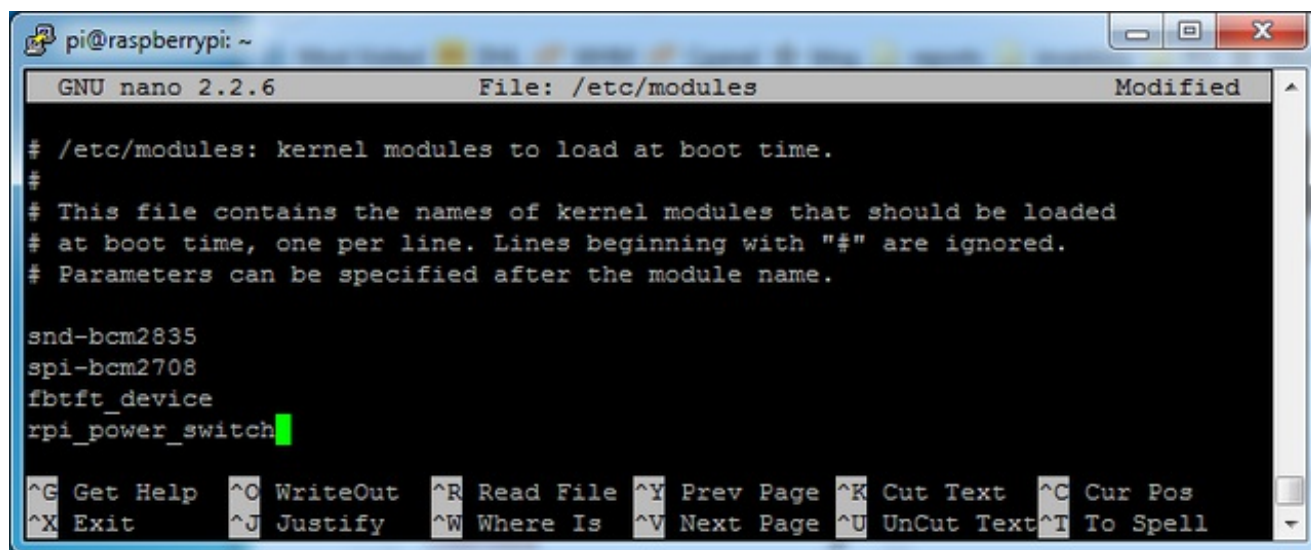
# Extras!

## Tactile switch as power button

Its 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, lets 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



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 its 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**.

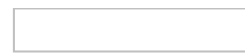
The step above will create a configuration file which tells X windows that it should use the PiTFT framebuffer (located at `/dev/fb1`) by default when it runs.

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



# HELP! (FAQ)

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

Did you do an `apt-get upgrade` or `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-bootloader=1.20150528-1
```

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

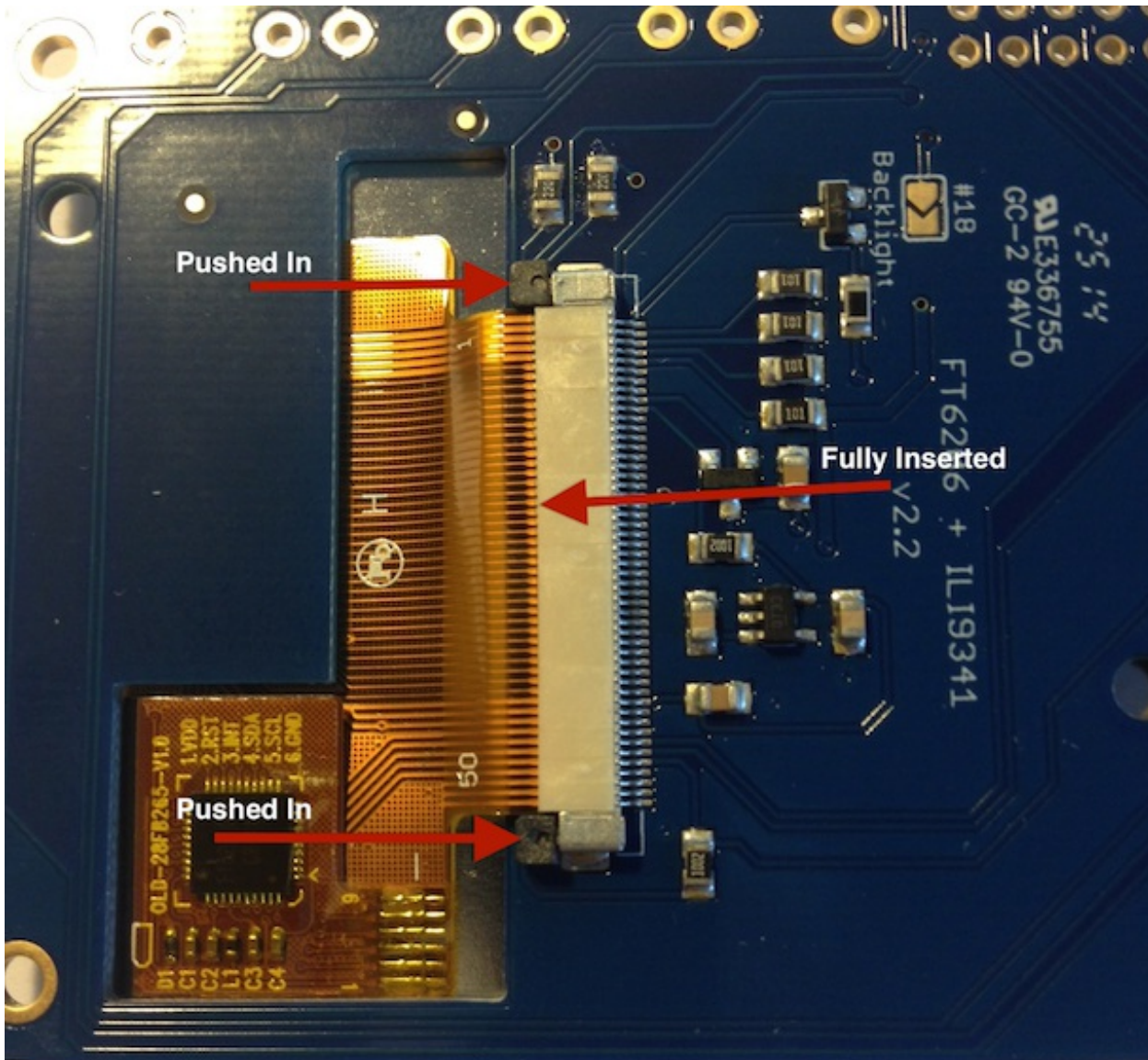
```
sudo apt-get install --reinstall raspberrypi-bootloader=1.20150528-1
```

...you can [check here \(http://adafru.it/eUK\)](http://adafru.it/eUK) and substitute the most recent version you see in the `=1.20150528-1` 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 trying 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 \(http://adafru.it/fbe\)](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) (<http://adafru.it/diV>) [b](http://adafru.it/diV) (<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

```
COM3 - PuTTY
pi@raspberrypi:~$ wget http://fbgrab.monells.se/fbgrab-1.2.tar.gz
--2014-04-21 19:26:22-- http://fbgrab.monells.se/fbgrab-1.2.tar.gz
Resolving fbgrab.monells.se (fbgrab.monells.se)... 66.33.214.148
Connecting to fbgrab.monells.se (fbgrab.monells.se)|66.33.214.148|:80... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 12836 (13K) [application/x-tar]
Saving to: `fbgrab-1.2.tar.gz'

100%[=====>] 12,836      ---K/s   in 0.03s

2014-04-21 19:26:22 (497 KB/s) - `fbgrab-1.2.tar.gz' saved [12836/12836]

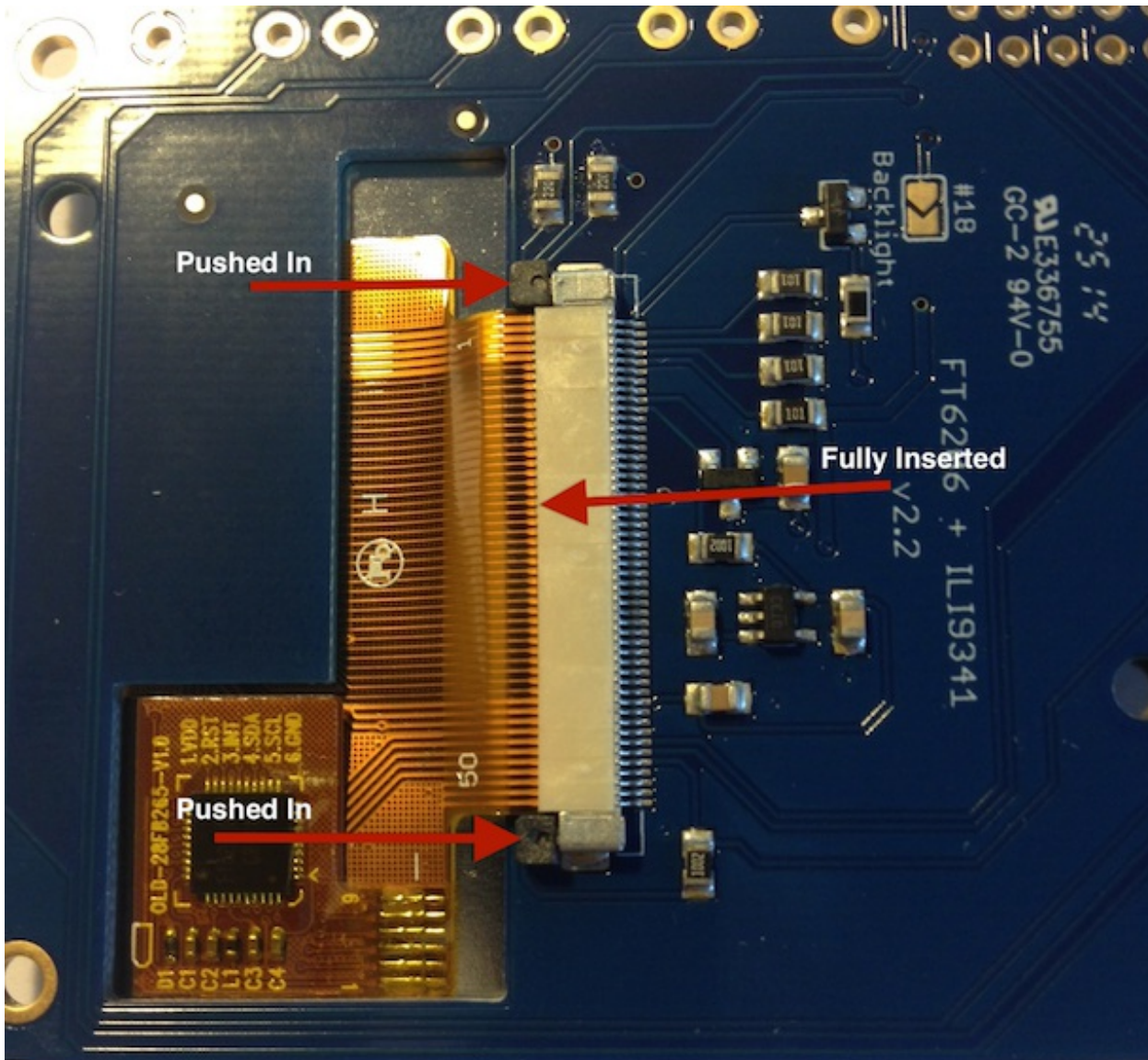
pi@raspberrypi:~$ tar -zxvf fbgrab-1.2.tar.gz
fbgrab/
fbgrab/fbgrab.c
fbgrab/INSTALL
fbgrab/fbgrab.1.man
fbgrab/COPYING
fbgrab/Makefile
pi@raspberrypi:~$ cd fbgrab/
pi@raspberrypi:~/fbgrab$ make
cc -g -Wall  fbgrab.c -lpng -lz -o fbgrab
gzip --best --to-stdout fbgrab.1.man > fbgrab.1.gz
pi@raspberrypi:~/fbgrab$ ./fbgrab
Usage:  ./fbgrab      [-hi] [-{C|c} vt] [-d dev] [-s n] [-z n]
          [-f fromfile -w n -h n -b n] filename.png
pi@raspberrypi:~/fbgrab$ ./fbgrab filemanager.png
Resolution: 320x240 depth 16
Converting image from 16
Now writing PNG file (compression -1)
```

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 \(http://adafru.it/ird\)](http://adafru.it/ird) for more information.

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

Check to make sure that the flat flex cable is fully seated in the connector 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>)

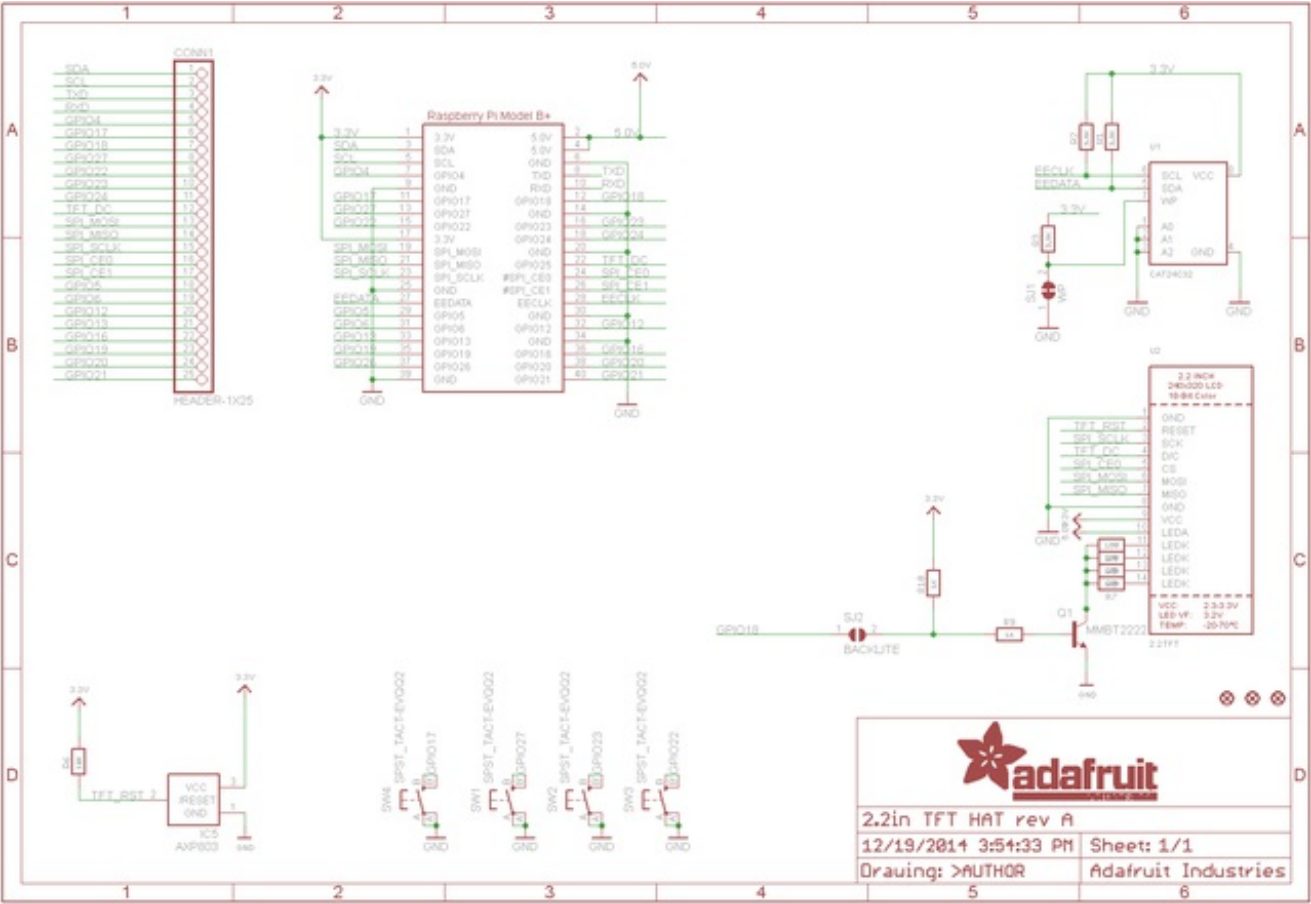


# Downloads

- [The latest kernel fork that adds all the TFT, touchscreen, and other addons is here on github](http://adafru.it/aPa) (<http://adafru.it/aPa>)
- [ILI9340 \(datasheet\)](http://adafru.it/ehr) (<http://adafru.it/ehr>) controller with built in pixel-addressable video RAM buffer
- [Display datasheet](http://adafru.it/ehs) (<http://adafru.it/ehs>)
- [EagleCAD PCB files on GitHub](http://adafru.it/rFP) (<http://adafru.it/rFP>)
- [Fritzing object in the Adafruit Fritzing Library](http://adafru.it/aP3) (<http://adafru.it/aP3>)

# Schematic

click to enlarge



# Fabrication Print

