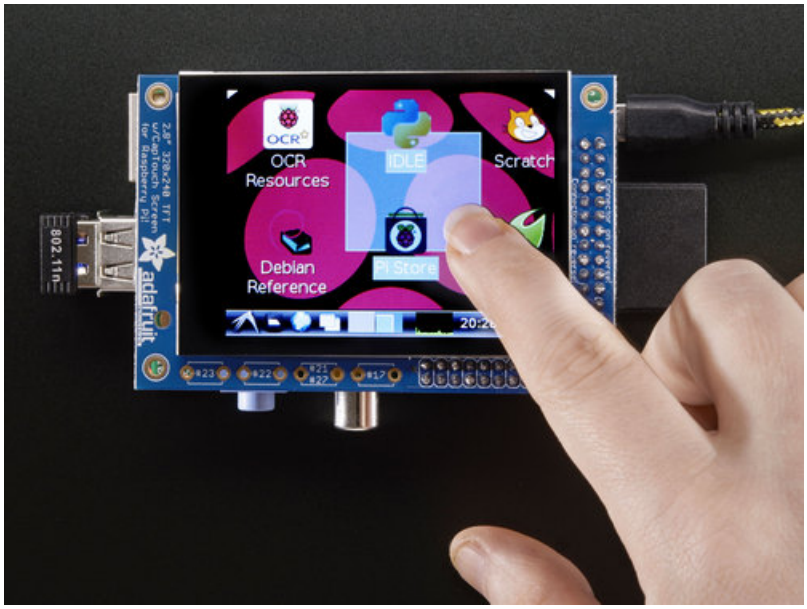
adafruit learning system

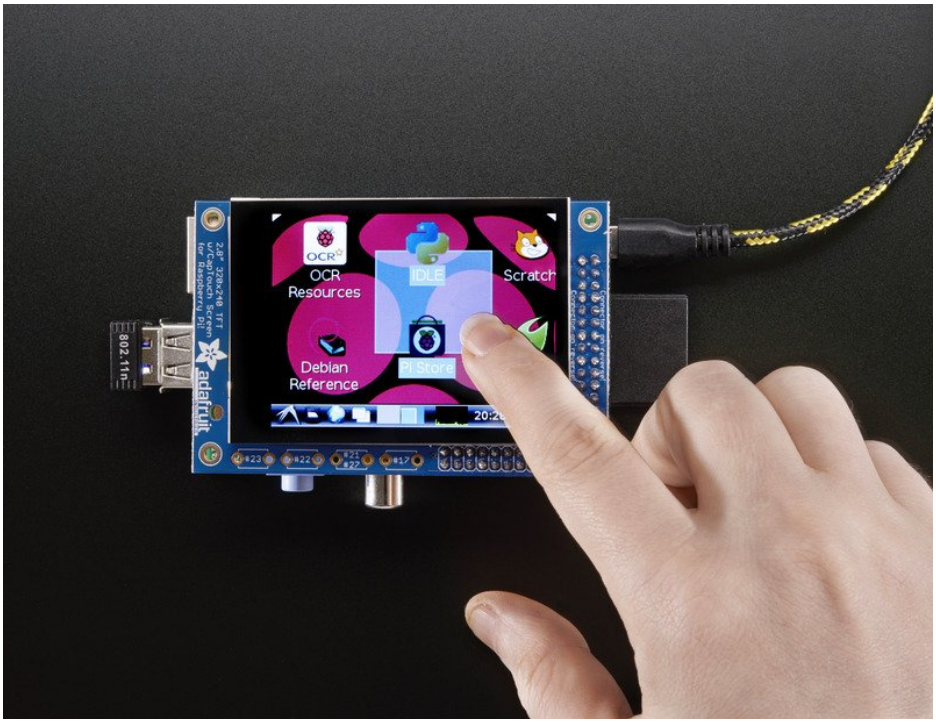
Adafruit 2.8" PiTFT - Capacitive Touch

Created by lady ada



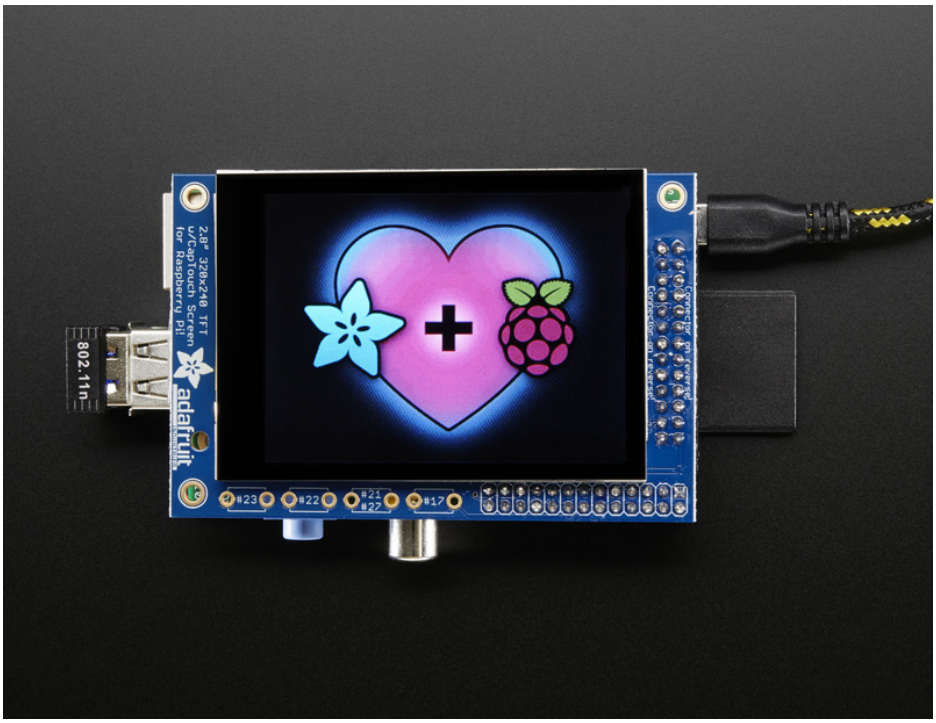
Last updated on 2020-03-09 11:33:41 PM UTC

Overview

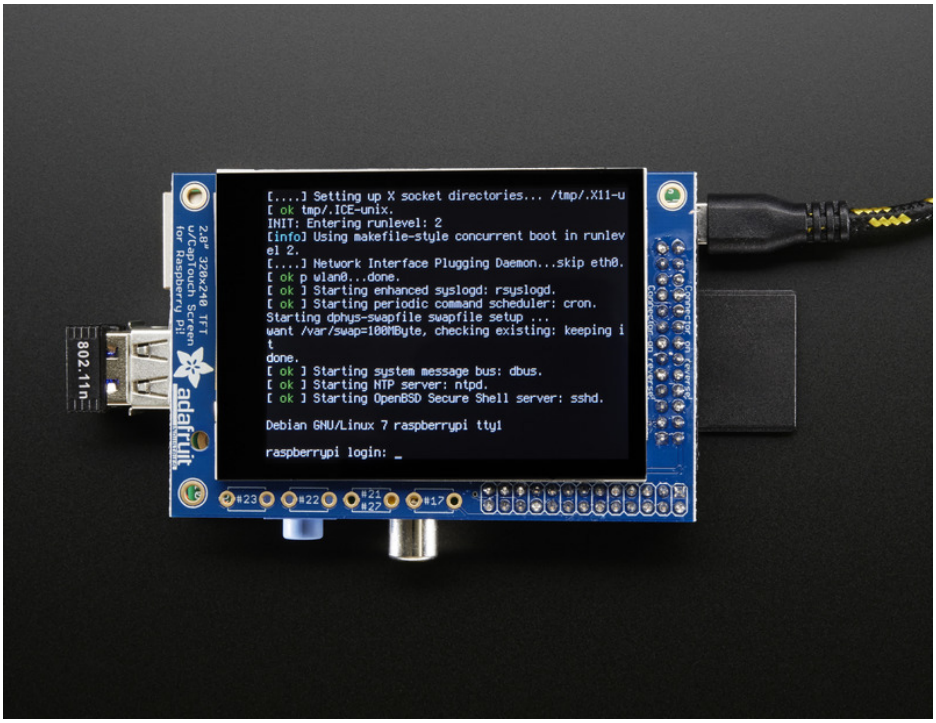


Our best-selling PiTFT just got a fancy upgrade, now we have a version with a **capacitive touchscreen!** That's right, instead of a resistive touchscreen, which requires a fingernail or stylus, you can now use a fingerpad. The screen looks much nicer, with a black bezel and glass overlay.

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!



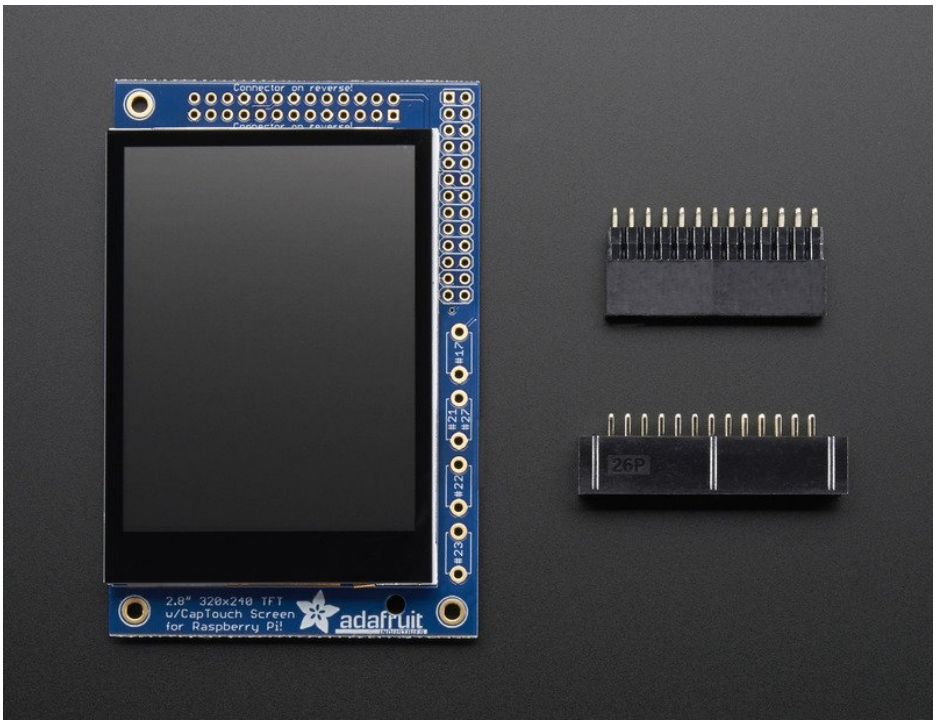
Uses the hardware I2C Pins (SDA & SCL), SPI pins (SCK, MOSI, MISO, CE0) as well as GPIO #25 and #24. All other GPIO are unused. Since we had a tiny bit of space, there's 4 spots for optional slim tactile switches wired to four GPIOs, that you can use if you want to make a basic user interface. For example, 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.

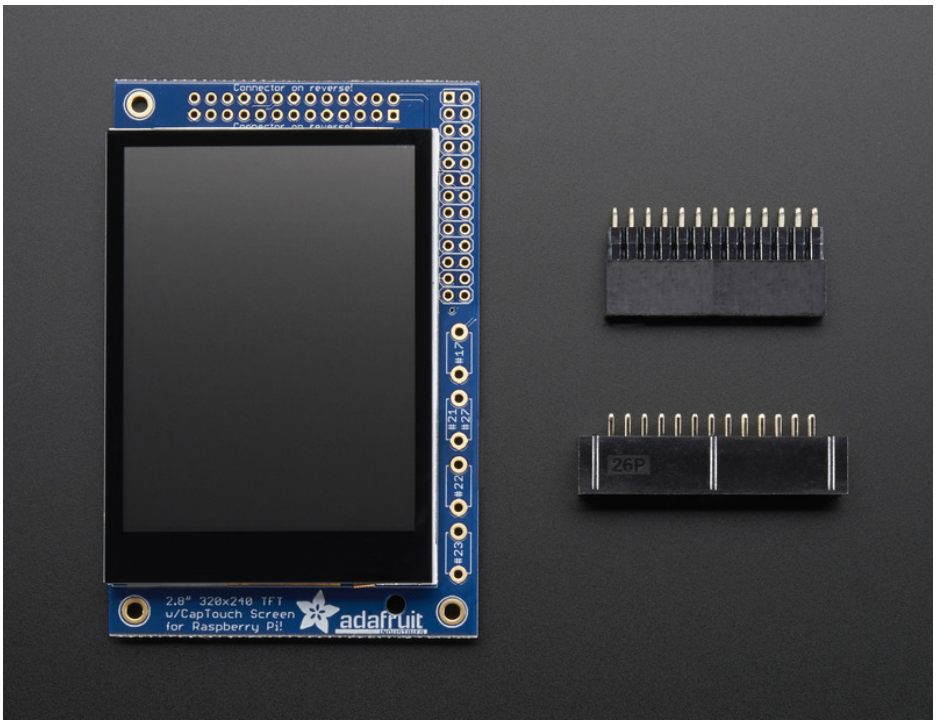
It's designed to fit nicely onto the Pi Model A or B rev 2 but also works perfectly fine with the Model B+ as long as you don't mind the PCB overhangs the USB ports by 5mm, see the photos above. Model B rev 1 have an older layout for the I2C pins and won't be able to use the touch screen

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) 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)



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

https://adafru.it/dDQ

https://adafru.it/dDQ

Easy Install

The PiTFT requires some device tree support and a couple other things to make it a nice stand-alone display. If you just want to get going, check out the following for easy-install instructions!

The same installer is used for all PiTFTs, you will pick and configure the setup during installation!

Install Raspbian on an SD Card

You'll need to start with Raspbian or Raspbian Lite.

The last known for-sure tested-and-working version is March 13, 2018 (https://downloads.raspberrypi.org/raspbian/images/raspbian-2018-03-14/ (https://adafru.it/F2K)) from https://downloads.raspberrypi.org/raspbian/images/ (https://adafru.it/BFU)

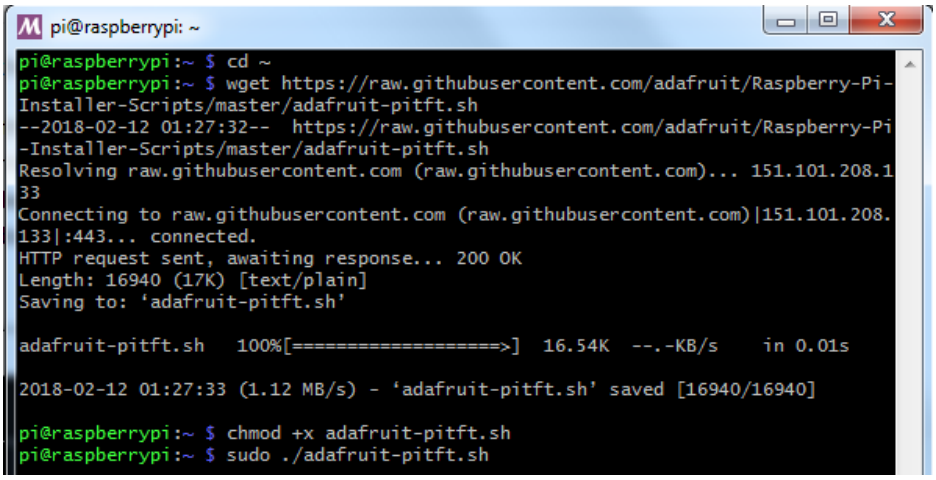
Raspbian does often 'break' stuff when new versions come out so to be safe, if you are having problems try this version!

Installer script

This script will do all the work for you, and install both device tree overlay support as well as configure rotation and any HDMI mirroring. PiTFT no longer needs any custom kernels or modules, so you can continue to update/upgrade your Pi and it will work with the most recent releases.

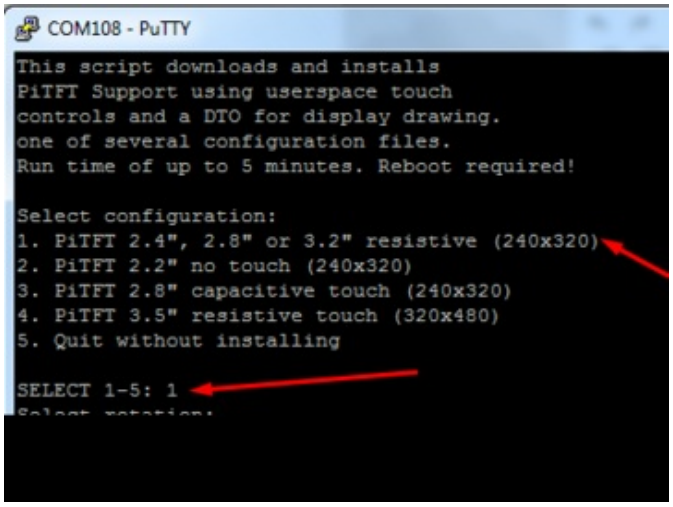
Here's the commands to run. Make sure your Pi has network access, it needs to download the software!

```
cd ~
wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/adafruit-pitft.sh
chmod +x adafruit-pitft.sh
sudo ./adafruit-pitft.sh
```

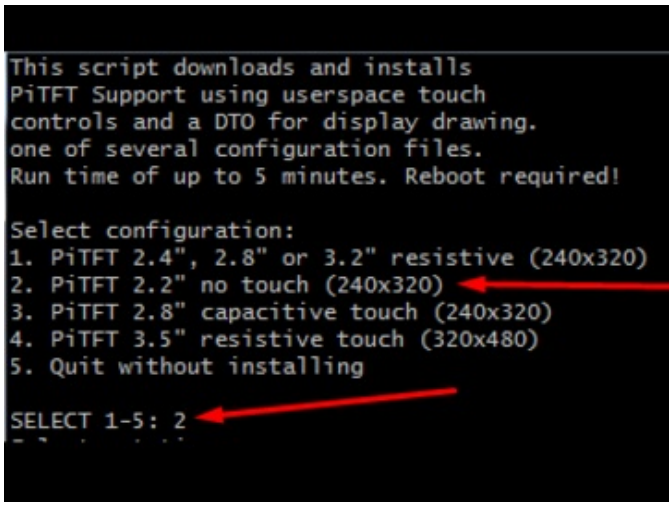


PiTFT Selection

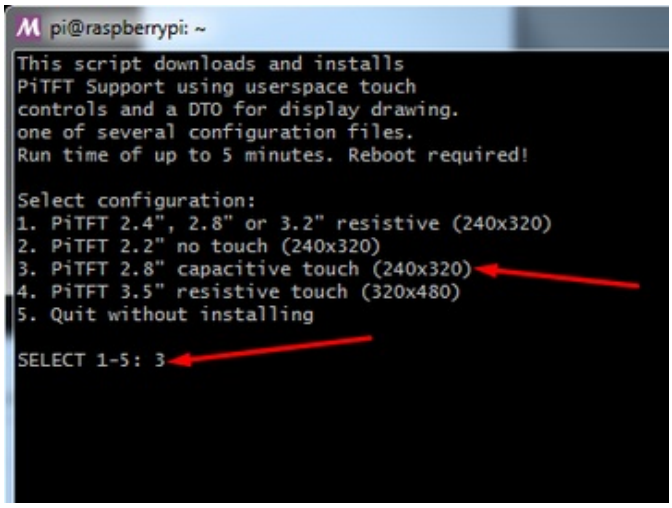
Once you run it you will be presented with menus for configuration.



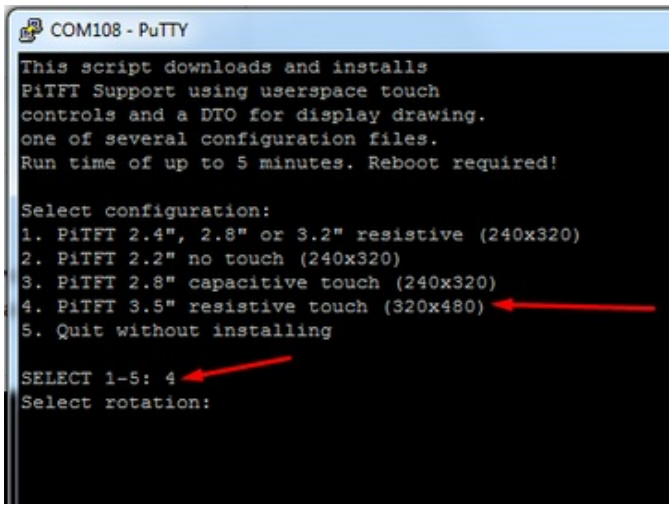
For the 2.4", 2.8" and 3.2" PiTFT with resistive touchscreen overlay select **#1**



For the 2.2" PiTFT select **#2**



For the 2.8" Capacitive PiTFT select **#3**

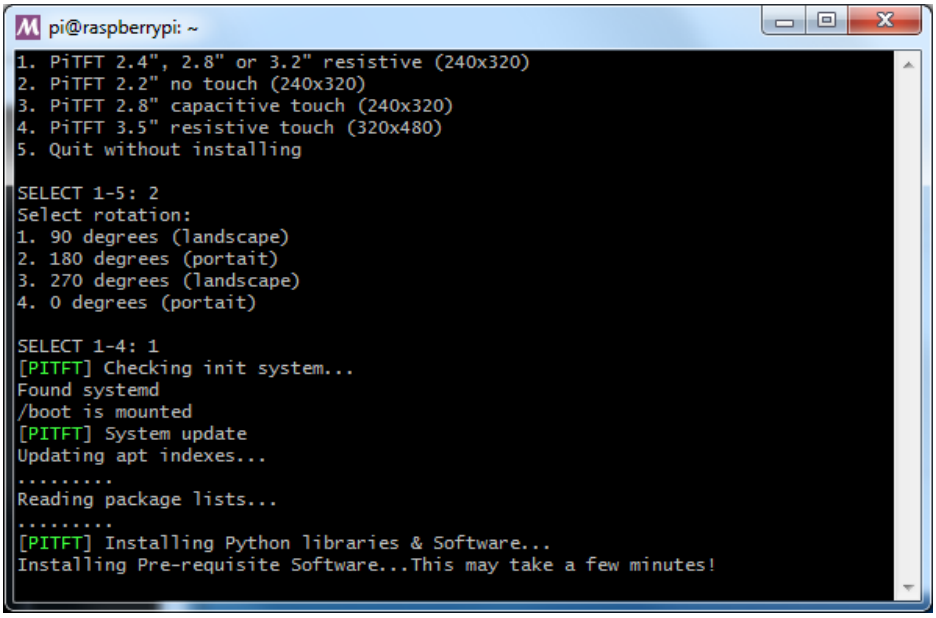


For the 3.5" PiTFT select **#4**

Rotation

Next you will be asked for the rotation you want, don't worry if you're not 100% sure which you want, you can always change this later by re-running the script

It will take a few minutes to install the software and download all the things...



Configuring what shows where

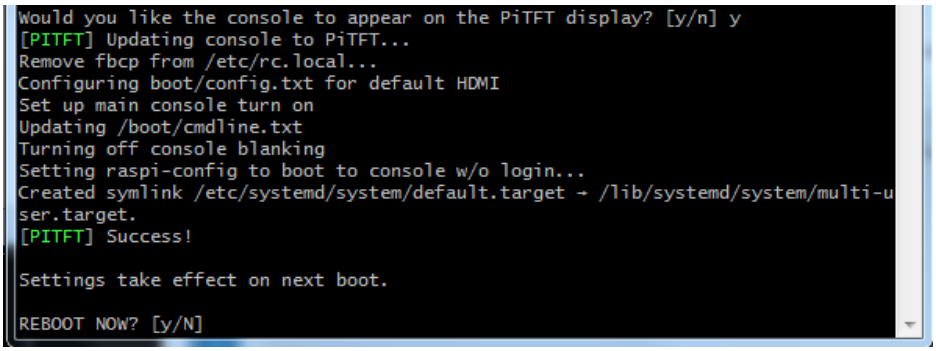
You have a few different ways to set up the PiTFT, we ask **2** questions to figure out what you want

PiTFT as Text Console (best for Raspbian 'Lite')

This is the simplest to set-up type of use. Its great if you have a simple text based or pygame/SDL based interface. If you want the PiTFT to act as a text console you can expect:

- HDMI will be 'deactivated' - nothing appears on the HDMI output but a black screen
- The login prompt appears on the Pi
- The Pi is all text, not a GUI (no PIXEL desktop)
- Keyboard and mouse are used only by the PiTFT interface
- Framebuffer-capable software (such as **fbi** for displaying images, **mplayer** for videos, or pygame software, etc) appear on the PiTFT
- OpenGL accelerated software *will not appear on the PiTFT* (it is unaccelerated framebuffer only)
- But, non-OpenGL-accelerated graphics software is a bit faster than using HDMI mirroring (not tons faster but you're not running **fbcp** which will always make it faster)

If you want that say **Yes** to the question `Would you like the console to appear on the PiTFT display`



Then simply reboot. Once rebooted you will not see anything on HDMI, but the console will appear on the PiTFT. That's it!

PiTFT as HDMI Mirror (Best for Raspbian Full/PIXEL)

This option is the easiest to understand: whatever appears on the HDMI display will be 'mirrored' to the PiTFT. Note that HDMI is much higher resolution so it's not like it turns the PiTFT into a 1080p display. This is great for when you want to run OpenGL-optimized software, PIXEL desktop software, or really anything. The down-side is its a little slower than drawing directly to the framebuffer. You may not notice it but it's worth us mentioning!

- HDMI will be 'activated' but at a lower resolution - you can change this later but it looks best at 320x240 (PiTFT 2.2", 2.4", 2.8" and 3.2") or 480x320 (PiTFT 3.5")
- The login prompt or GUI appears on both HDMI and PiTFT at the same time
- Keyboard and mouse are shared, since the display is mirrored
- All graphics appear on both HDMI and PiTFT, thanks to **fbcp**

If you want that say Yes to the question Would you like the HDMI display to mirror to the PiTFT display?

PiTFT as Raw Framebuffer Device

For advanced users who are comfortable using framebuffer devices, it is possible to have the PiTFT and HDMI graphics be *both* active and display different data.

- HDMI will be active and act like a normal Pi
- The login prompt or GUI (PIXEL) appears on the HDMI
- PiTFT appears black, nothing appears on it
- Keyboard and mouse are used by the HDMI interface but can, in theory, be captured and used to change graphics on PiTFT through programming
- Framebuffer-capable software (such as **fbi** for displaying images, **mplayer** for videos, or pygame software, etc) *can* appear on the PiTFT if you set it up to display to **/dev/fb1**
- OpenGL accelerated software *will never appear on the PiTFT* (it is unaccelerated framebuffer only)

If you want that, say **No** to both of the configuration questions!

You can always change your mind after setting up one of the configurations, depending on your needs! Just re-run the script

Unsupported Full Images

Historically, we provided full 'images' of Raspbian. This worked OK until Raspbian started doing releases every few months. These are no longer supported, and won't even boot on Pi 3B+, so we recommend the script above.

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 (https://adafru.it/aMW)

PiTFT 2.2" Images

- Raspbian Jessie 2016/10/23-based image (https://adafru.it/sbg)
- Raspbian Jessie Lite 2016/10/23-based image (https://adafru.it/sbh)
- Raspbian Jessie 2016/03/25-based image (https://adafru.it/mAe)
- Raspbian Jessie Lite 2016/03/25-based image (https://adafru.it/mAf)
- Raspbian Jessie 2015/09/24-based image (https://adafru.it/iDC)

- Raspbian Wheezy 2015/09/09-based image (https://adafru.it/idt)

PiTFT 2.4"/2.8"/3.2" Resistive Images

- Raspbian Jessie 2016/9/23-based image (https://adafru.it/s7f)
- Raspbian Jessie Lite 2016/9/23-based image (https://adafru.it/s7A)
- Raspbian Jessie 2016/03/25-based image (https://adafru.it/mA9)
- Raspbian Jessie Lite 2016/03/25-based image (https://adafru.it/mAa)
- Raspbian Jessie 2015/09/24-based image (https://adafru.it/iDA)
- Raspbian Wheezy 2015/09/09-based image (https://adafru.it/idJ)
- Raspbian 2014/06/20-based image (https://adafru.it/dSM)
- Raspbian 2014/09/09-based image (https://adafru.it/e12)

PiTFT 2.8" Capacitive

- Raspbian Jessie 2016-09-23-based image (https://adafru.it/saM)
- Raspbian Jessie Lite 2016-09-23-based image (https://adafru.it/saN)
- Raspbian Jessie 2016-03-25-based image (https://adafru.it/mAc)
- Raspbian Jessie Lite 2016-03-25-based image (https://adafru.it/mAd)
- Raspbian Jessie 2015/09/24-based image (https://adafru.it/iDy)
- Raspbian Wheezy 2015/09/24-based image (https://adafru.it/idz)
- Raspbian 2014/09/18-based image (https://adafru.it/e11)
- Raspbian 2014/06/20-based image (https://adafru.it/dSO)
- Raspbian image from 2015/03/03 (https://adafru.it/eUI)

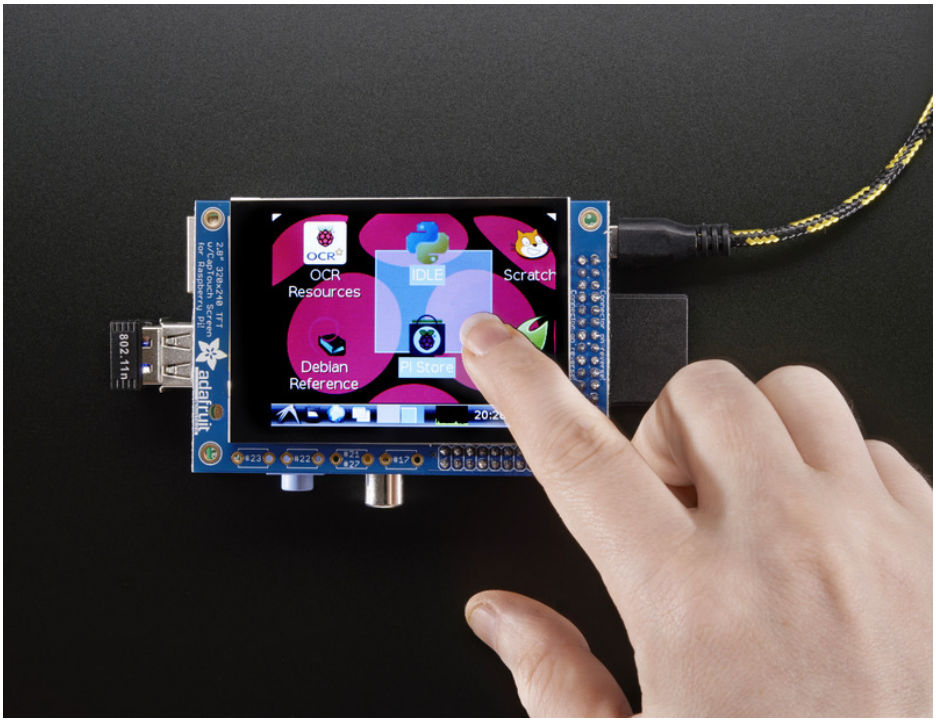
PiTFT 3.5" Images

- Raspbian Jessie 2016/9/23-based image (https://adafru.it/siF)
- Raspbian Jessie Lite 2016/9/23-based image (https://adafru.it/sja)
- Raspbian Jessie 2016/03/25-based image (https://adafru.it/mAb)
- Raspbian Jessie 2016/03/25-based image (https://adafru.it/mAG)
- Raspbian Jessie 2015/09/24-based image (https://adafru.it/iDD)
- Raspbian Wheezy 2015/09/24-based image (https://adafru.it/idy)
- Raspbian 2014/09/09-based image (https://adafru.it/e10)
- Raspbian 2015/03/12 image (https://adafru.it/eUE)

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 capacitive touch driver in Raspbian no longer works with "ts_test" because it doesn't emit 'proper' touchscreen events. It works in LXDE though.



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** or **dmesg | grep EP0110M09**

If you are running **EP0110M09** driver

Run

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

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

```
SUBSYSTEM=="input", ATTRS{name}=="EP0110M09", ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```

If you are running FT6236 driver

Run

```
sudo nano /etc/udev/rules.d/95-ft6236.rules
```

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

```
SUBSYSTEM=="input", ATTRS{name}=="ft6236", ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```

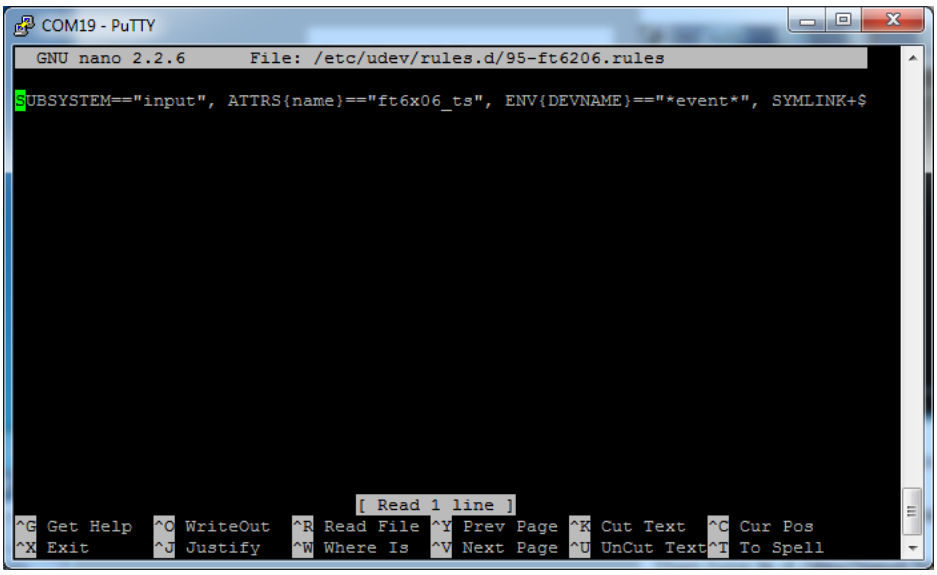
If you are running FT6X06 driver

Run

```
sudo nano /etc/udev/rules.d/95-ft6206.rules
```

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

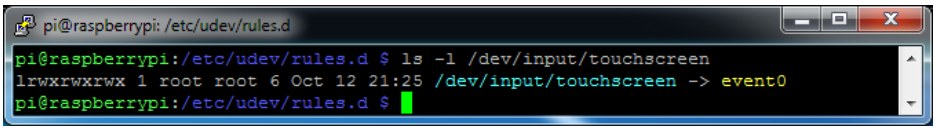
```
SUBSYSTEM=="input", ATTRS{name}=="ft6x06_ts", ENV{DEVNAME}=="*event*", SYMLINK+="input/touchscreen"
```



Reboot the Pi with **sudo reboot**

Then type `ls -l /dev/input/touchscreen`
It should point to **eventX** where X is some number, that number will be different on different setups since other

keyboards/mice/USB devices will take up an event slot

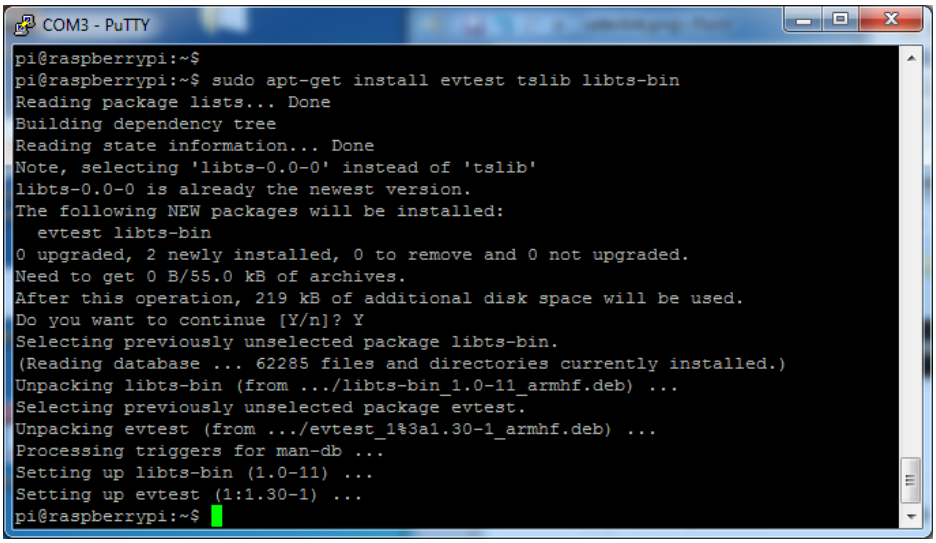


(https://adafru.it/dIX)

Event Testing

Even though capacitive touch screens don't require calibration, there are some useful tools we can use to debug the touchscreen. Install the "event test" and "touchscreen library" packages with

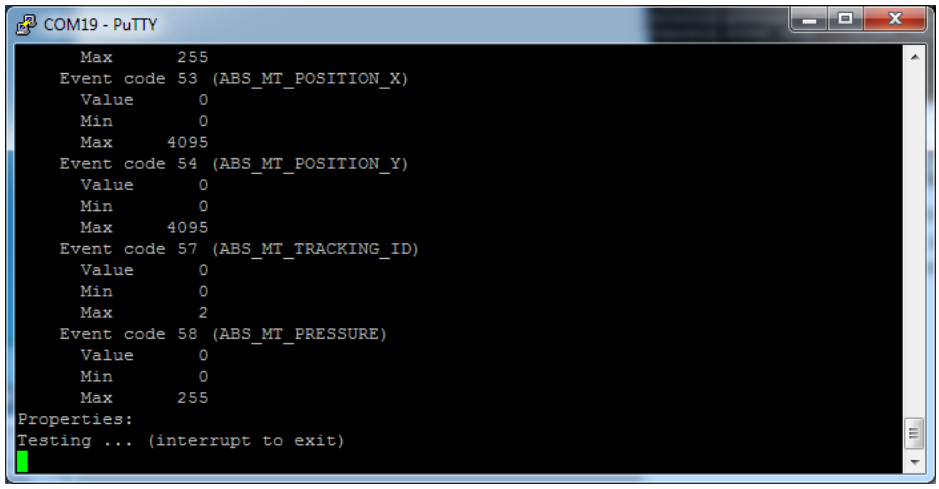
```
sudo apt-get install evtest tslib libts-bin
```

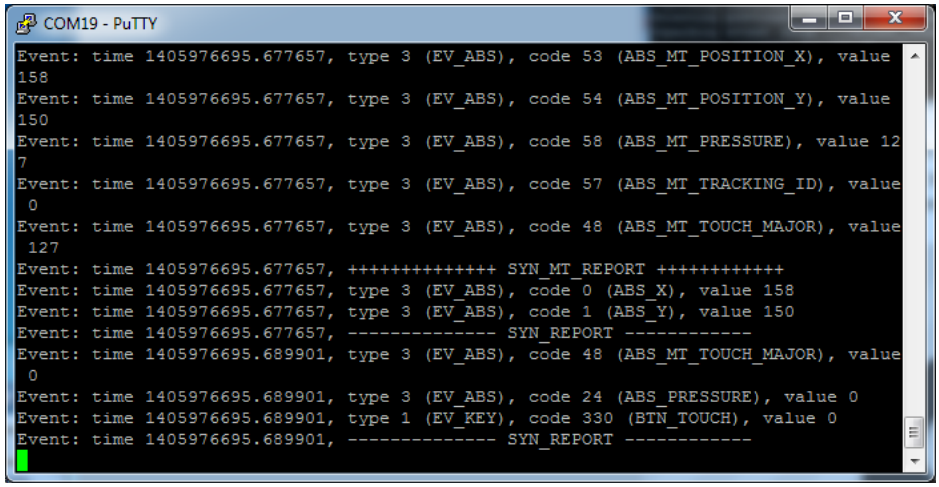


Now you can use some tools such as

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

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





AutoMagic Calibration Script

If you rotate the display you need to recalibrate the touchscreen to work with the new screen orientation. You can manually run the calibration processes in the next section, or you can re-run the installer script and select a new rotation:

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

TSLIB calibration

In order to use TSLIB - basically, the touchscreen without X11 - you'll need to set the calibration for TSLIB in /etc/pointercal

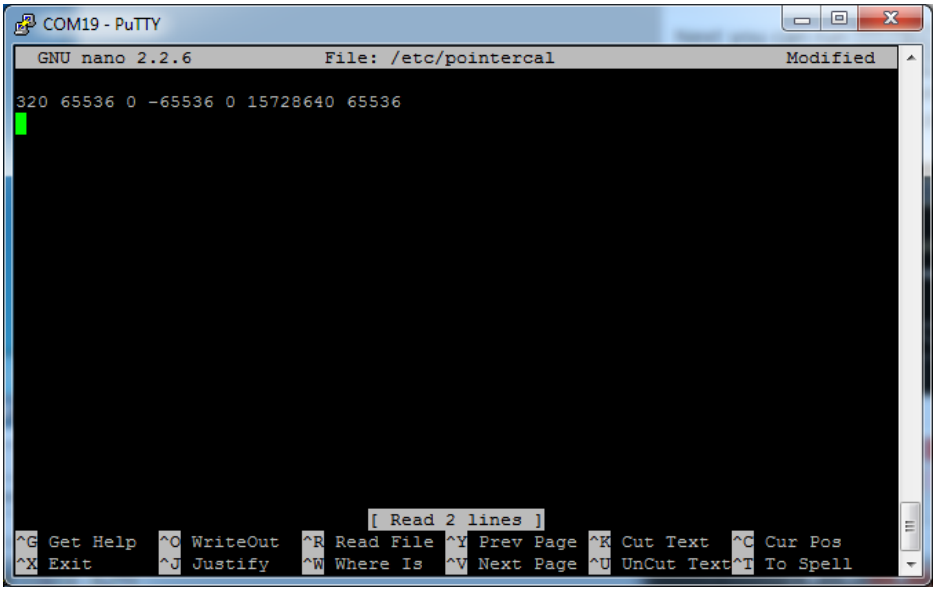
With a resistive touchscreen, you have to calibrate it. Since capacitive touchscreens don't require calibration you can

just input the numbers directly. Run

```
sudo nano /etc/pointercal
```

And enter in the following values (there's a single space between each number) and hit return afterwards. Then save

```
320 65536 0 -65536 0 15728640 65536
```



Next you can run

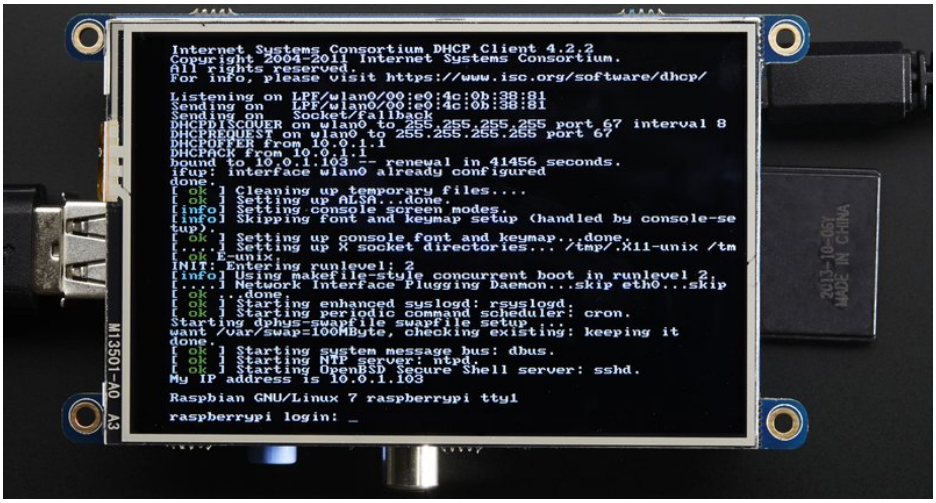
```
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_test
```

X11 Calibration

X11 uses a *different* calibration system than TSLib/PyGame. You can see how to run xtcal here (https://adafru.it/BFX) except use **EP0110M09** as the name of the device

Console Configuration

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



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

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

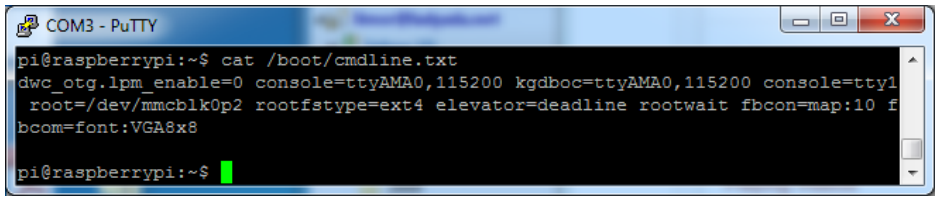
```
sudo nano /boot/cmdline.txt
```

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

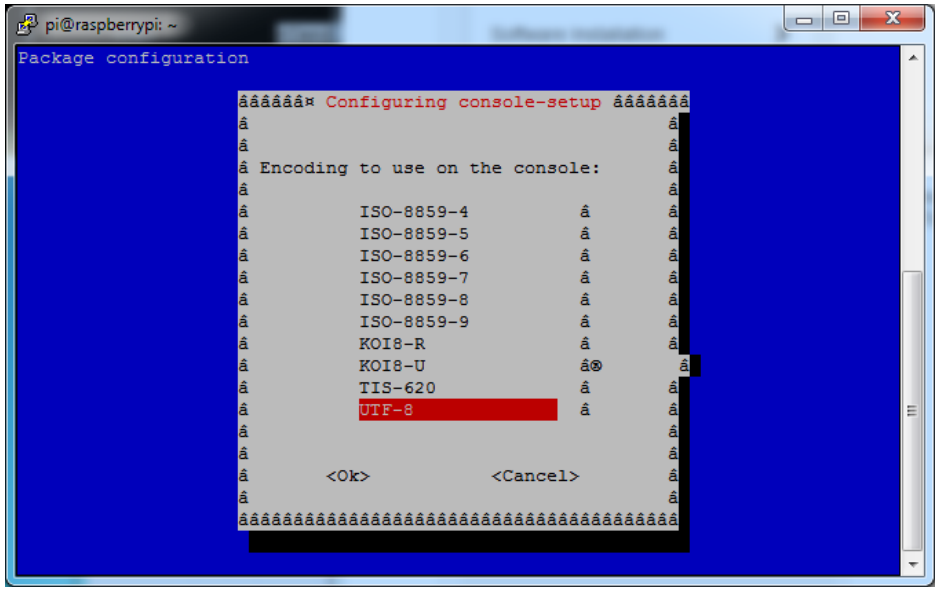
At the end of the line, find the text that says `rootwait` and right after that, enter in: `fbcon=map:10 fbcon=font:VGA8x8` then save the file.

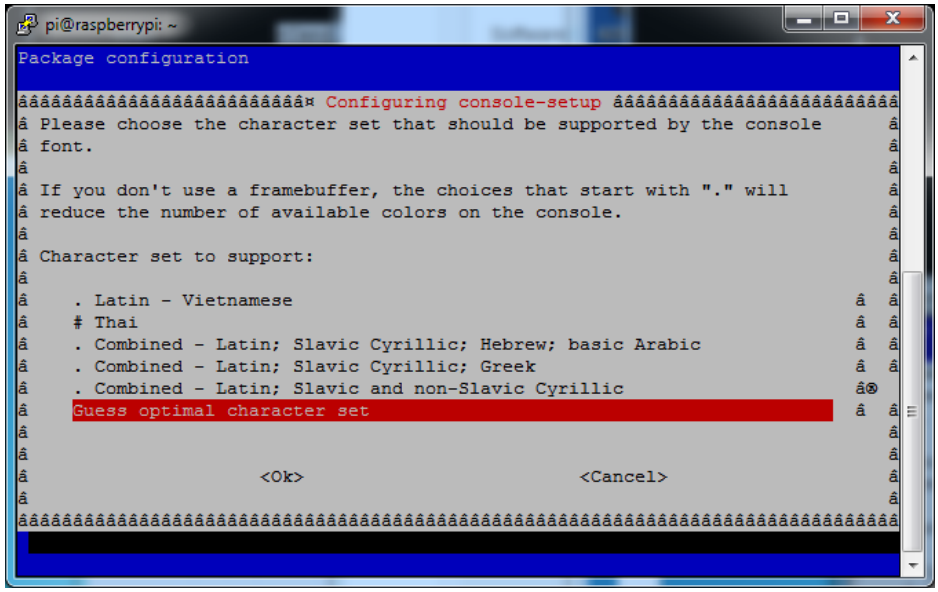
On the next boot, it will bring up the console.

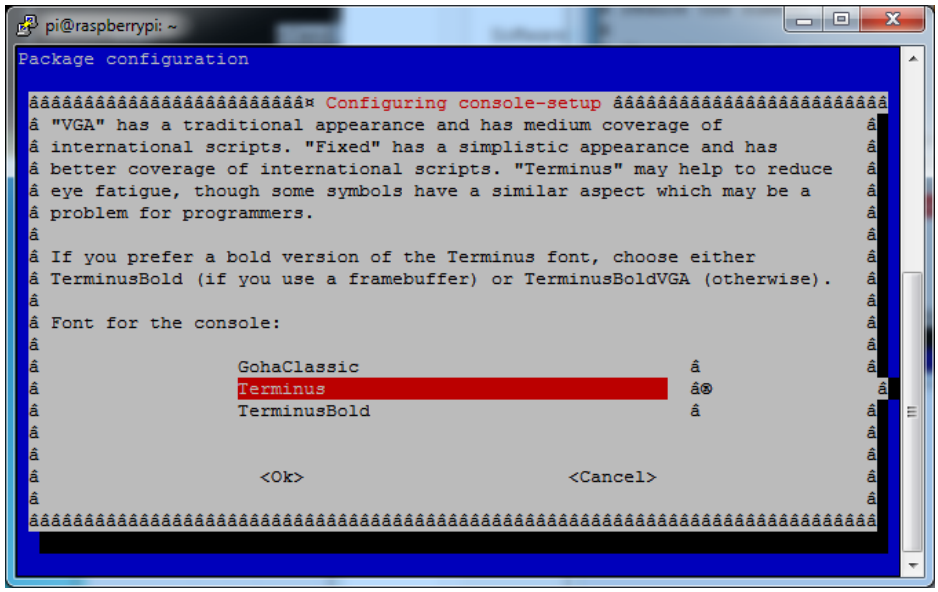
Note that the kernel has to load up the display driver module before it can display anything on it so you won't get the rainbow screen, a NooBs prompt, or a big chunk of the kernel details since the module is loaded fairly late in the boot process.

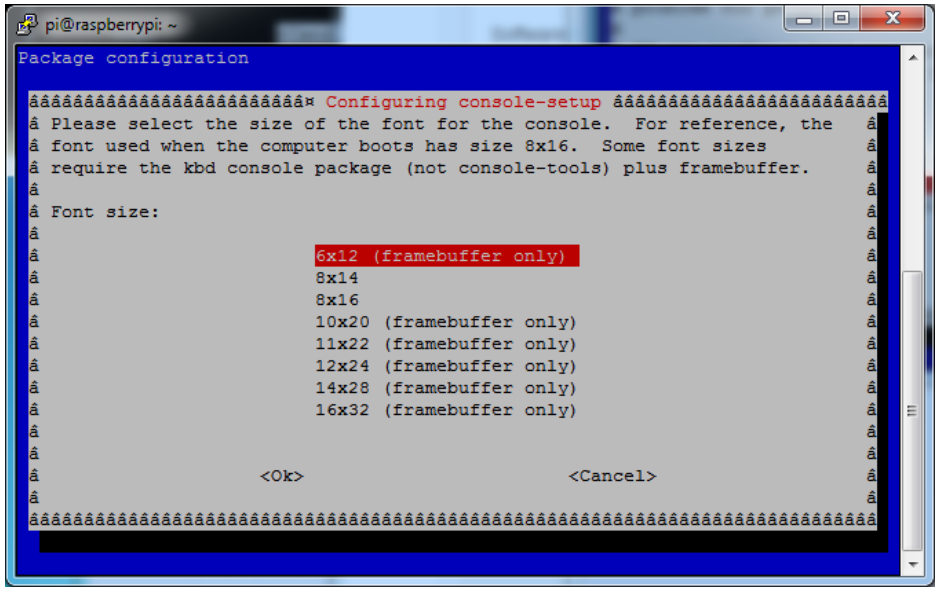


I think the VGA8x8 font is a bit chunky, you probably want 12x6 which is what is shown in the photo above. To change the font, run **sudo dpkg-reconfigure console-setup** and go thru to select Terminus 6x12


pi@raspberrypi: ~
Package configuration
Configuring console-setup
Encoding to use on the console:
ISO-8859-4
ISO-8859-5
ISO-8859-6
ISO-8859-7
ISO-8859-8
ISO-8859-9
KOI8-R
KOI8-U
TIS-620
UTF-8
<Ok>
<Cancel>


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





Turn off Console Blanking

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

Raspbian Jessie

Add the following line to /etc/rc.local

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

on the line before the final `exit 0`

Raspbian Wheezy

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

`BLANK_TIME=30`

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

```
BLANK_TIME=0
```

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 (https://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
```

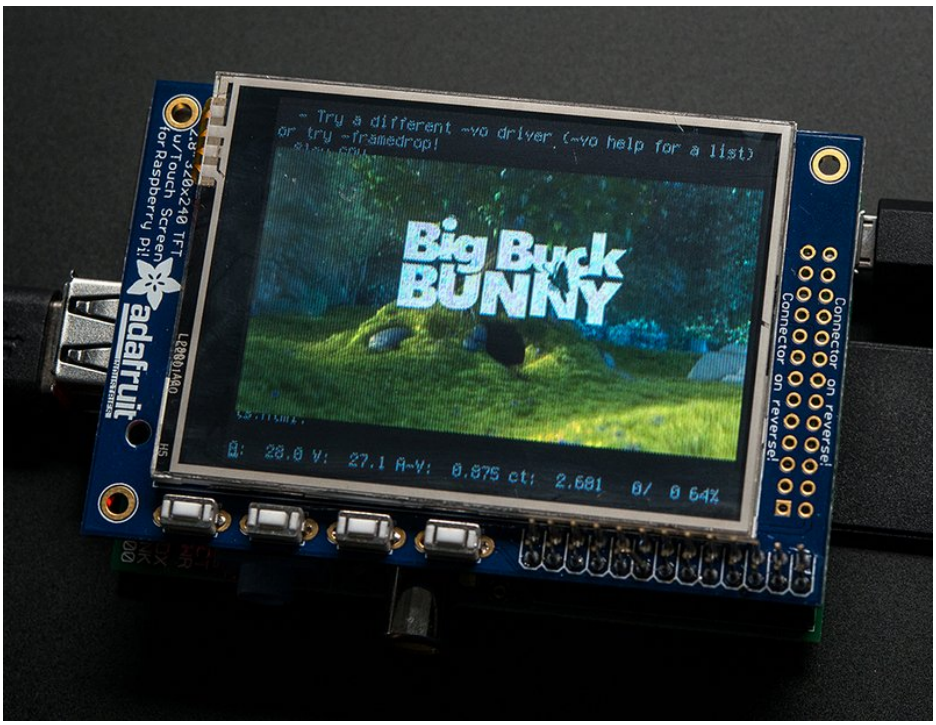
to install it. It may take a few minutes to complete

```
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-compat2 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!

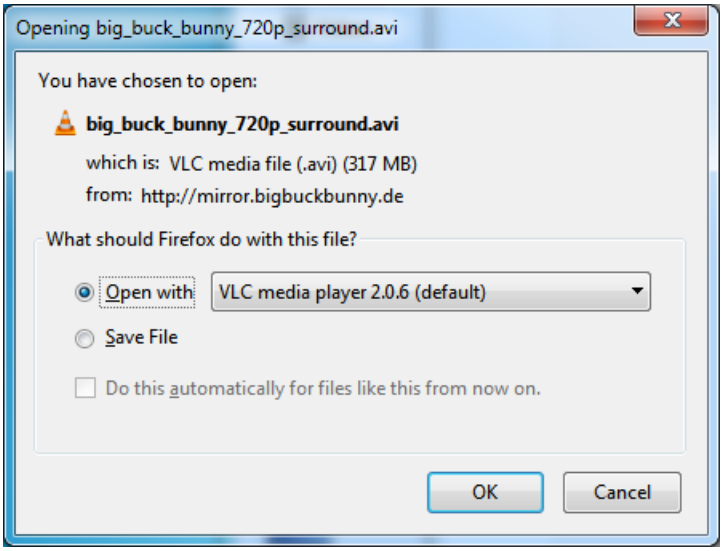


Converting/Resizing Videos

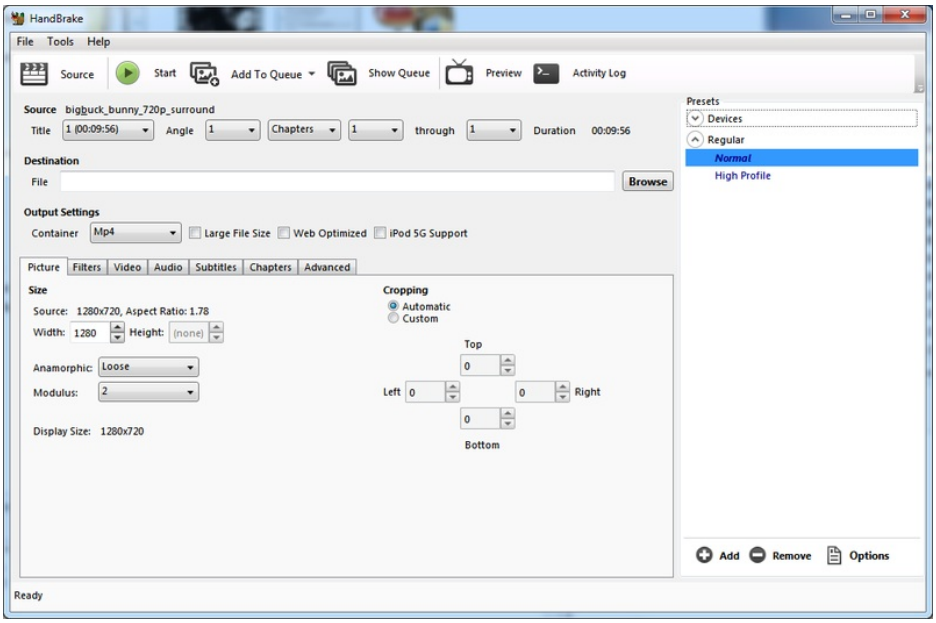
It's possible to play full length videos on the TFT plate, but since the screen is small and the Pi cant use hardware accelleration to play the videos its best to scale them down to 320x240 pixels. This will be easier for the Pi to play and also save you tons of storage space. For this demo, we'll be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

video, which is creative commons and also very funny!

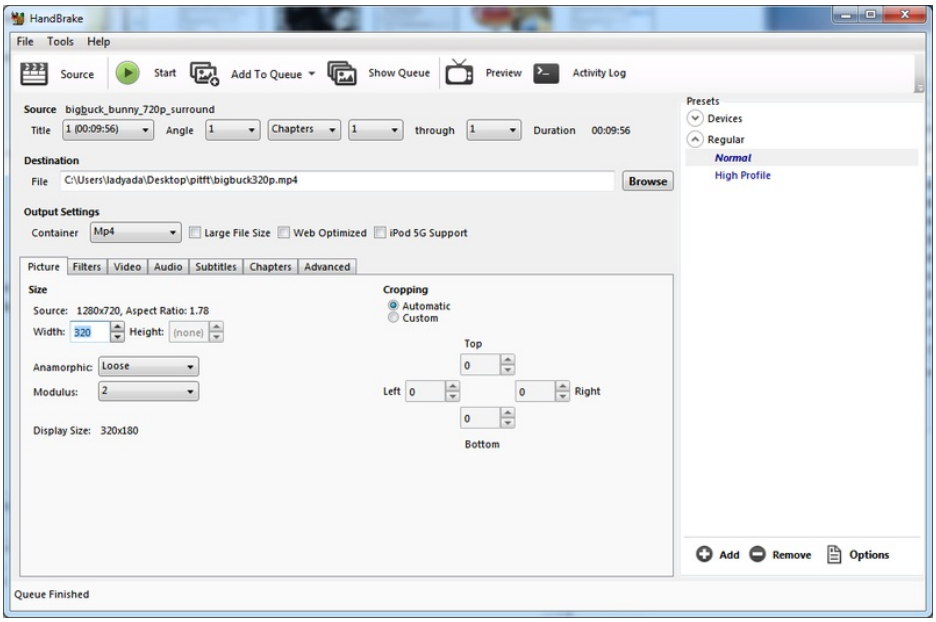
You can download it from the link above, we'll be using the 720p AVI version.



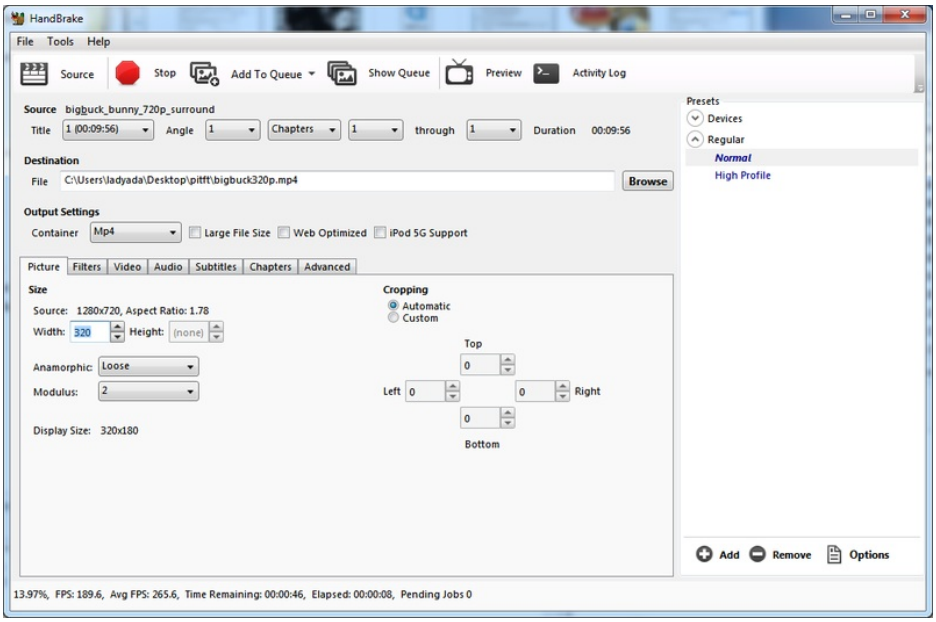
To do the conversion itself, we suggest HandBrake (https://adafru.it/cXT) which works great and is open source so it runs on all operating systems! Download and install from the link. Then run the installed application and open up the AVI file from before. The app will pre-fill a bunch of information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

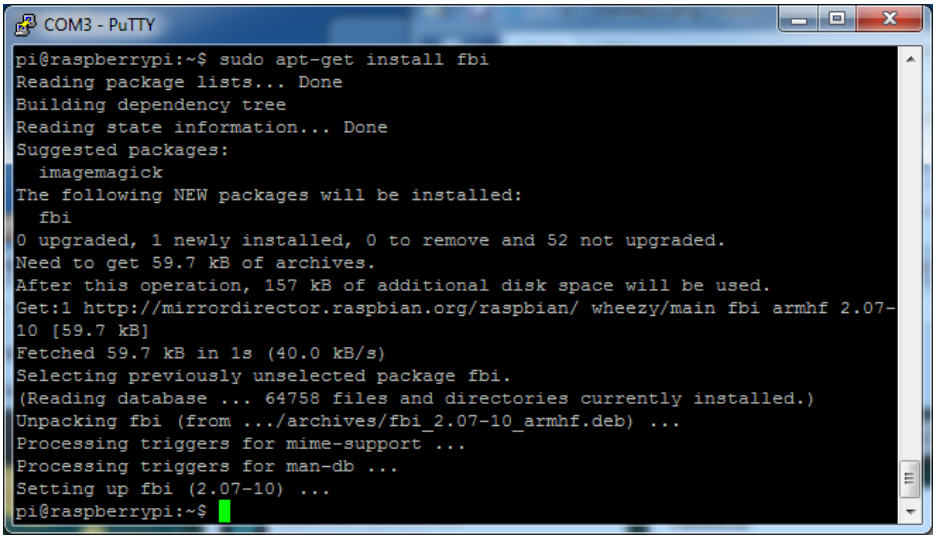


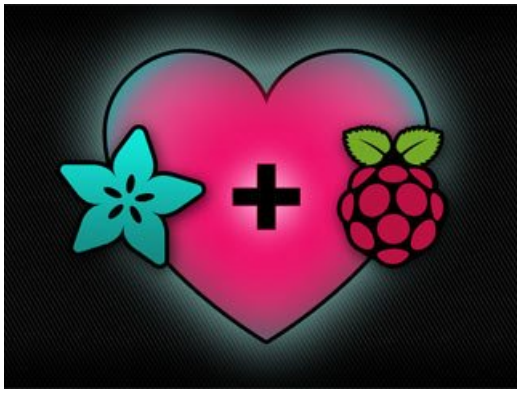
That's it! You now have a smaller file. Don't forget to play it on your computer to make sure it plays right before copying it to your Pi

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





Grab our lovely wallpapers with

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

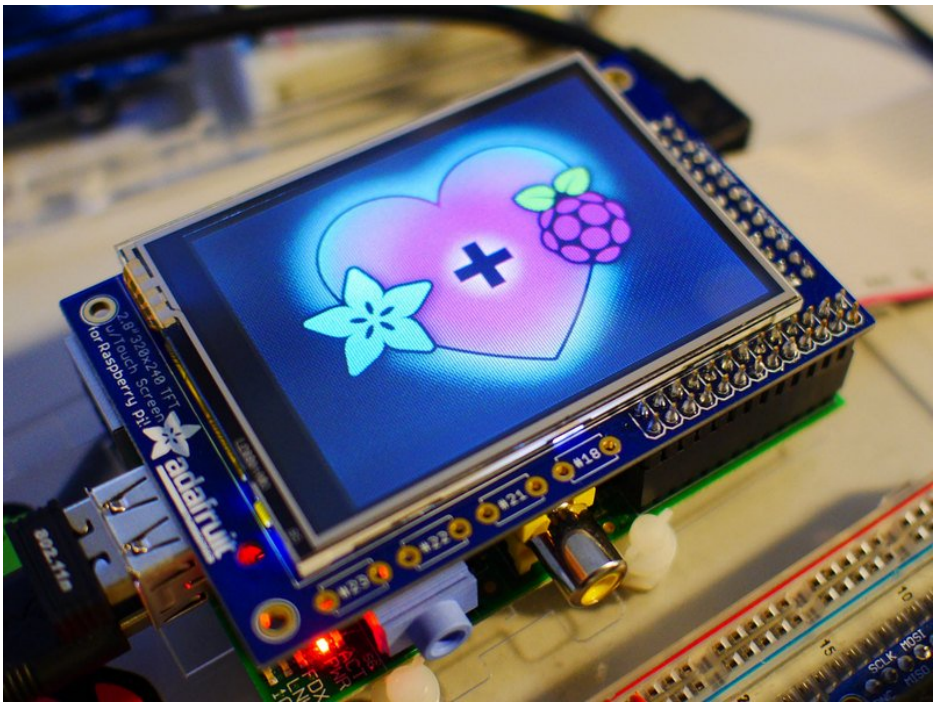
For 320x240 PiTFTs (2.2", 2.4", 2.8" or 3.2") view it with

```
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 (https://adafru.it/cXU).png
```

That's it!


adafruit
2.8"x320x240 TFT
w/Touch Screen
for Raspberry Pi!
#18
#21
#22
#25

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!

https://adafru.it/fbe

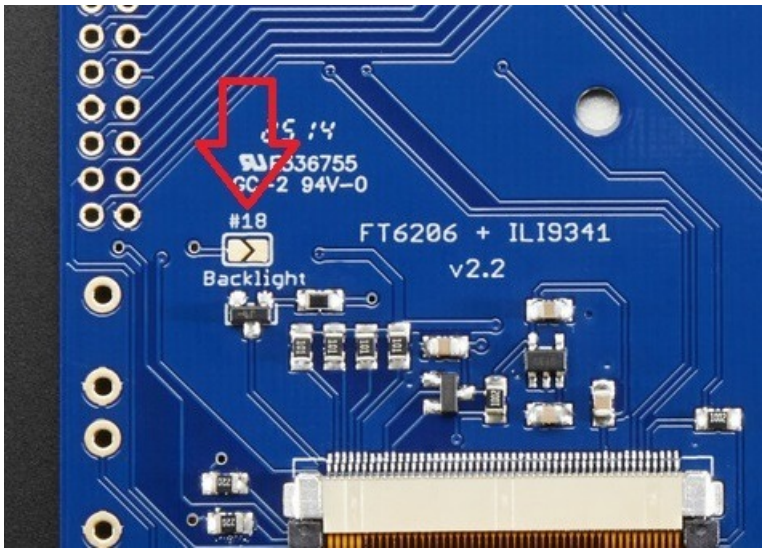
https://adafru.it/fbe

Backlight Control

Unlike the resistive PiTFT, the capacitive version does not have a resistive touch controller chip that we can take advantage of as an extra backlight control pin. Instead, you can set up 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

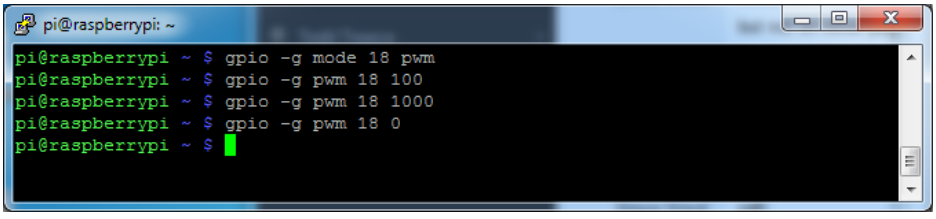
To enable using GPIO #18 as a backlight, solder closed the #18 backlight jumper on the PiTFT capacitive PCB!



OK now you can use the PWM output on GPIO 18. There's python code available for controlling the PWM pin but you can also just use the WiringPi 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
```



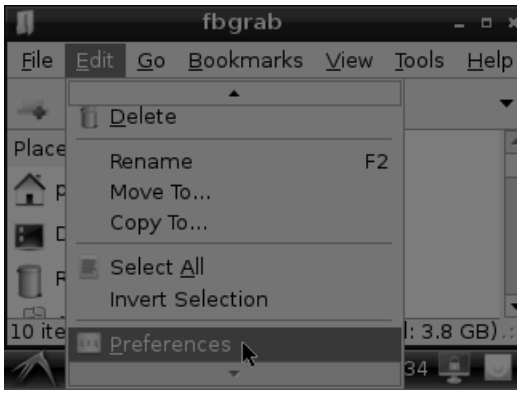
Try other numbers, from 0 (off) to 1023 (all the way on)!

Extras!

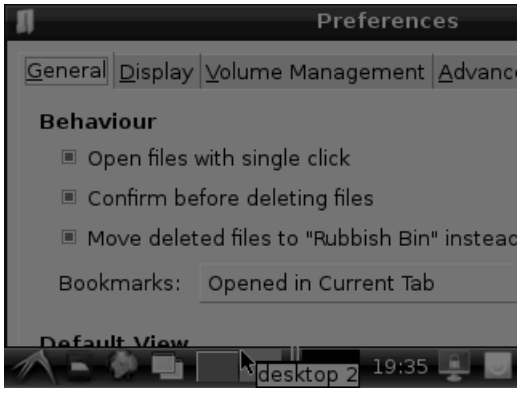
Making it easier to click icons in X

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

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



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



Right-click on a touchscreen

Obviously if you have a touchscreen, it cannot tell what finger you are pressing with. This means that all 'clicks' are left

clicks. But if you want a right-click, you *can* do it.

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

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

So for example your file will look like:

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

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

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

Gesture Input

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

Installation

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

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

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

Now download, compile, and install xstroke by executing:

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

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

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

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

Usage

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

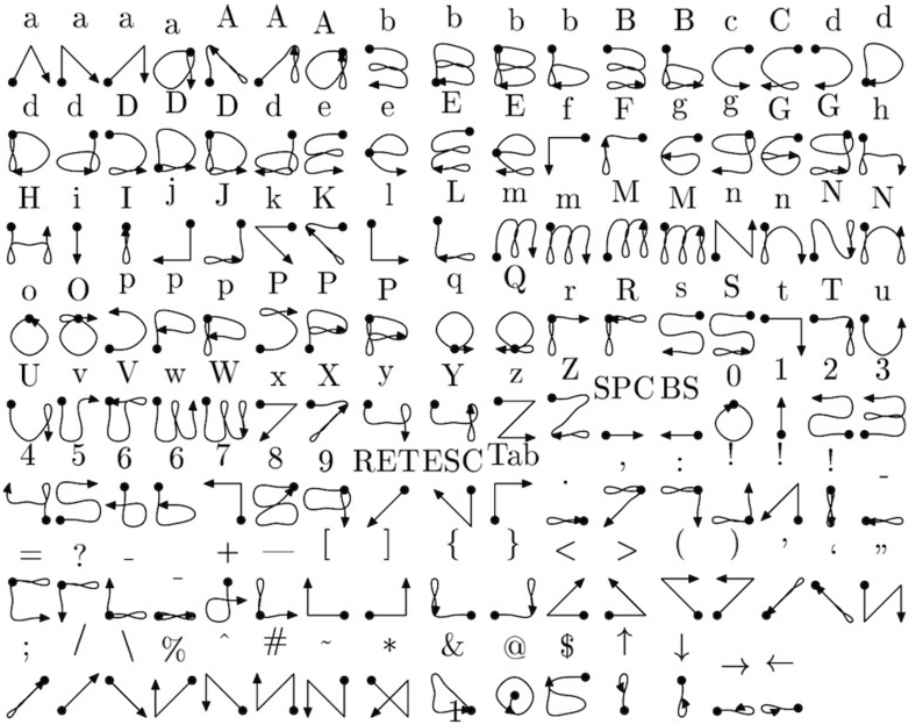
Don't use a ballpoint pen or sharp metal stylus as it could scratch or damage the touchscreen!

Start X-Windows on the PiTFT and open the LXDE menu by clicking the icon in the lower left corner. Scroll up to the **Accessories** menu at the top and notice the new **XStroke** and **XStroke Kill** commands.

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

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

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



(credit to Carl Worth for the image above)

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

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

Note that when xstroke is looking for gestures you might not be able to click or control the cursor as you normally would expect. To stop xstroke's gesture recognition carefully press the xstroke pencil icon again until the 'abc' text disappears. I've found this process can be a little finicky as the icon is very small and any movement will be interpreted as a gesture. Use a light touch and try a few times to click the icon.

If you get stuck completely and can't disable xstroke by clicking the icon, connect to the Raspberry Pi in a terminal/SSH connection and run 'killall xstroke' (without quotes) to force xstroke to quit. The normal way to stop xstroke is to

navigate to the **Accessories** -> **XStroke Kill** command, but you might not be able to do that if xstroke is listening for gesture input.

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

PiTFT PyGame Tips

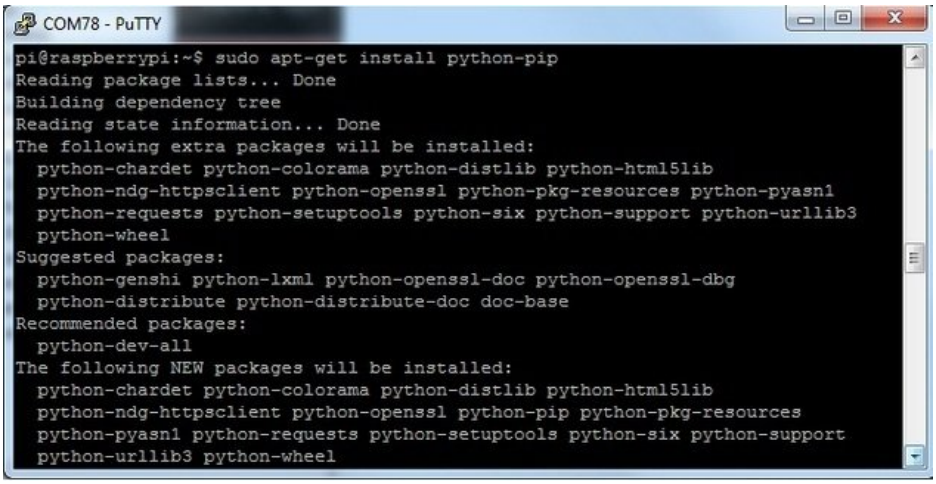
Since the PiTFT screen is fairly small, you may need to write custom UI programs. Pygame is the easiest way by far to do this.

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

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

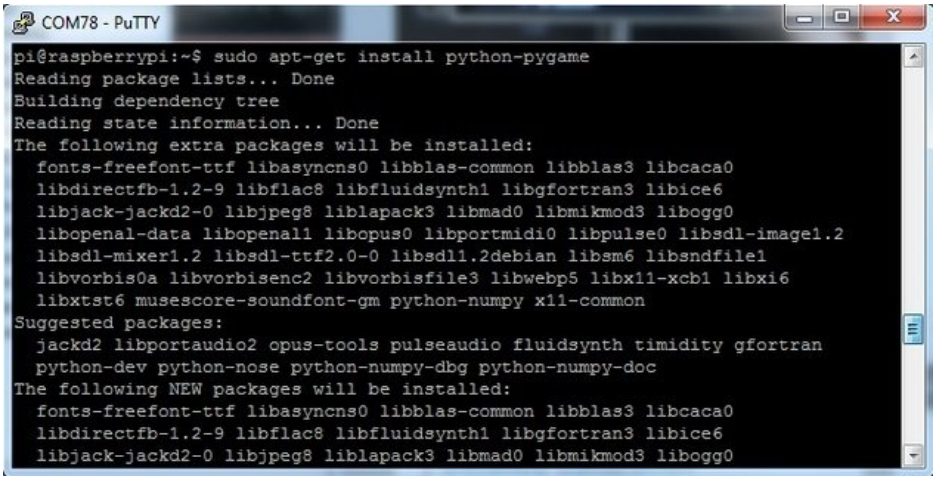
Install pip & pygame

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



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

(this will take a while)



Ensure you are running SDL 1.2

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

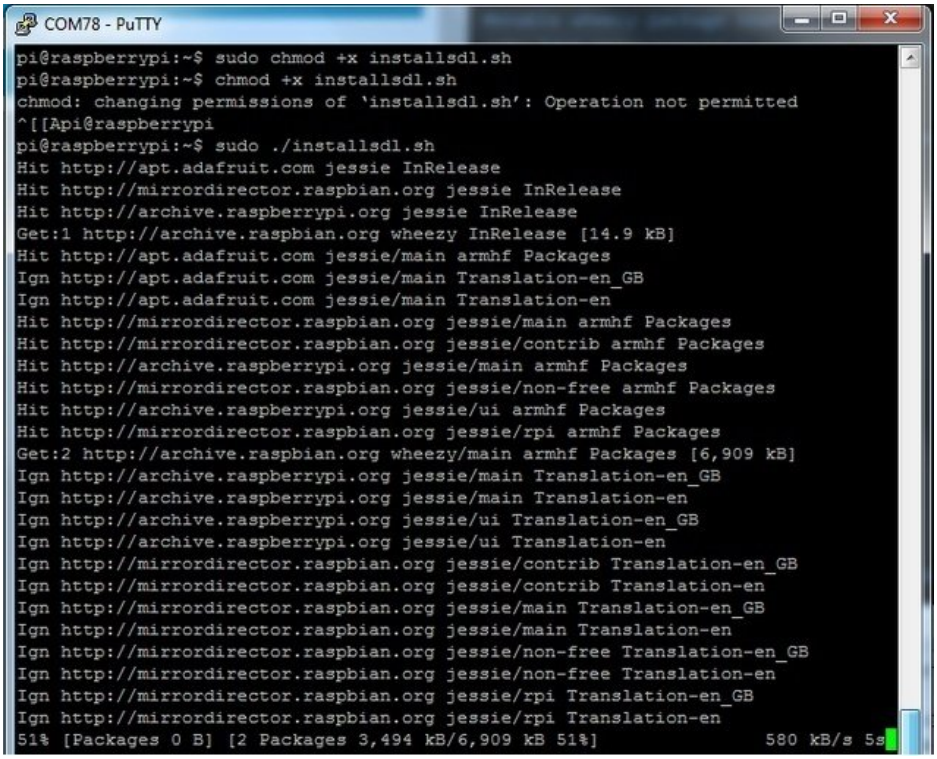
Edit a new file with **sudo nano installsdl.sh**
and paste in the following text:

```
#!/bin/bash

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

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

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

# install
apt-get update
apt-get -y --allow-downgrades install libsdl1.2debian/wheezy
```

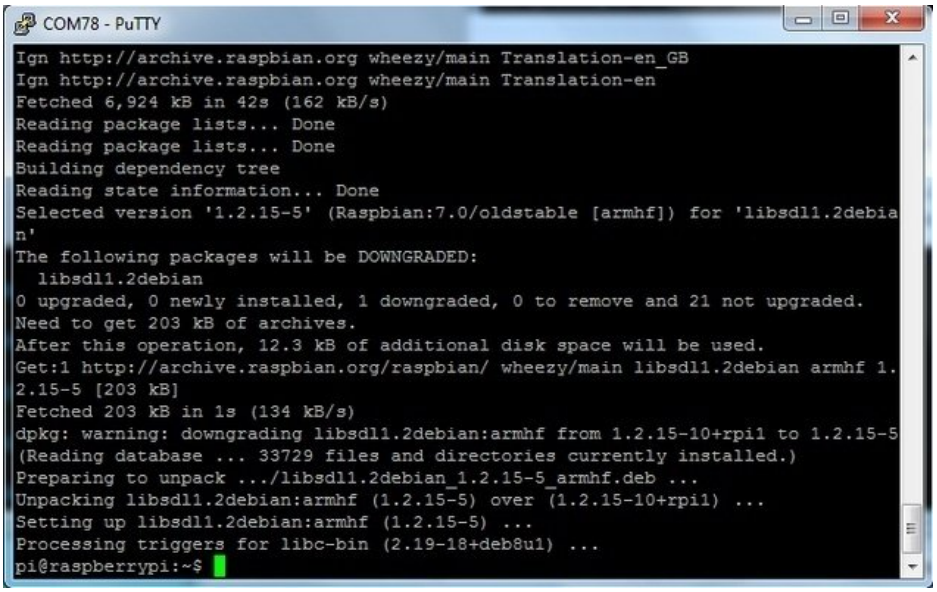
run

sudo chmod +x installsdl.sh

sudo ./installsdl.sh



it will force install SDL 1.2



OK **now** you can continue with pygame

Using the Capacitive touch screen in PyGame

The 2.8" Capacitive touch screen driver may not work by default in pygame, but this handy script shows how you can capture the device messages in python to create a UI

- https://github.com/nift4/pigame (https://adafru.it/CYv)
  For examples:

https://github.com/nift4/Raspberry-Pi-Testing (https://adafru.it/CYy)

F.A.Q.

The display works, but the capacitive touch part doesnt

Check that you installed the right image, there's one for resistive and one for capacitive PiTFT's

If that doesn't help, you can verify your RasPi model number with the command **cat /proc/cpuinfo**, if it's revision # **0002** or **0003** that means it's a rev 1 Model B, and will not work due to the I2C pins changing.

Does this screen do multi-touch?

Nope! This capacitive touch screen is single-touch only.

Hey...I was looking at the FT6206 datasheet and it looks like it can support multitouch (two points)!

The chip does in fact support multitouch, but the screen layout itself is single-touch.
We'll keep looking for a low cost multitouch screen, but we found that at the small size of this screen, single-touch is pretty good! Also, very few linux programs support MT.

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

Check out the 2.8" resistive PiTFT FAQ for an answer to this common question.

I have some more questions!

Check out the 2.8" Resistive PiTFT FAQ page for some other questions you may want answered

https://adafru.it/dJ2

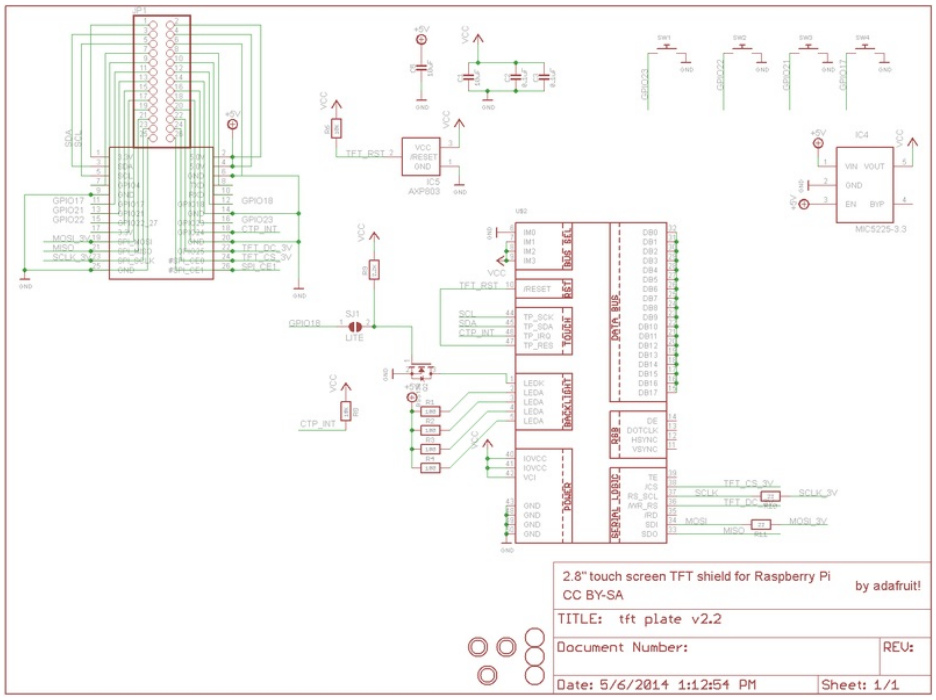
https://adafru.it/dJ2

Downloads

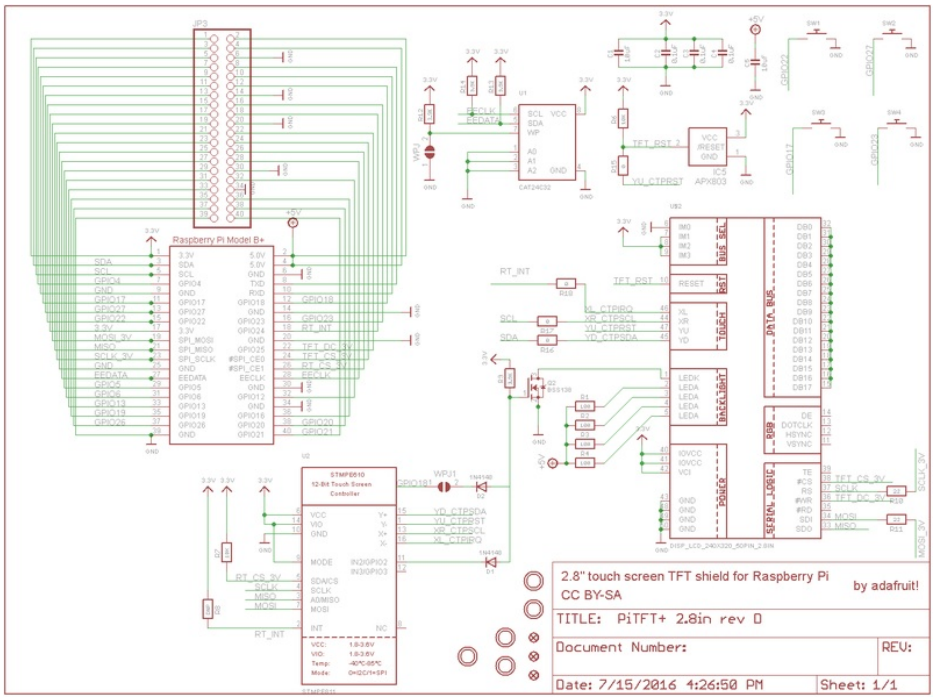
Files

- The latest kernel fork that adds all the TFT, touchscreen, and other addons is here on github (https://adafru.it/aPa)
- Datasheet for the 'raw' 2.8" TFT display (https://adafru.it/sEt)
- FT6206 Datasheet (https://adafru.it/sEu) & App note (https://adafru.it/dRn) (capacitive chip)
- EagleCAD PCB files on GitHub (https://adafru.it/oYC)

Schematic for Pi 1 Version

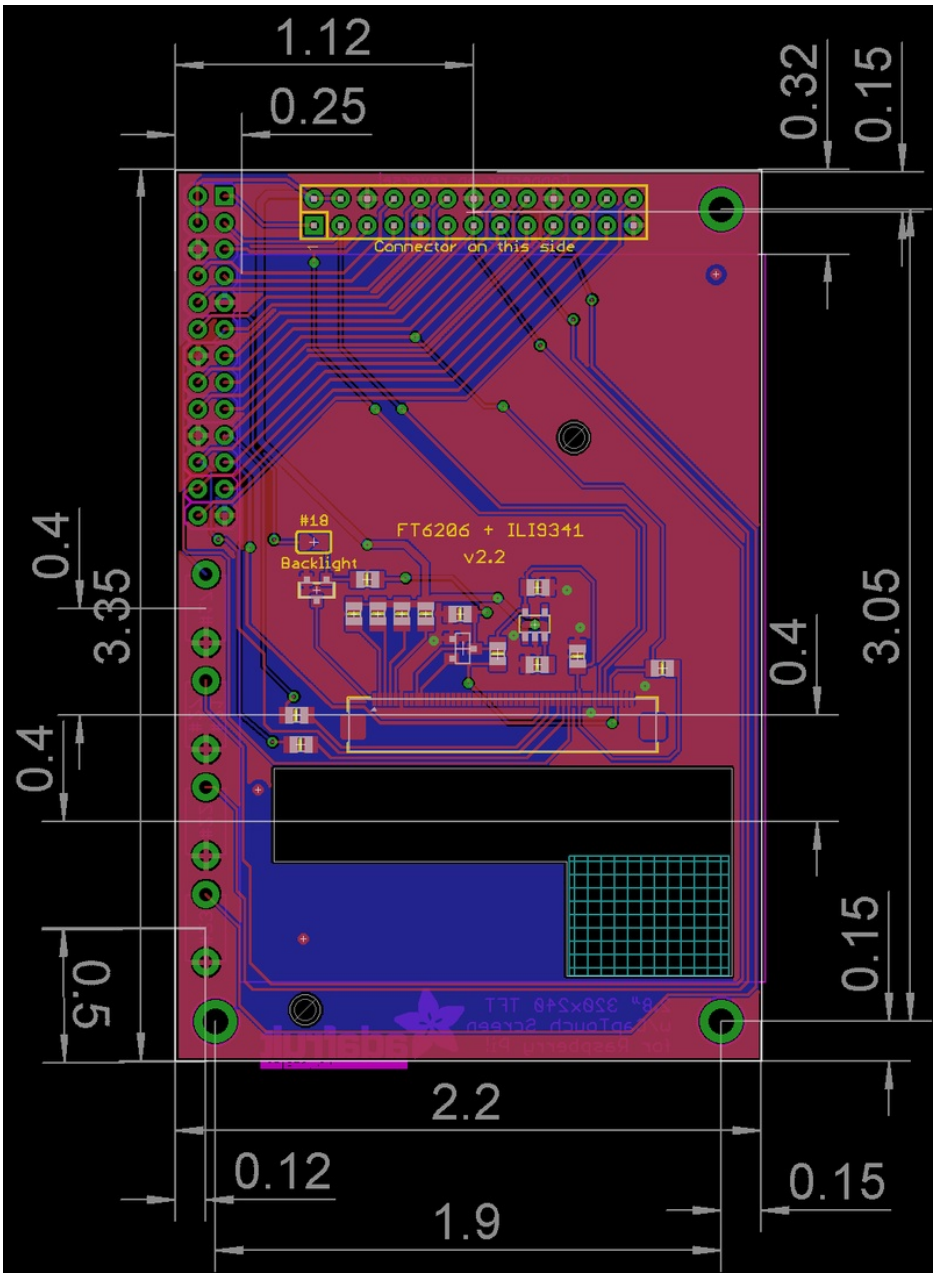


Schematic for PiTFT Plus (B+/Pi 2 shape)



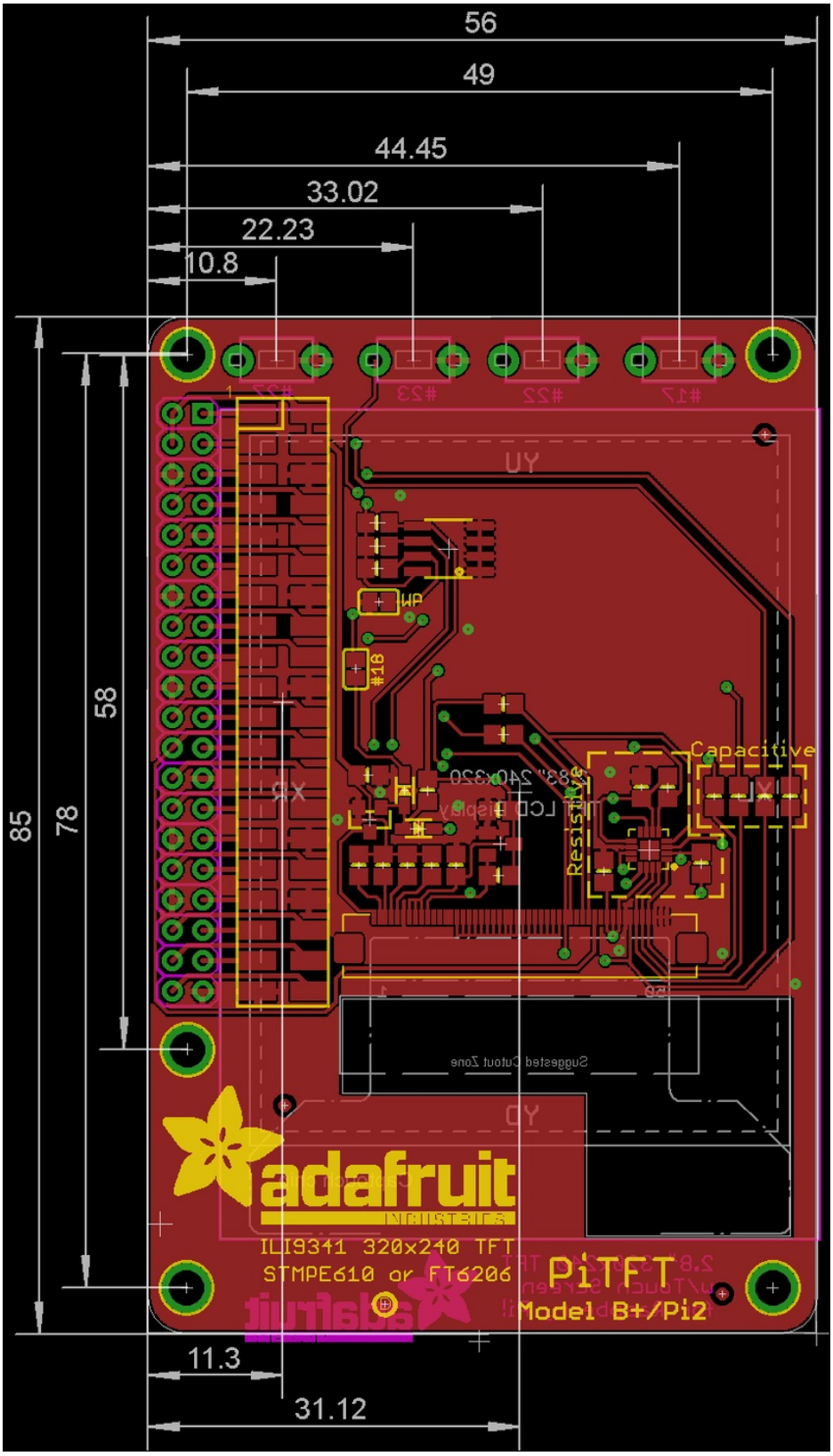
Fabrication Print (Pi 1 Version)

Dimensions in Inches



Fabrication Print (B+/Pi 2 Version)

Dimensions in mm


56
49
44.45
33.02
22.23
10.8
85
78
58
11.3
31.12
Capacitive
adafruit
INDUSTRIES
ILI9341 320x240 TFT
STMPE610 or FT6206
PiTFT
Model B+/Pi2