DIY WiFi Raspberry Pi Touchscreen Camera

Created by Phillip Burgess

Guide Contents

Overview

This project explores the Adafruit PiTFT touchscreen and the Raspberry Pi camera board to create a simple point-and-shoot digital camera. One can optionally use WiFi and Dropbox (a cloud file storage and synchronization service) to automatically transfer photos to another computer for editing.

This isn't likely to replace your digital camera (or even phone-cam) anytime soon…it's a learning exercise and not a polished consumer item…but as the code is open source, you or others might *customize it* into something special that your regular camera can't do.

Things You'll Need:

- **Raspberry Pi computer**, *any* model will work. This project is not especially demanding, so it's a great use for an older board if you have one around.
- **PiTFT display.** Various sizes and models are available — for current Raspberry Pi boards (Pi 3, Pi 2, B+ and A+) you'll want one of the *PiTFT Plus* variants. For older Pi boards (original Model B or A), a "non-Plus" PiTFT is needed. See featured product links to the right. It *must* be a PiTFT (connecting to the GPIO header,**not composite or HDMI**) with **320x240** resolution (no higher, i.e. 3.5" PiTFT won't work), *must* have **resistive or capacitive touch**.
- **Raspberry Pi Camera Board.** Current "v2" cameras or original v1, any will do. For general work you'll want the **regular version (http://adafru.it/3099)**. For special projects like security or nighttime nature photography, you can experiment with the **infrared version (http://adafru.it/1567).**
- **MicroSD memory card (http://adafru.it/1294),** 4GB or larger.

The following parts are **optional:**

- A **WiFi adapter (http://adafru.it/814)** allows the camera to upload photos to Dropbox (requires account, free). *The Raspberry Pi **3** has WiFi **built in**, so this part isn't needed with that board.*
- A **USB battery pack (http://adafru.it/1565)** makes the whole camera portable. If using WiFi, you'll want a robust battery pack that can provide 1 Amp (some are limited to 500 mA max).

Some additional parts, tools and skills are also required:optional buttons or headers on PiTFT displays require soldering; some means of holding all the pieces together — could be as simple as a few rubber bands, to a drilled-out plastic electronics enclosure, to an elaborate custom 3D-printed case. This all depends on your available resources. Read through to see what's involved in the project and come up with ideas along the way.

Other Raspberry Pi – Point & Shoot Cameras!
James Wolf made a Raspberry Pi - Point & Shoot Camera all inside the original Pi case (except battery). He made a little board attached to a cut down ribbon cable, just for the pull up resistors and the button - instructions, pictures and a link to the simple Python file are located on his site (http://adafru.it/d65).

Pi Setup

The optional tactile buttons on the PiTFT are not required for this project. You can install the buttons for other things if you like, but the camera software is entirely touchscreen-based.

Install & Test PiTFT Operating System

To ensure that all the software interdependencies work, it's best to **start with a clean installation**.

Format a **4GB or larger** SD card and load it up with the Raspbian operating system. This guide explains how to prepare a card for the Raspberry Pi (http://adafru.it/aWq).

You'll save a ton of setup time by using one of our pre-built Raspbian images with PiTFT support already included. The Raspbian "Lite" edition is recommended for this project. You'll find links to download PiTFT-ready Raspbian Lite images here:

- Use this link for PiTFT **RESISTIVE TOUCH** screens (http://adafru.it/sha) (select the "Lite" version)
- Use this link for PiTFT **CAPACITIVE TOUCH** screens (http://adafru.it/shb) (select the "Lite" version)

Make sure to download the correct Raspbian image for your screen type. The first link is for **resistive** touch screens, second is for **capacitive** screens. They won't work the other way 'round!

IMPORTANT: performing an 'apt-get upgrade' on a working system will BREAK PiTFT SUPPORT. Don't do this. We'll update links for newer images as they're available.

Make sure you've got the Raspberry Pi booting with the Text mode display on the PiTFT before you continue. You'll need to have that PiTFT stuff all working!

Once you have it working, log in and then shutdown with **sudo shutdown** at the command line

Install Camera

With the Raspberry Pi powered off, install the Pi camera ribbon connector, install the PiTFT

display atop the GPIO header, connect a USB keyboard, power the Raspberry Pi, and work through the usual first-time boot configuration.

Run **sudo raspi-config** after logging in to run through the configuration

The raspi-config menu is a bit scrunched on the PiTFT, but the vital options are all present.

•

The following option is **required**:

- *Expand Filesystem*

The following are very useful and **recommended:**

- Under *Internationalization Options,* select *Change Timezone* and *Change Keyboard Layout* to match your region. If your keyboard is producing unexpected characters, this is usually the reason why.

The following are **optional:**

- Under *Advanced Options,* select *Hostname* to give this Pi a unique name (such as "picam") to distinguish it from other Raspberry Pi's on the network.
- Other settings can be configured to your liking.

The following should **not** be used:

- *Overclock.* This is a portable, battery-operated project and an overclocked Pi will draw more current. Overclocked systems don't always play well with the PiTFT and are more likely to corrupt the SD card filesystem. Do not enable this option.

Reboot when prompted, then more configuration awaits…
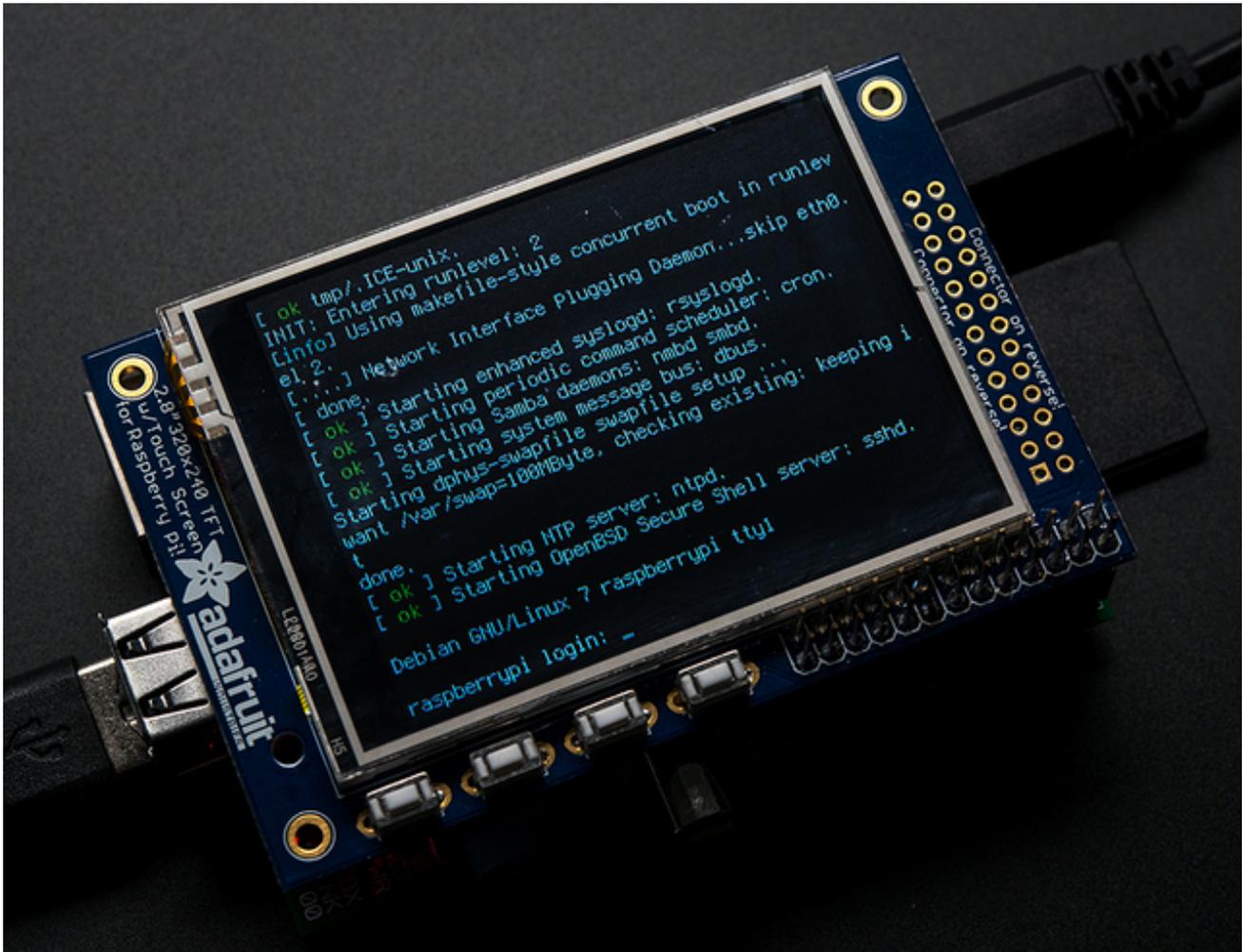
Even if you don't plan to use the Dropbox functionality of this project, it's **necessary to get the Raspberry Pi on your network** at least temporarily to download additional software. This can be done using the wired **Ethernet** jack (no additional configuration needed), or over **WiFi** using either a USB WiFi adapter or the Pi 3's built-in wireless networking. **This guide may be helpful for setting that up (http://adafru.it/aUB).**

Do not continue until the Pi is successfully on the network. You can log in and try "ping adafruit.com" from the command-line to test.

Easy Install

Our camera software requires a complex set of software dependencies. We've written a script that takes care of all the ugly parts. You can download and run with these three lines (this is easiest if you login via *ssh* and copy-and-paste these lines):

cd ~
curl -O https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/pi-touch-cam.sh

```
sudo bash pi-touch-cam.sh
```

The script explains what it's about to do and prompts for a "Y" before continuing; any other input will cancel.

It may take **10 minutes** or so to run. Afterward, you'll be prompted to **reboot** the system again. If this all works, skip ahead to the "**Testing**" section below.

Used the "Easy Install" route? Skip ahead to "Testing" below!

Complex Install From Scratch

If you used the "**Easy Install**" directions above, you can **ignore this section** and **skip ahead** to "**Testing**" below.

These are the steps taken by the pi-cam-install.sh script, if for some reason you need to perform any or all of these steps manually…

System Tweaks

The file /**boot/config.txt** is modified to enable the camera (if not already active via raspi-config) and also boost the speed of the PiTFT display (changing the "speed" and "fps" values). These lines are usually at the bottom of the file.

```
dtoverlay=pitft28c,rotate=90,speed=80000000,fps=60
start_x=1
gpu_mem=128
```

Install Prerequisite Software

Our code relies on a few libraries for handling the camera and screen output. One of these, it's necessary to intentionally *downgrade,* in order for the touchscreen to interoperate correctly. It's complicated. These lines in the script set that up (but don't actually perform the downgrade yet):

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

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

# Set priority for libsdl from wheezy higher then the jessie package
echo "Package: libsdl1.2debian
```

Pin: release n=jessie
Pin-Priority: -10
Package: libsdl1.2debian
Pin: release n=wheezy
Pin-Priority: 900
" > /etc/apt/preferences.d/libsdl

Update the APT package index files and install Python libraries:

sudo apt-get update
sudo apt-get -y --force-yes install python-picamera python-pygame python-imaging

Now we perform the actual SDL library downgrade (it has to follow the "apt-get update" above):

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

Download the Camera and Dropbox Scripts

Just a couple more steps to download and uncompress these…

cd ~pi
wget https://github.com/andreafabrizi/Dropbox-Uploader/archive/master.zip
unzip master.zip
rm master.zip
mv Dropbox-Uploader-master Dropbox-Uploader

wget https://github.com/adafruit/adafruit-pi-cam/archive/master.zip
unzip master.zip
rm master.zip

Modify /etc/rc.local for auto-start (but don't enable)

These lines are added to /etc/rc.local — the second is intentionally commented out by default. It's a good idea to test the camera software manually before throwing the switch. These are inserted just before the final "exit 0" line that's normally present.

Enable this line to run camera at startup:
cd /home/pi/adafruit-pi-cam-master ; sudo python cam.py

Testing

Now give it a try. The software must be run as root (using the *sudo* command) in order to access the TFT display:

cd adafruit-pi-cam-master
sudo python cam.py

If all goes well, after a few seconds' initialization you should see a live viewfinder preview on the screen, as well as two onscreen buttons.

If this *doesn't* happen, an error message should give some sort of troubleshooting guidance; missing library or driver, etc.

There's still some work to be done if we want to use Dropbox, so quit the camera program for the time being…tap the gear icon (settings), the left arrow and then the confirmation button. You'll be back at the command line now.
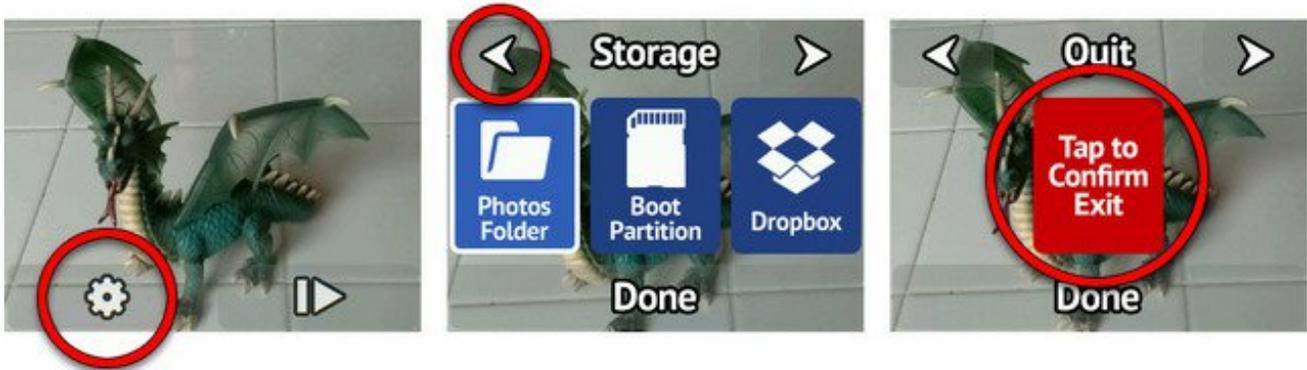
Standalone mode

You can have the Pi boot straight into the camera software at startup by editing **/etc/rc.local** (this must be done as root, so "sudo" your text editor of preference):

sudo nano /etc/rc.local

If you use the Easy Install script, the following line is already present, but commented out. Delete the initial "#" character on the line, save changes and reboot.

Otherwise, if doing this the long way, add the following line just before the final "exit 0":

cd /home/pi/adafruit-pi-cam-master; python cam.py

Next time you reboot you should see the text console for a few seconds and then it will start the cam.py software.

Dropbox Setup

Dropbox is a "cloud" file storage and syncing service. A basic account is free and offers two gigabytes of storage. We can leverage this as a means of getting photos off the Raspberry Pi onto other devices (desktop computer, etc.) without cables or swapping cards.

This pretty much requires a WiFi connection; trying to work a camera with an Ethernet cable continually attached would get tiresome. It also requires being in range of your wireless access point, with the Raspberry Pi suitably configured for access.

We'll use **this guide from RasPi.TV (http://adafru.it/d56)** (with minor changes as Dropbox has evolved) to get Dropbox and the Raspberry Pi to work together. The steps include:

- Set up Dropbox account if you don't already have one.
- Download and set up Dropbox Uploader on the Raspberry Pi (this is done from the command line).
- Create a Dropbox app and get access credentials.

1. Create a Dropbox account if you don't already have one.

A basic account (2GB limit) is free. Sign up at Dropbox.com (http://adafru.it/sCo).

2. Download and set up Dropbox Uploader software

If you used the "**Easy Install**" script from the prior page, the Dropbox Uploader software is **already present** in your home directory and this step can be skipped.

cd ~pi
wget https://github.com/andreafabrizi/Dropbox-Uploader/archive/master.zip
unzip master.zip
rm master.zip
mv Dropbox-Uploader-master Dropbox-Uploader

3. Create a new Dropbox App through the Dropbox developer site

Visit https://www.dropbox.com/developers/apps (http://adafru.it/dfp) and log in with your

Dropbox account credentials.

Use the "Create app" button to begin the process…

My apps   Create app

Select "Dropbox API" and "Full Dropbox," then assign your app a unique name (e.g. "Bob's Raspberry Pi Camera"), then click "Create App."

Create a new app on the Dropbox Platform

1. Choose an API

Dropbox API
For apps that need to access files in Dropbox. Learn more

Dropbox Business API
For apps that need access to Dropbox Business team info. Learn more

2. Choose the type of access you need

Learn more about access types

○ App folder – Access to a single folder created specifically for your app.

● Full Dropbox – Access to all files and folders in a user's Dropbox.

3. Name your app

My Unique Camera App Name

Create app

In the Settings tab for your new app, there's a section with the heading "OAuth 2." Look for the button "Generate access token." This will give a long string of seemingly random letters — a unique identifier for tying your camera to your Dropbox account.

This last step is easiest if you have SSH enabled on the Pi and are logged in through a terminal program, so you can just copy-and-paste rather than having to type in that whole string exactly…

4. Set up Dropbox Uploader

You should be inside the Dropbox-Uploader directory at this point (i.e. typing "pwd" should return "/home/pi/Dropbox-Uploader").

Run the dropbox_uploader.sh script:

./dropbox_uploader.sh

You'll be prompted to enter your access token that was generated in Step 3. This must be *exact,* which is why ssh and copy-and-paste is so helpful.

If you mess up this process, you can run "./dropbox_uploader.sh unlink" to clear out Dropbox Uploader's settings and start over.

Let's test it!

echo 12345 > foo.txt
./dropbox_uploader.sh upload foo.txt /

This will create a small text file (foo.txt) containing the string "12345" and will upload it to the root level of your Dropbox drive. If Dropbox is installed on your "main" computer the file should appear there after a moment, else use a web browser to log into your Dropbox account and you should see it there among your files.

If this works, then next time you run the cam.py script you can go into *Settings* and select *Dropbox* for storage. Photos will be saved both in the "Photos" folder and uploaded immediately to Dropbox.

Using the Camera

You briefly saw the camera interface during testing. We can explain in more detail now how it works and what options are available.

Upon startup, the camera program presents a live viewfinder and a couple of buttons. The majority of the screen itself functions as the shutter "button" — tap to take a still photo.

At the bottom of the screen are two buttons. The left button (with the gear icon) will take you to various settings. The right button (with the "play" symbol) lets you review previously-taken photos (if no photos have been taken yet, the camera will let you know it's "empty").
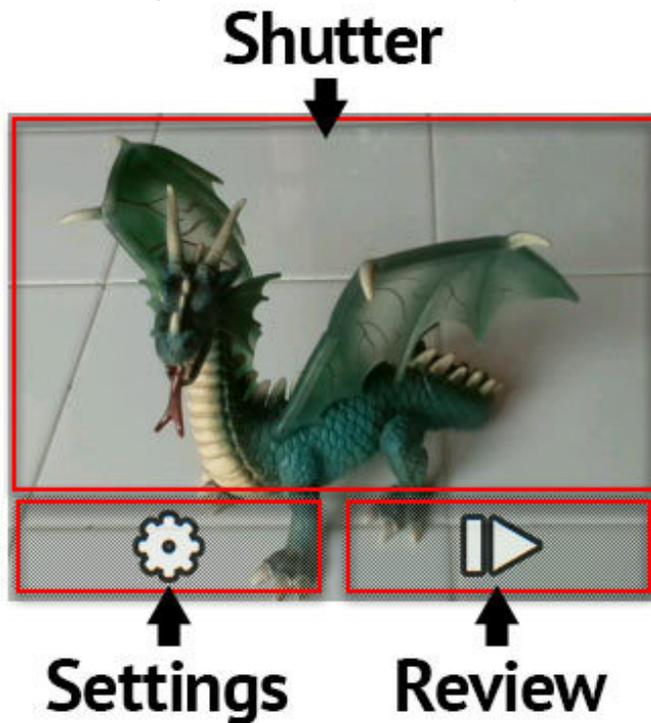
The *Settings* menu provides access to camera settings. This is not an exhaustive list of every feature possible with the Raspberry Pi camera, just a few essentials to get you started.

The left/right arrow buttons at the top of the screen select among the settings options:

The *Storage* screen selects between three different options, each with some pros and cons:

- **Photos Folder:** images will

-

be saved inside a "Photos" folder in your Raspberry Pi home directory (the folder will be created if it doesn't exist). They can be easily accessed from other programs on the Pi, but this partition isn't easily accessed when inserting the SD card in other computers.

- **Boot Partition:** images will be saved in the folder "/boot/DCIM/CANON999" on the boot partition. When the SD card is inserted in another computer, it mimics a card from a digital camera and may import photos automatically, depending on system settings. The downside is that space in the boot partition is <u>very</u> limited; you might only store a dozen or so photos there.
- **Dropbox:** as previously discussed, images are saved in the Photos folder as well as uploaded to Dropbox (if WiFi is connected and Dropbox configured).

The *Size* screen selects from three different image sizes:

- **Large (2592x1944, 4:3 ratio):** this is the largest size (5 megapixels) supported by the Raspberry Pi camera. The actual area

- captured stretches well beyond what's shown in the live viewfinder though.
- **Medium (1920x1080, 16:9 ratio):** HD resolution, widescreen, 2 megapixels.
- **Small (1440x1080, 4:3 ratio):** 1.5 megapixels.

The latter two modes *should* display the actual full photo boundaries in the live viewfinder mode, but don't yet (something in the camera library documentation doesn't correspond to reality). This is a work in progress and will be addressed once its understood.
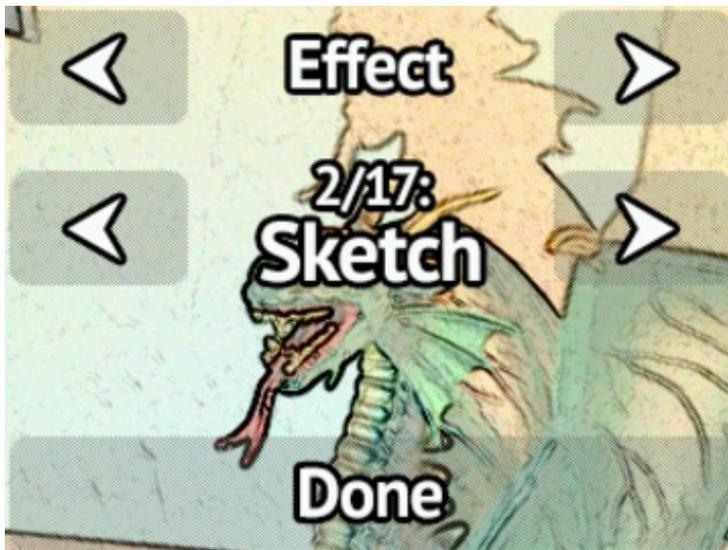
The *Effect* screen is where all the fun happens.

There are 16 different artistic effects that can be applied to photos (plus "normal," no effect). Make your photos look like an oil painting, or a pen sketch, or turn the colors weird! These all operate on the live preview as well.

The *ISO* setting adjusts the camera's sensitivity to light.
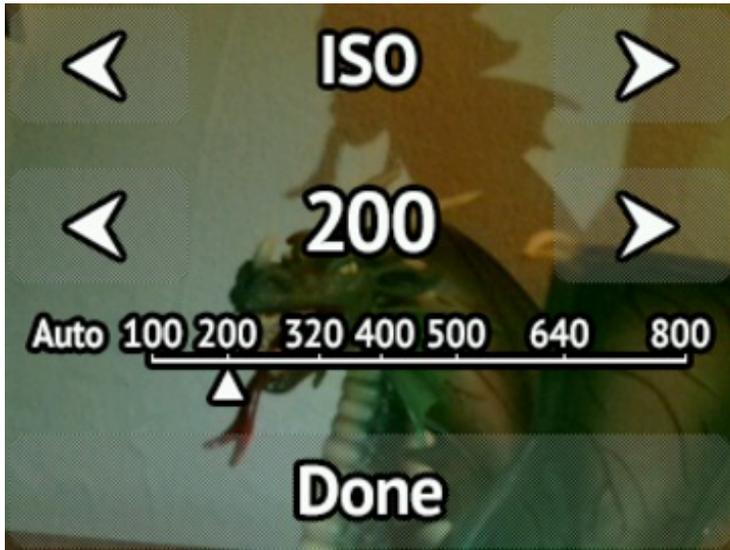
This is a tradeoff…more sensitive settings (higher numbers) work better in low light, but the resulting image may be grainy.

ISO has no effect on the live viewfinder, only captured photos.

The last screen is an option to quit the camera program, returning to the command line.

Tap the red button to exit, the arrow buttons for other settings, or the Done button to cancel.

The "Done" button returns to viewfinder mode.

All the camera settings will be saved; next time you run the script, all prior settings will be as you left them.

Next Steps…

This is just a taster. Where you go next is up to you…

- The camera board could be swapped out for the infrared-sensitive Pi NoIR (http://adafru.it/1567) camera.
- The Icon and Button classes in thecam.py script are pretty rudimentary, but might be sufficient for your own touchscreen applications (whether photography-related or otherwise).
- The camera code could be expanded to do new things. There are *lots* of camera settings we haven't even touched (http://adafru.it/d59). Other capabilities like a self-timer, intervalometer or motion detection could be added. Or capturing video.

-

Look at this embarrassing assemblage of *rubber bands!* There's so much more that could be done here for a proper enclosure…perhaps even laser-cut or 3D-printed if you have access to such.

The package could be slimmed down considerably; there's a *huge* amount of empty space between the PiTFT and Raspberry Pi (even more with a Model A board). Advanced

makers could squeeze a slim LiPo battery and a 5V boost converter in there, connecting to the expansion header at the right edge of the TFT board instead of the side-protruding USB power connector. The result would be similar in size to some consumer point-and-shoot digital cameras.

3D Printing an Enclosure

This case design fits the ORIGINAL Model B Pi. For newer boards such as the Pi 3, check Thingiverse.com for alternative case designs that could be adapted.
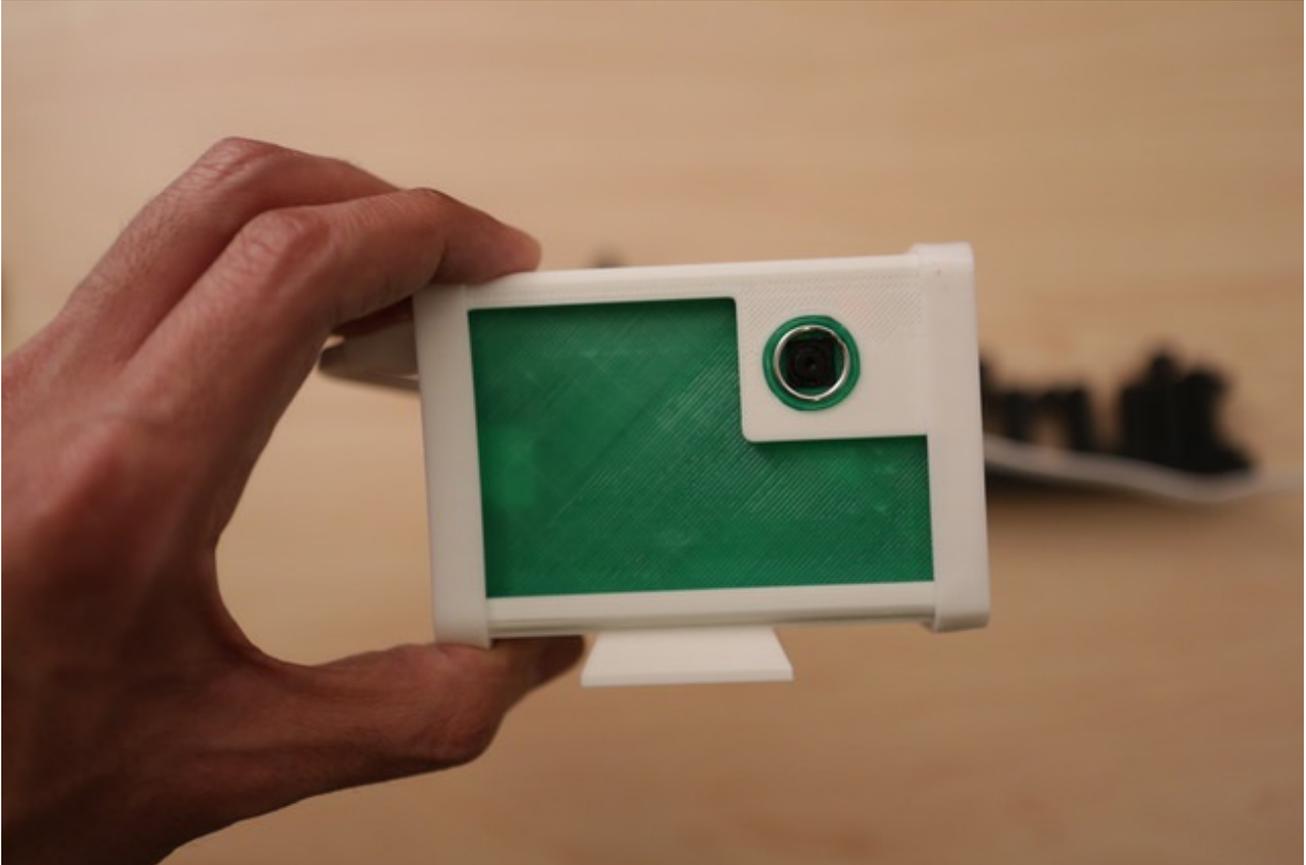
3D Print a Camera Enclosure!

We designed a multi-part enclosure that snap fits together. Optimized for PLA to print without any rafts or supports! Follow our recommend print/slicing settings for best results. Our assembly guide walks your through the process of putting it together.
[Download STLs](http://adafru.it/d9s)
http://adafru.it/d9s

Snap-fit color enclosure

Print pitouch-cover.stl file in a different color for a stylish look. We used transparent green PLA for the cover and true white for the rest, but you can choose any color you like.

Optional Parts

- **ptouch-tripod.stl** If you don't want a Tripod Mount
- **ptouch-band.stl** If you don't want Grippy edges
- **ptouch-camring.stl** If you don't want interchangeable lenses.

|		
pitouch-top.stl about 75 minutes 14g	Houses the camera and Raspberry Pi. Snaps to**pitouch-bottom.**	PLA @230 2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-bottom.stl about 70 minutes 13g	Houses the touch screen and Raspberry P.	PLA @230 2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-cover.stl	Fits inside **pitouch-top.stl**. Raspberry Pi Camera press-fits to	PLA @230 2 shells 15% Infill

about 30 minutes 6g	cover.	2.0 Layer Height 90/150mm/s
pitouch-tripod.stl about 35 minutes 6g	Mounts to bottom of **pitouch-top.stl** and **pitouch-bottom.stl**	2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-band.stl about 25 minutes 5g	Attaches to sides of pitouch-top.stl and **pitouch-bottom.stl**. Adds grip and keeps **pitouch-tripod** mounted to **pitouch-top.stl** and **pitouch-bottom.stl**.	NinjaFlex @225 2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-camring.stl about 3 minutes 0.5g	Snaps to **pitouch-top.stl** camera hole.	PLA @230 2 shells 15% Infill 2.0 Layer Height 90/150mm/s

# Assembling Enclosure
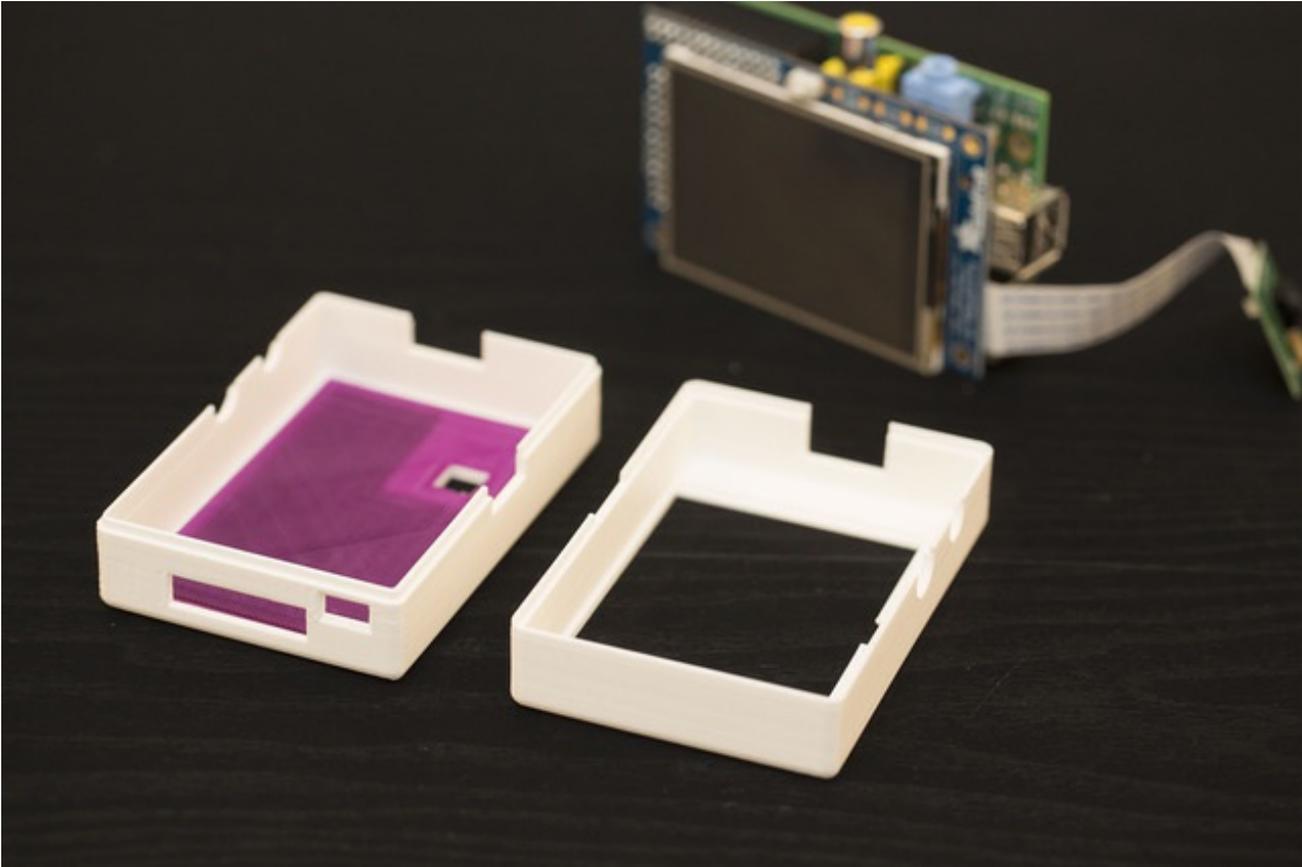
Mounting Pi Cam

Start by adding a piece of tape to the back of the camera's PCB, to
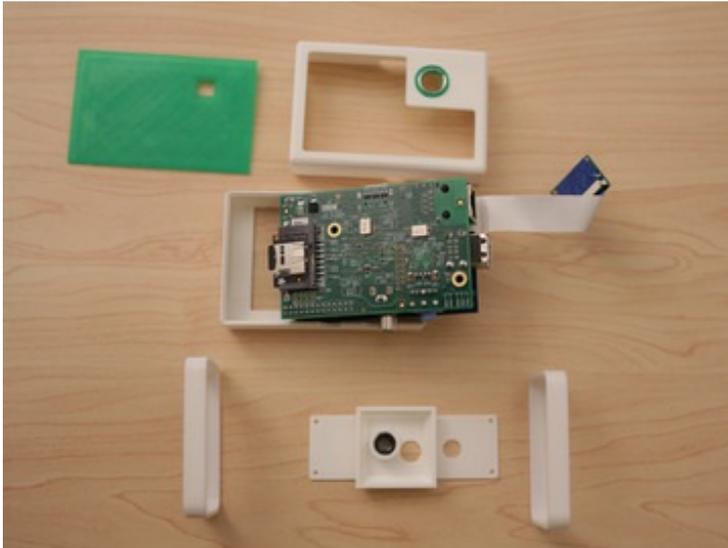
- prevent any shorts. Use a short ribbon cable and flip it over to the back of the Pi, covering the ethernet port.

Position the camera with the camera hole in the pitouch-cover.stl part. Align it up so the nubs snap into the left and right mounting holes of the pi cam. Press fit the cover into the **pitouch-top** part.
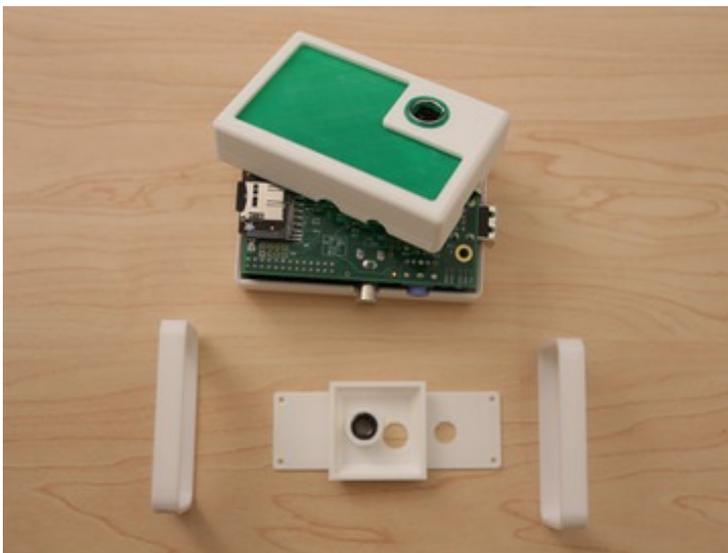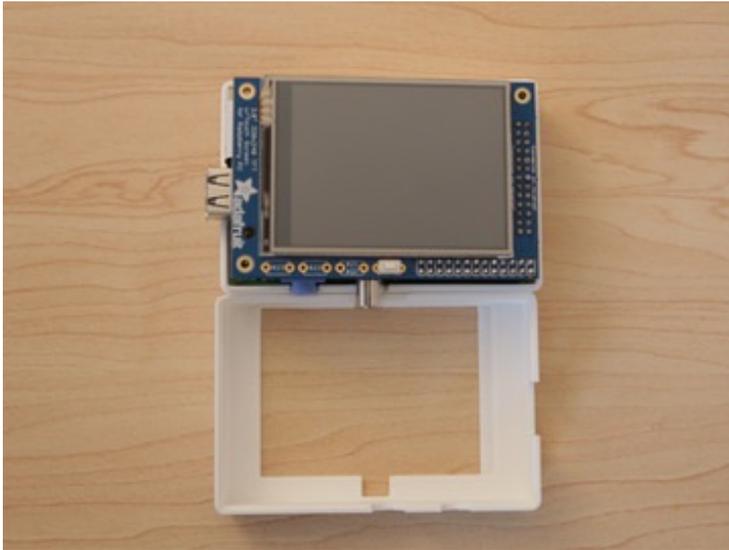

- Position the **pitouch-top** part over the Raspberry Pi and angle it down so the SD card slot slips into the proper opening of the pitouch-part.

Adding Lenses

Snap the **pitouch-camring** part into the camera hole of the **pitouch-top** part. Insert a 12mm split ring into the **pitouch-camring**. The photojojo lens use magnets for mount and connects to the split ring.

-

Mounting the Raspberry Pi and Touch Screen

With the camera and pi mounted to the **pitouch-top** part, flip the the body and place the **pitouch-bottom** part over the touch screen. Press the **pitouch-bottom** part down snapping the parts together.

Adding Side Bands and Tripod Mount

You can optionally add a tripod mount and grippy bumpers. The Tripod mount can be attached to the bottom of the camera body with 4 small screws. Align up the holes and add screws. Add a standard tripod nut to the bottom of the tripod for attaching to other tripod mounts.

Upload, Share and Make

Plug in power to the micro-USB port and photos to your hearts content! If you do make one, please be sure to snap a picture of your creation and click the "I Made One (http://adafru.it/d9t)" button to our thingiverse (http://adafru.it/d9u) design so we can feature it on our blog!