adafruit learning system

DIY WiFi Raspberry Pi Touchscreen Camera

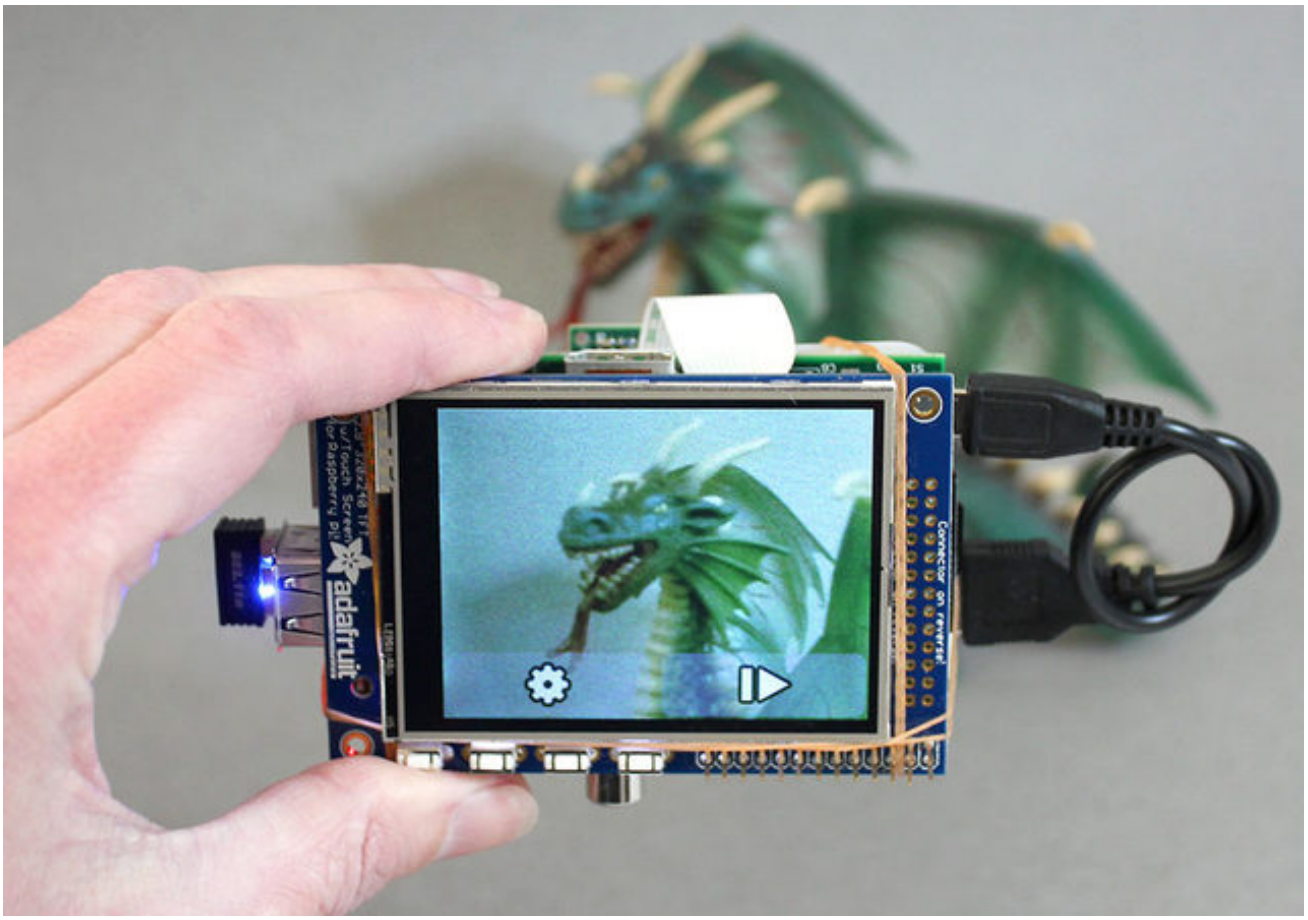
Created by Phillip Burgess

Last updated on 2014-12-29 07:00:12 PM EST

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 simplistic learning exercise and not a polished consumer item…but as the code is open source, you or others might *customize it* into something your regular camera can't do.

Things You'll Need:

- **Raspberry Pi computer**, either the **Model B (http://adafru.it/998)** or **Model A (http://adafru.it/1344)**

With Ethernet and two USB ports, the **Model B** (http://adafru.it/998) is usually easier to set up. The **Model A** (http://adafru.it/1344) saves a few bucks and uses less power…but with only one USB port and no Ethernet, it requires some added skill and/or shenanigans to get all the software fully loaded.

- **PiTFT Mini Kit (http://adafru.it/1601)** — TFT+Touchscreen for Raspberry Pi
- **Raspberry Pi Camera Board.** For general work you'll want the **regular version (http://adafru.it/1367)**. For special projects like security or nighttime nature photography, you can experiment with the **infrared version (http://adafru.it/1567).**
- **SD memory card (http://adafru.it/102),** 4GB or larger. To make the camera slimmer and more portable, you can optionally use a **microSD** card with either our **low-profile (http://adafru.it/966)** or **Shortening (http://adafru.it/1569)**

adapters.

The following parts are **optional:**

- A **WiFi adapter (http://adafru.it/814)** allows the camera to upload photos to Dropbox (requires account, free).
- A **USB battery pack (http://adafru.it/1565)** makes the whole camera portable. If using WiFi and/or a Model B Pi, you'll want a robust battery pack that can provide 1 Amp (some are limited to 500 mA max).
- In some situations a **USB to TTL Serial Cable (http://adafru.it/954)** may be the preferred way to log in and configure the Raspberry Pi, if a spare keyboard and monitor are unavailable.

Some additional parts, tools and skills are also required: soldering iron and solder for connecting the header to the PiTFT display; 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.

This was written for the original (resistive) PiTFT touch display. It can be made to work on the newer capacitive screen, but requires some extra steps detailed on the next page.

Other Raspberry Pi – Point & Shoot Cameras!
James Wolf made a Raspberry Pi - Point & Shoot Camera all inside the original 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

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

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

Connect a monitor and keyboard (or use a USB-to-serial console cable), power the Raspberry Pi from a USB phone charger or powered hub, and work through the usual first-time boot configuration.

The following options are **required**:

- *Expand Filesystem*
- *Enable Camera*

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

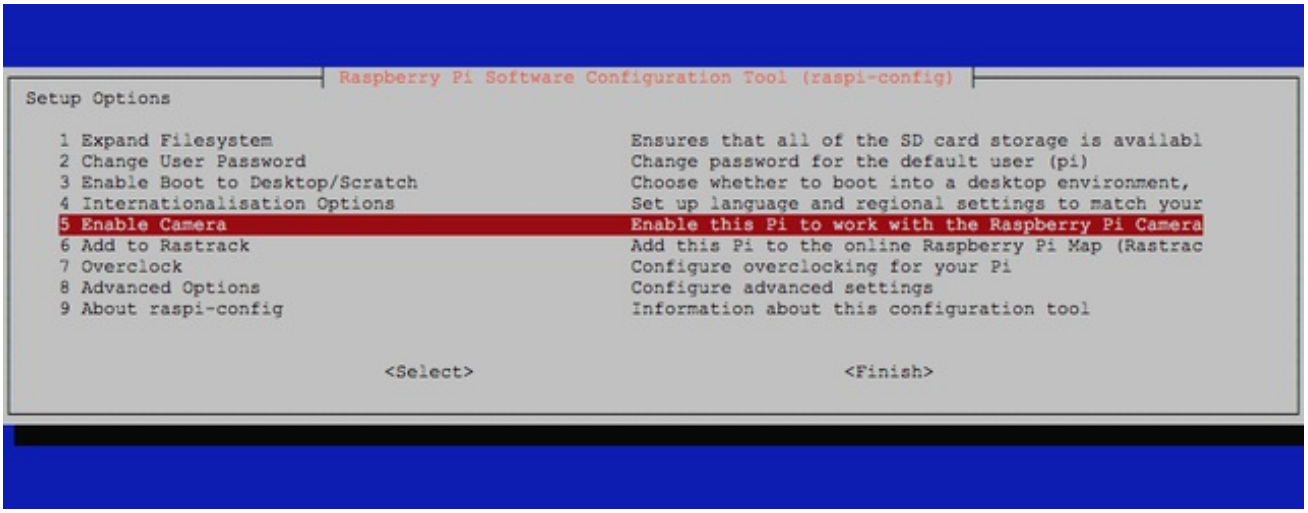
- Under *Internationalization Options,* select *Change Timezone* and *Change Keyboard Layout* to match your region.

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.
- Under *Advanced Options,* select *SSH* to enable command line access from the network (helpful for further configuration and troubleshooting).
- 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 are also more likely to corrupt the SD card filesystem. Do not enable this option.



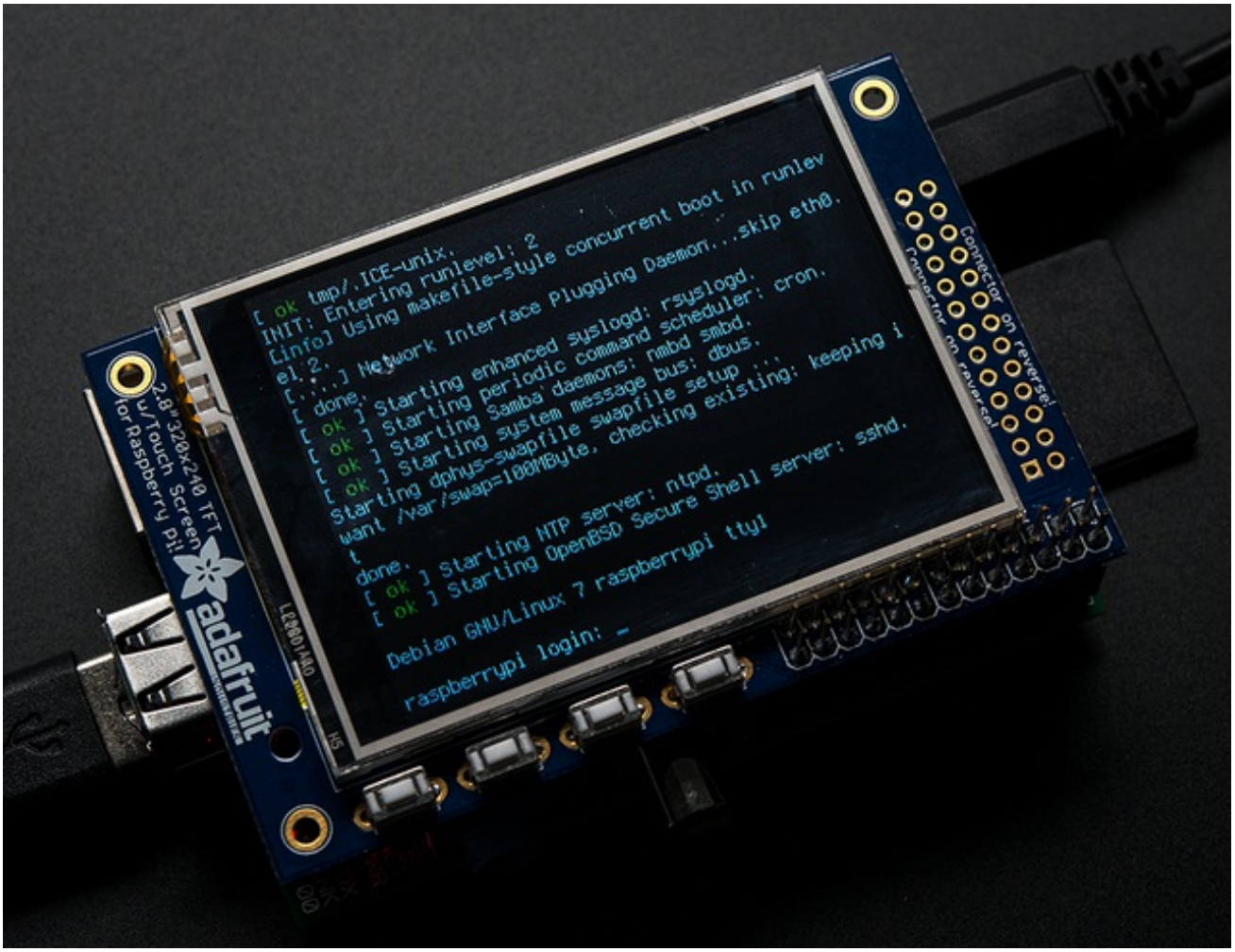
Once the basic system configuration is done, you can also set up wireless networking if you plan on using this capability. This guide may be of assistance (http://adafru.it/aUB).

Once the Pi is fully configured and on the network, work through the PiTFT tutorial:

Adafruit PiTFT — 2.8" Touchscreen Display for Raspberry Pi (http://adafru.it/d4W)

Work through the Assembly, Software Installation, Touchscreen Install & Calibrate and Using the Console pages, at least. The others (e.g. Playing Videos, adding the shutdown button) are not crucial to the camera project but can be done if you're also interested in exploring these capabilities. Once you have a Pi that boots to a login prompt on the 2.8" TFT, you're in good shape.

Likewise, 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.



Fetching the camera software is just a few extra steps…

First, install the Python library for the Raspberry Pi camera:

```
sudo apt-get install python-pip
sudo pip install picamera==0.8
```

(It's important to use version 0.8 with this, not the shiny new 1.0. The latter is still a bit unstable with some features we're using. We're revise this with a newer release once the situation changes.)

Then retrieve our Python camera application:

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

Finally, 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 using a capacitive touch PiTFT, see the notes below regarding some extra steps that need to be made first.

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.

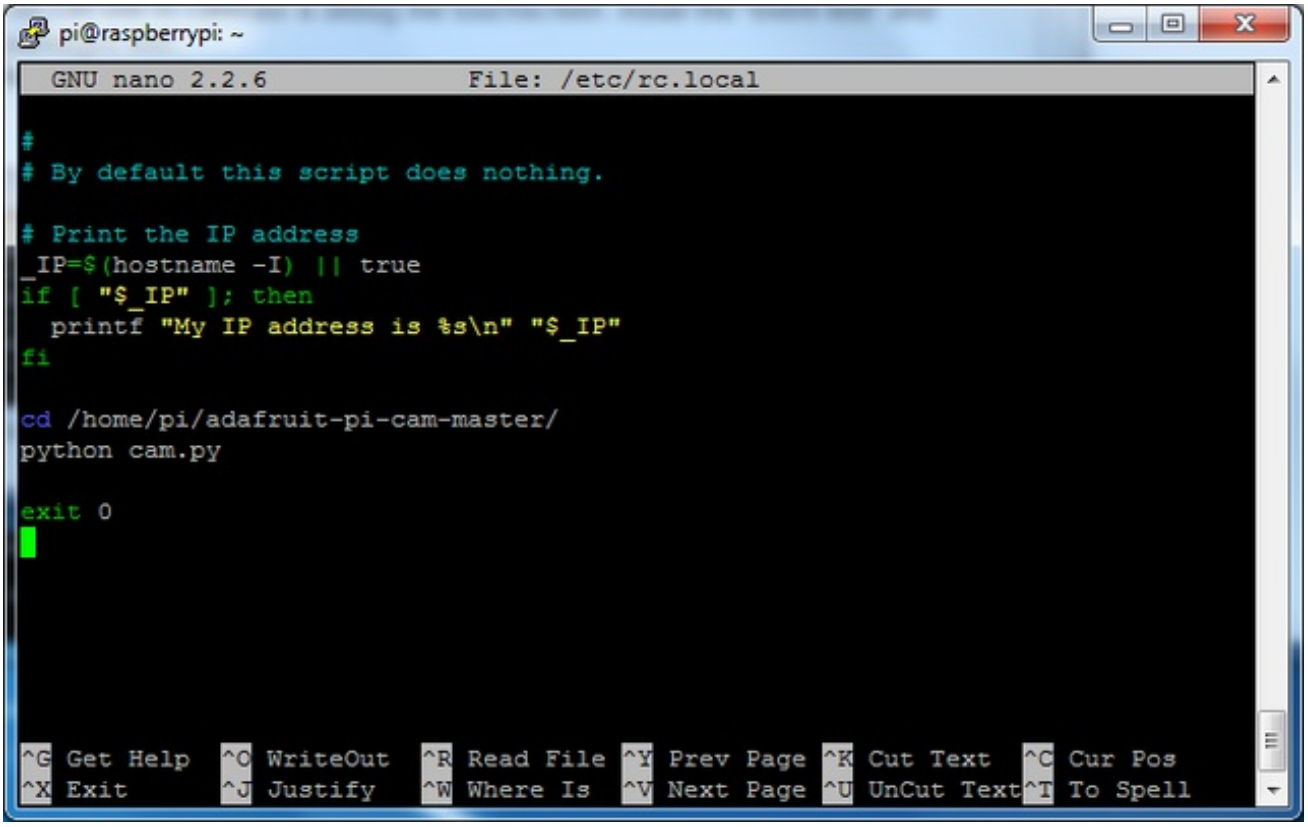
If this *doesn't* happen, an error message should give some sort of troubleshooting guidance; missing library or driver, etc. This is why we recommend working through the TFT tutorial first.

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

If you'd like to have the Pi in 'standalone' camera mode, start by turn on the console output to the PiTFT (see the PiTFT tutorial for this) and also turn on the power on/off button extra on one of the button pads so you can easily and safely turn the Pi off without logging in or connecting a keyboard.

Once youve got that working, have the Pi boot straight into the camera software by editing **/etc/rc.local** and adding the following lines before **exit 0**



Next time you reboot you should see the text console and then it will start the picam software. To shutdown, use the button you assigned to the power switch module

Using the Capacitive Touch PiTFT

For capacitive screens, before the cam.py script can be launched, you need to enter these additional four commands:

```
gpio -g mode 0 in
gpio -g mode 1 in
gpio -g mode 28 alt0
gpio -g mode 29 alt0
```

Then try launching the camera script:

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

If it works, then you can combine everything in the **/etc/rc.local** file (before the 'exit 0') to make the camera script launch at startup:

```
gpio -g mode 0 in
gpio -g mode 1 in
gpio -g mode 28 alt0
gpio -g mode 29 alt0
cd /home/pi/adafruit-pi-cam-master
python cam.py
```

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.

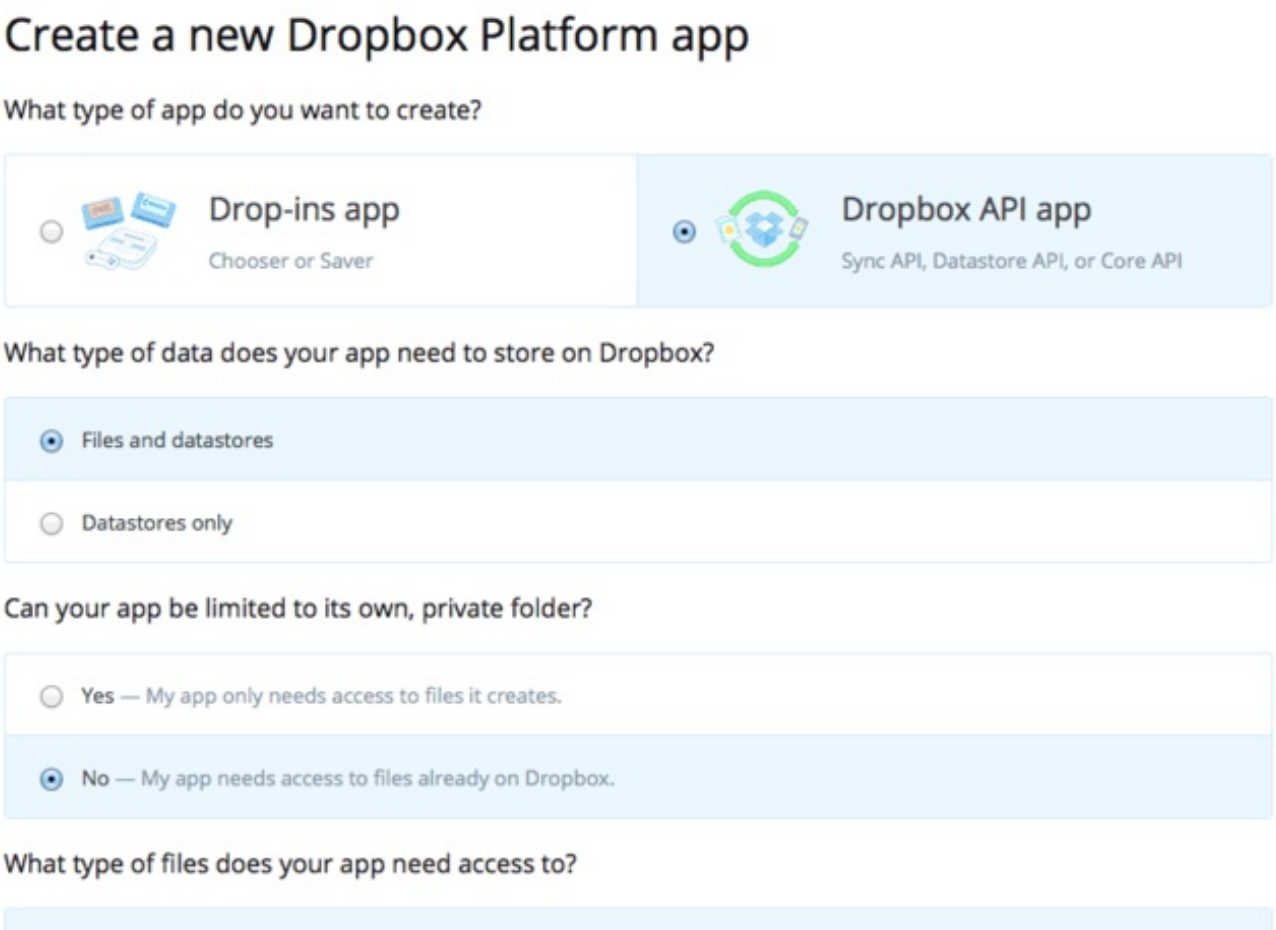
The following tutorial at RasPi.TV explains the process of setting up Dropbox and the Raspberry Pi to work together:

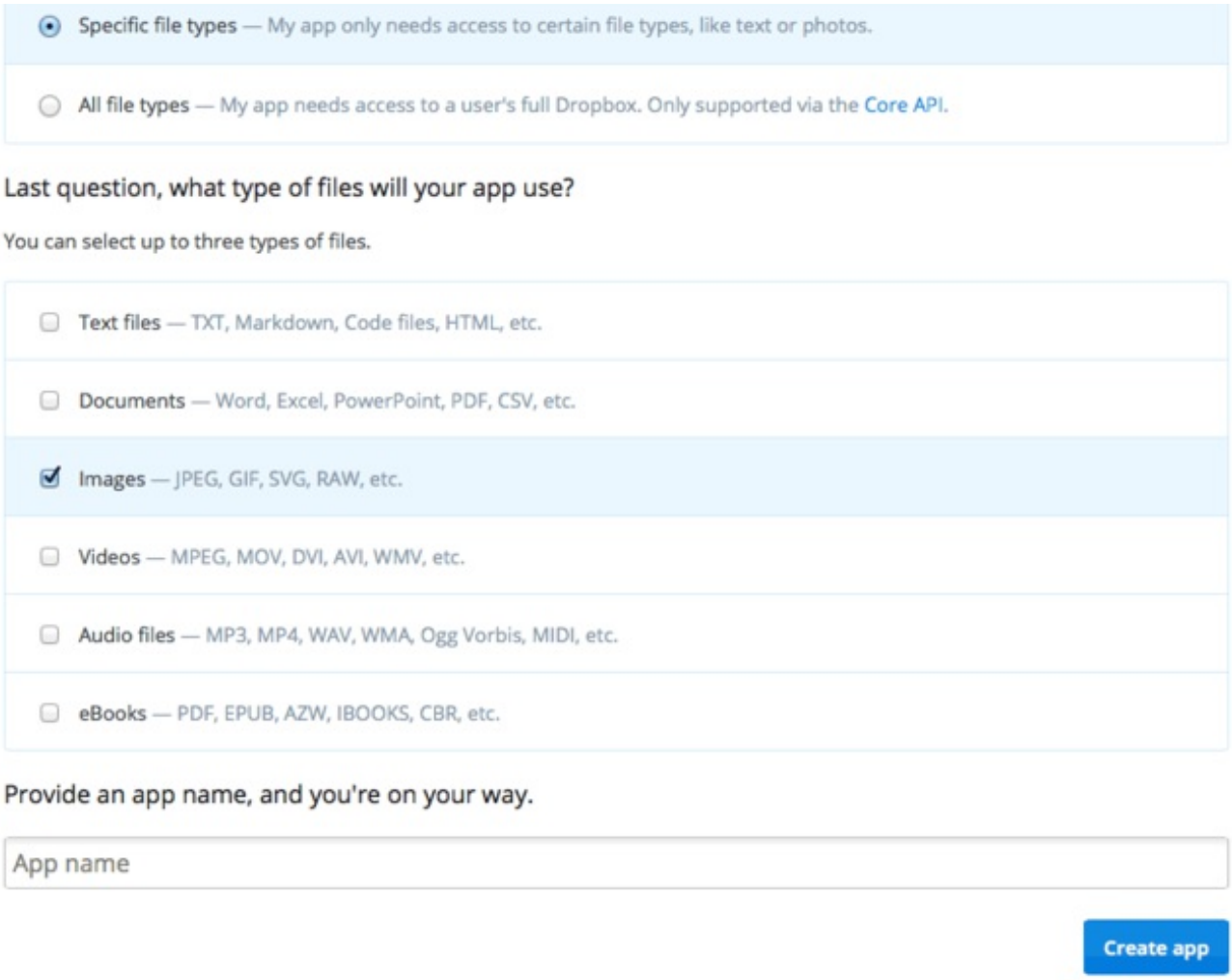
How to use Dropbox with Raspberry Pi (http://adafru.it/d56)

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.

These are the settings I used on the Dropbox developer site when creating a new app for the camera:





One more step:

- Test the Dropbox Uploader program on the Raspberry Pi by uploading a file (anything will do).

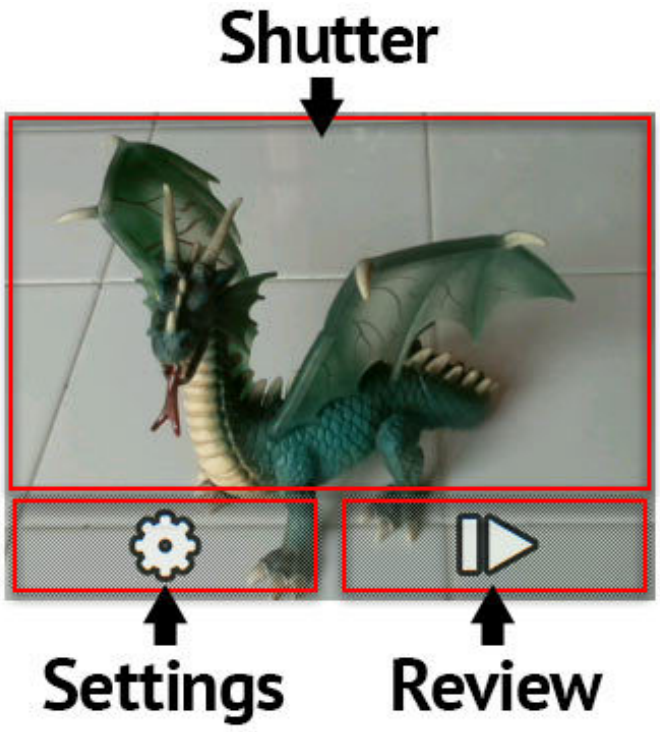
If this works, then next time you run the cam.py script you can go into *Settings* and select *Dropbox* for storage. Photos will be saved both in the "Photos" folder and uploaded immediately to Dropbox.

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.

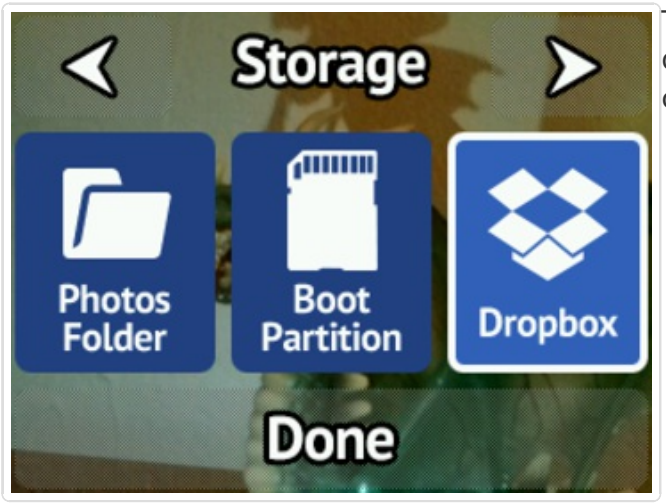
Upon startup, the camera program presents a live viewfinder and a couple of buttons. The majority of the screen itself functions as the shutter "button" — tap to take a still photo.

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").



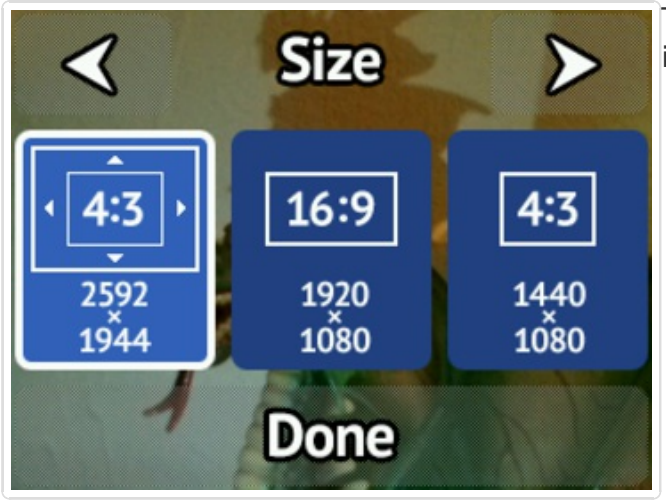
The *Settings* menu provides access to camera settings. This is not an exhaustive list of every feature possible with the Raspberry Pi camera, just a few essentials to get you started.

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 very 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.

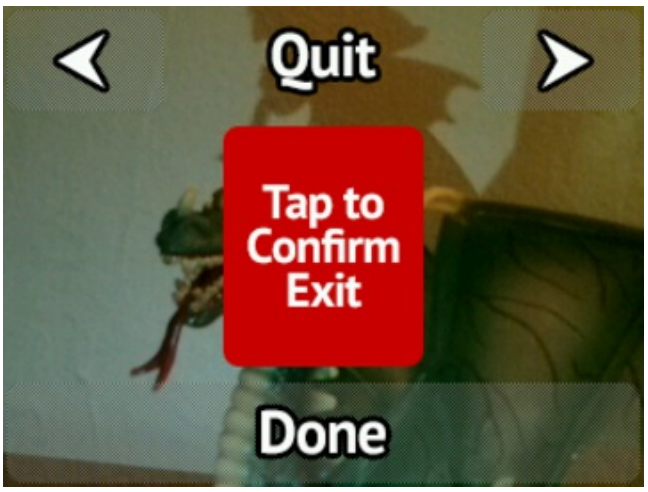
The *Effect* screen is where all the fun happens.

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.

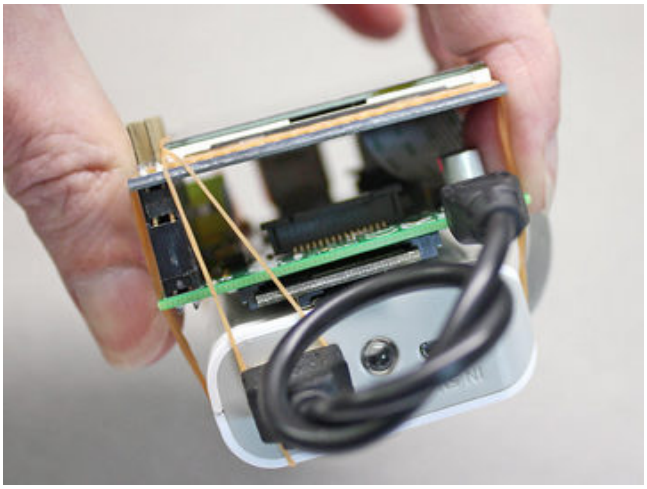
All the camera settings will be saved; next time you run the script, all prior settings will be as you left them.

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 the cam.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 the Enclosure

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

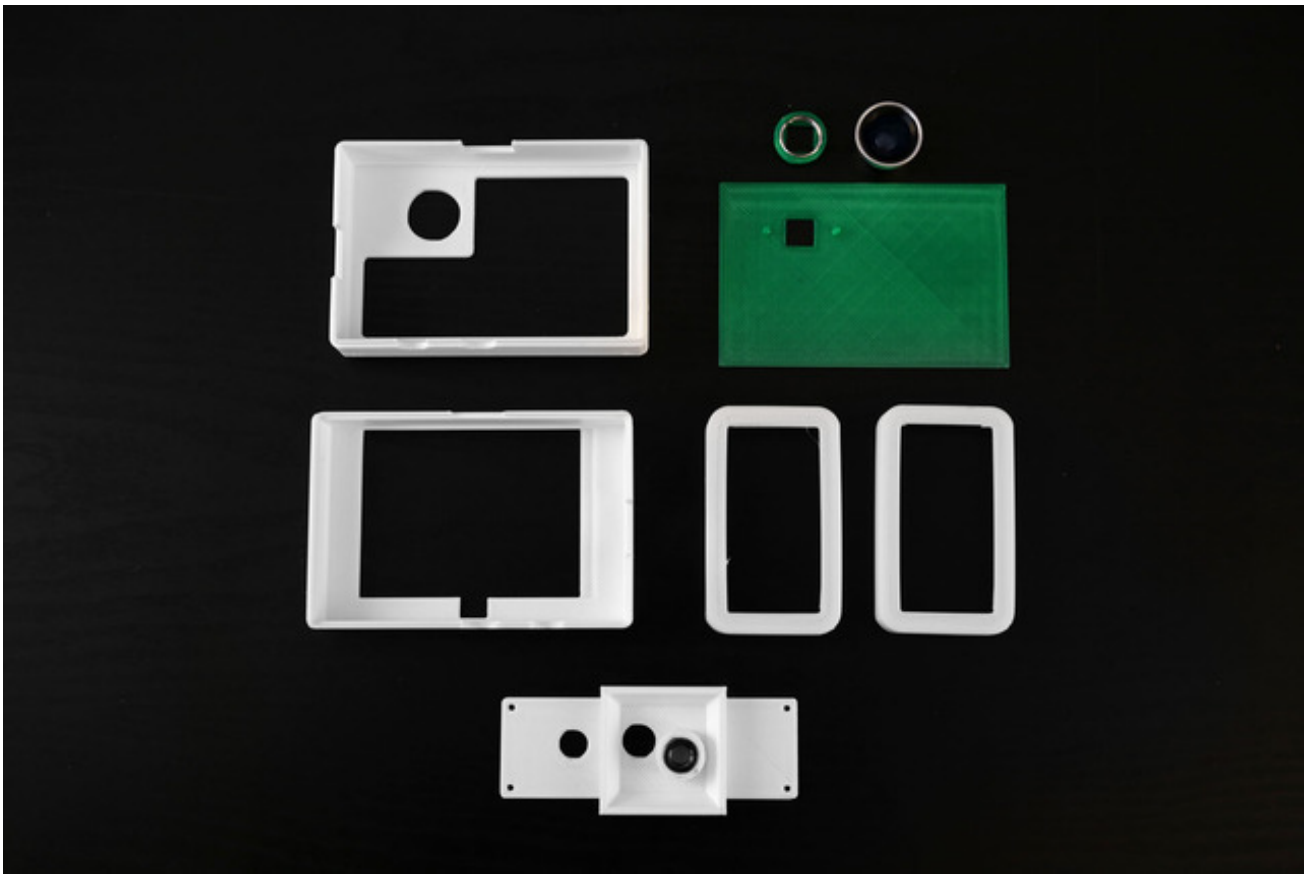
http://adafru.it/d9s

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

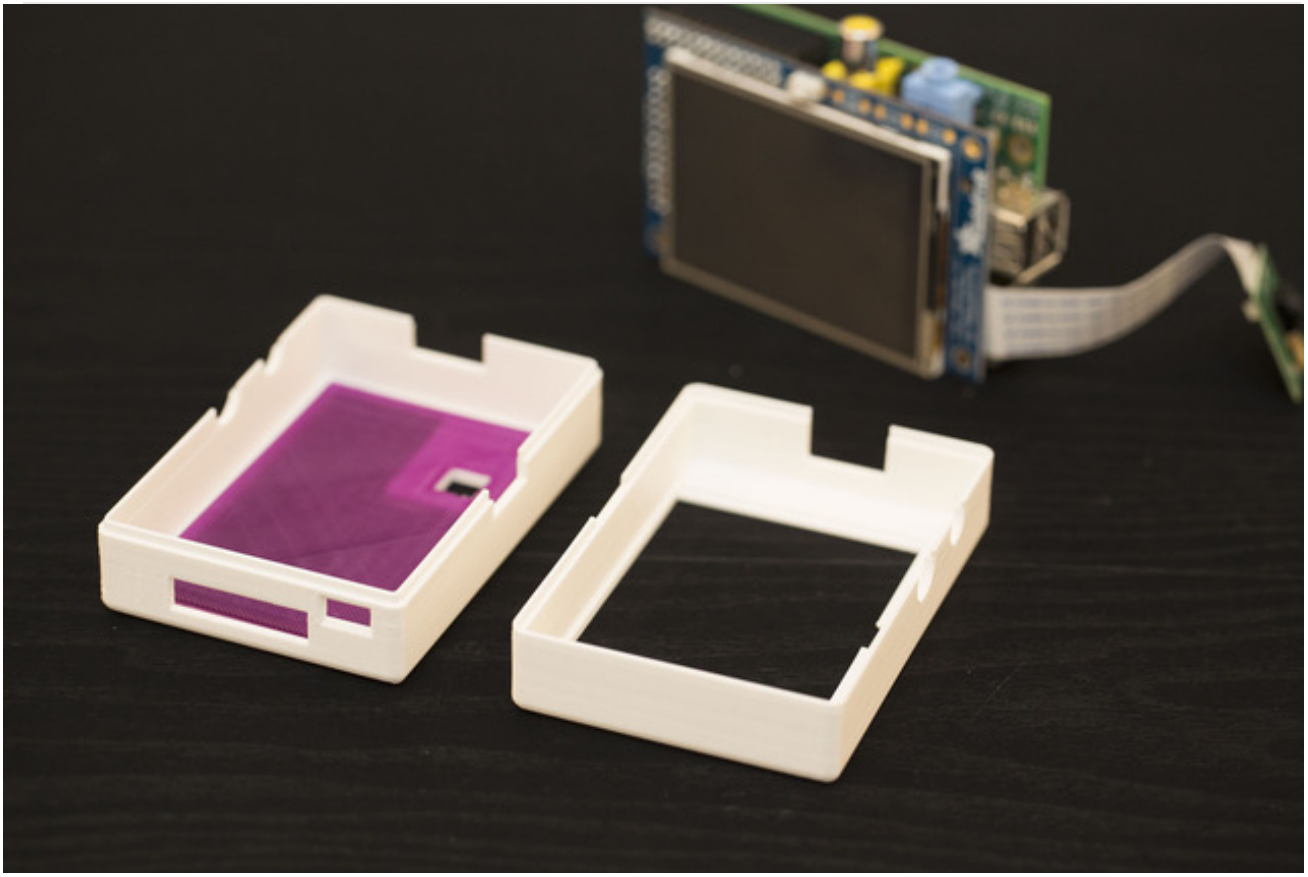
- **ptouch-tripod.stl** If you don't want a Tripod Mount
- **ptouch-band.stl** If you don't want Grippy edges
- **ptouch-camring.stl** If you don't want interchangeable lenses.

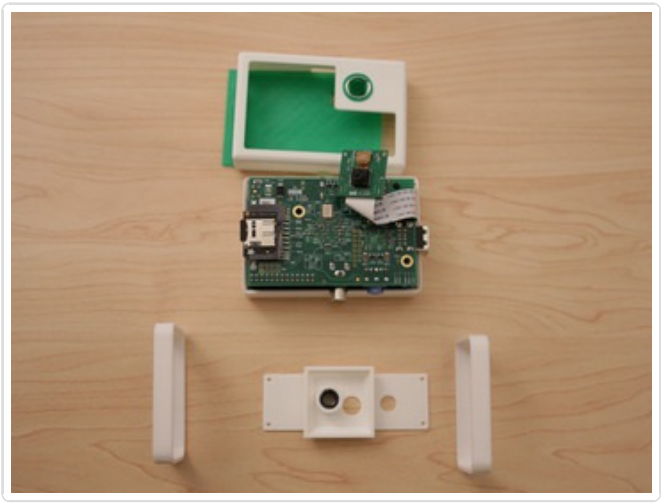


pitouch-top.stl about 75 minutes 14g	Houses the camera and Raspberry Pi. Snaps to **pitouch-bottom.**	PLA @230 2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-bottom.stl about 70 minutes 13g	Houses the touch screen and Raspberry P.	PLA @230 2 shells 15% Infill 2.0 Layer Height 90/150mm/s
pitouch-cover.stl about 30 minutes 6g	Fits inside **pitouch-top.stl**. Raspberry Pi Camera press-fits to cover.	PLA @230 2 shells 15% Infill 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
	Attaches to sides of pitouch-top.stl and	NinjaFlex @225

pitouch-band.stl about 25 minutes 5g	**pitouch-bottom.stl**. Adds grip and keeps **pitouch-tripod** mounted to **pitouch-top.stl** and **pitouch-bottom.stl**.	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

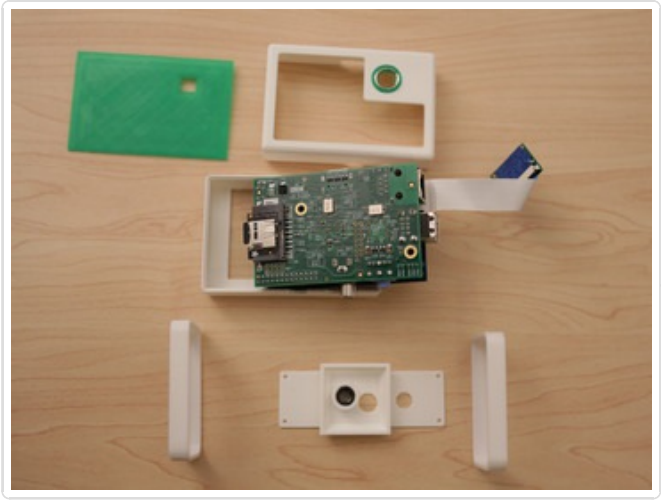




Mounting Pi Cam

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.

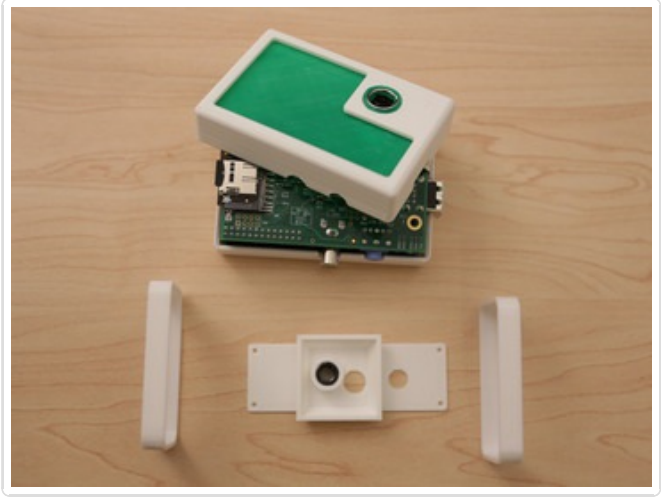
Position the camera with the camera hole in the pitouch-cover.stl part. Align it up so the nubs snap into the left and right mounting holes of the pi cam. Press fit the cover into the **pitouch-top** part.

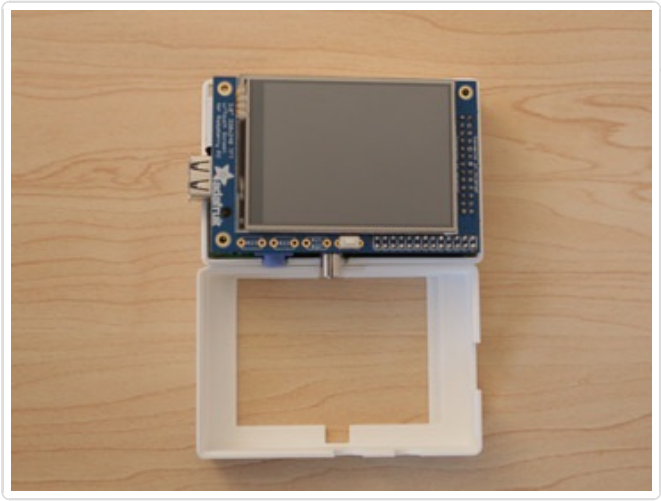


Position the **pitouch-top** part over the Raspberry Pi and angle it down so the SD card slot slips into the proper opening of the pit ouch-part.

Adding Lenses

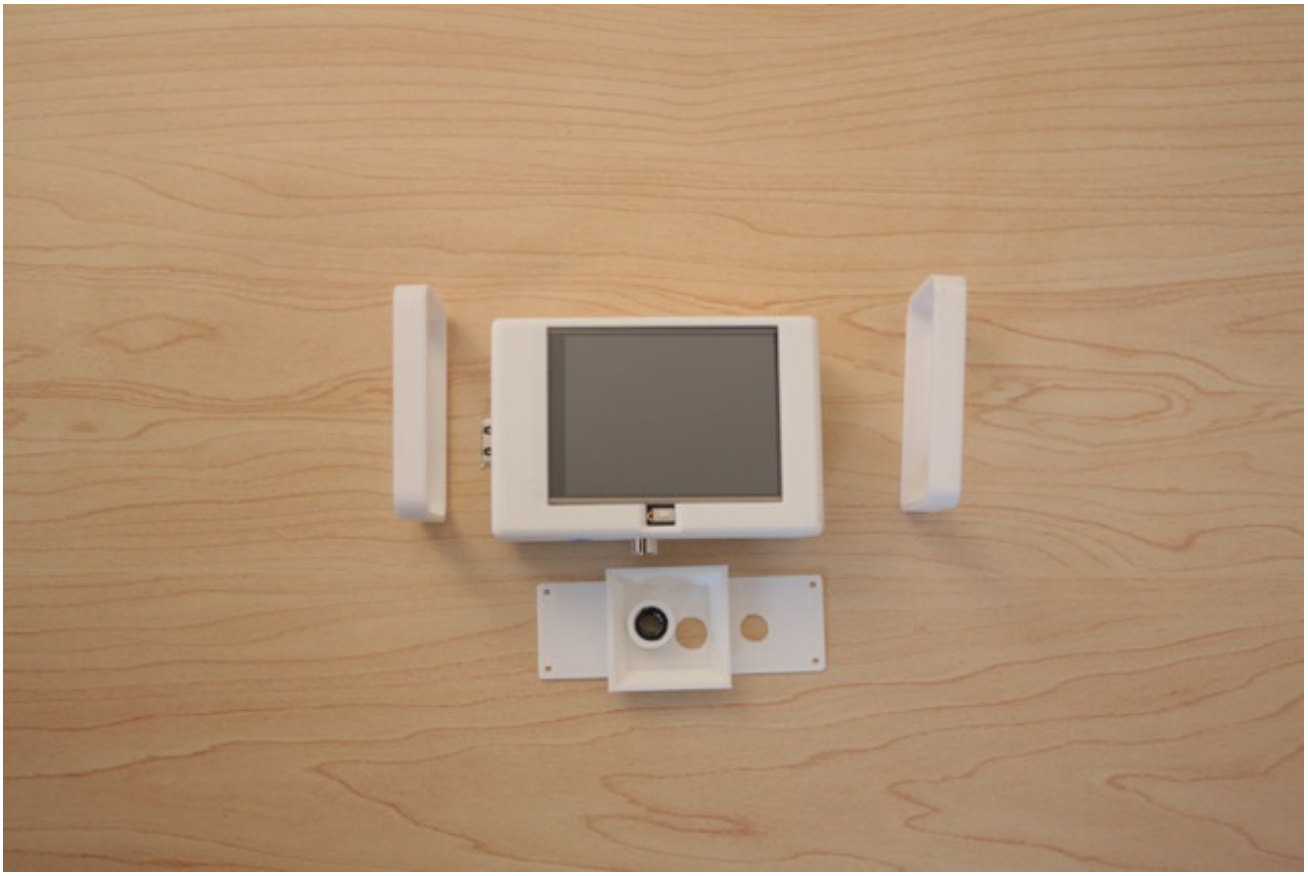
Snap the **pitouch-camring** part into the camera hole of the **pitouch-top** part. Insert a 12mm split ring into the **pitouch-camring**. The photojojo lens use magnets for mount and connects to the split ring.





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!