Instant Camera using Raspberry Pi and Thermal Printer

Created by Phillip Burgess

https://learn.adafruit.com/instant-camera-using-raspberry-pi-and-thermal-printer

Last updated on 2023-09-21 01:44:56 PM EDT

Table of Contents

Overview

This guide is marked as DISCONTINUED — thermal printers and accessories are no longer carried by Adafruit, and some of the APIs and drivers referenced here are no longer available or compatible. Code will not be updated and guide feedback will not be reviewed. The information remains online because a few elements may be of use for anyone who still has this hardware, but will still need to DIY their own code.

"Instant photography" with Polaroid cameras was a thing up through the 1990s until ubiquitous digital photography took hold…though, like vinyl music, the medium has since made a nostalgic resurgence.

In this project, we'll replace chemical film with more modern electronic parts: a Raspberry Pi computer and camera paired with a diminutive thermal printer, all working off a battery. Press a button, get a print!

This camera-and-thermal-printer combination is not a new idea…Nintendo released their Game Boy Camera and Printer as mass-market devices in 1998. What's exciting is that we no longer need the resources of a giant corporation to create something similar…inexpensive computer power and open source software make infinite customization possible!

Parts from Adafruit:

- Any model Raspberry Pi with a camera connector (i.e. all but Pi Zero). The Model A+ (http://adafru.it/2266) is an excellent choice because it's small and power-efficient, but if you have a different model on-hand this'll work all the same (though might need your own case). The Raspberry Pi 3 only works with our USB receipt printer so it might not be the best option.
- 2GB or larger microSD card (or full-size SD for older Raspberry Pi boards).

- Any model Raspberry Pi Camera board; visible light or NoIR (infrared), original 5 megapixel or newer 8 MP models. This project does not require high resolution, so if you've upgraded another project to the latest camera, it's an excellent opportunity to repurpose the "classic" unit.
- Any model Adafruit Thermal Receipt printer. We'll use the "Nano TTL (http://adafru.it/2752)" model because it's compact, but any of the other varieties can work as well.
- Thermal Paper Roll(s). The Nano printer requires a special nano roll (http://adafru.it/2755). There are corresponding fitting rolls available for the Mini and full-size receipt printers.
- 4xAA Battery Holder with On/Off Switch (http://adafru.it/830). Or you might be able to adapt a different 5-ish Volt power supply...it requires a couple Amps of current, and most USB battery packs didn't quite cut it.
- Perma Proto HAT — No EEPROM (http://adafru.it/2310). This is used for connecting buttons and power to the Raspberry Pi GPIO header...though if you're resourceful, any Pi Cobbler or even some M-F jumper wires could be adapted to the task.
- 16mm Panel-Mount Momentary Push Button or similar, two required. Various (http://adafru.it/1503) interesting (http://adafru.it/1445) colors (http://adafru.it/1502) are (http://adafru.it/1504) available (http://adafru.it/1505).
- Latest version of Raspbian Jessie Lite () for Raspberry Pi web site. No not use the full version, it's enormous...the Lite version is sufficient for this project.

Parts Not From Adafruit:

- 4 (four) Panasonic Eneloop rechargeable NiMH batteries. Yes, this is actually important, not a shill...we tested with other cells and they lacked the same "oomph." Additionally, the Eneloops hold a charge much longer when stored.
- NiMH battery charger.

You will also need the usual electronics project items: soldering iron and related paraphernalia, some bits of wire, etc. Optionally, you can add an LED (any color) and resistor (100 Ohm or thereabouts) as a status indicator.

Some prior Raspberry Pi experience is assumed — downloading the OS, writing an SD card, basic system and network configuration, etc. You can search the Adafruit Learning System for related guides if any of this is unfamiliar.

During setup, you will temporarily need a monitor, keyboard and network connection. The Model A+ may require using a USB hub and a wireless adapter...if you already

have a working Raspberry Pi system that's networkable, it may be easier to borrow that for the software setup, then move the card over to the A+ board.

Creating an enclosure for your camera is a DIY project. We'll upcycle the box the thermal printer comes in, but you might want to create something fancier than that... maybe 3D printed, or even fit into an old camera body.

Read through the full guide before commiting to any purchases. You might already have some parts on-hand... or there may be some "gotchas" that require specific components.

System Setup

The Raspberry Pi 3 is not currently compatible with TTL printers. Our USB-capable "Tiny" printer can work with this board...or substitute any other model of Raspberry Pi (e.g. A+) instead. Pi 3 is a power-hungry board anyway and not the best choice this battery-powered project.

If you're using a TTL printer (not USB), DO NOT connect it to the Raspberry Pi yet! It will spit paper like mad. Some system configuration is required first...

Basic Setup

Download the latest version of Raspbian Jessie Lite () if you haven't already. You do not need the regular full version; "Lite" is adequate for this project's needs and can fit on a 2GB card with room to spare. You'll also need a keyboard and monitor attached,

and a network connection…if using a Pi Model A+, this may require a USB hub and a WiFi or Ethernet adapter…or, if you have another working Pi system, it's often easier to borrow that for setup, then move the working card over to the A+ board.

Write the OS to an 2GB or larger SD card, insert in the Raspberry Pi and power it up. After a minute or so you'll get a login prompt. Log in with the usual default pi/raspberry user and password, then run the raspi-config utility:

```
sudo raspi-config
```

The following options are required:

- Expand Filesystem.
- Enable Camera.
- Under "Advanced Options," select "Serial" and disable the serial console.

The following are optional but recommended:

- Under "Internationalisation Options," configure the keyboard (and optionally the locale and time zone) to match your needs. If you're getting unexpected output from the keyboard, this is why.
- Change User Password because everybody knows the default.
- Advanced Options: Change Hostname to distinguish this system from other Raspberry Pi systems on the network.
- Advanced Options: Enable SSH to allow remote administration via network.
- Advanced Options: Disable Overscan gives a little more working space if using an HDMI monitor.

When you're done, tab over to the button and reboot the system when prompted. Log in as the "pi" user again, using the password you assigned above (or the default "raspberry" if you opted not to change it).

Networking

This is most easily done with a wired Ethernet connection — usually just a matter of plug it in & go. The Model A+ board doesn't have an Ethernet port, which is why we recommend doing the setup on a more capable Pi system, then moving the card over when finished. Alternately, a USB hub and WiFi adapter can work.

Connecting to wireless networks usually involves editing either /etc/network/ interfaces or /etc/wpa_supplicant/wpa_supplicant.conf with your wireless network name and credentials. Beyond the scope of this guide, it's explained in other guides here, or Googling can quickly turn up some reference material.

The network is only needed during setup…once the camera's tested and working, it's all self-contained.

Install Software

First we'll install printer support (CUPS — the Common UNIX Printing System) and some related development tools…

```
sudo apt-get update
sudo apt-get install git cups wiringpi build-essential libcups2-dev libcupsimage2-
dev
```

Then install the raster filter for CUPS. This processes bitmap images into the thermal printer's native format…

```
cd
git clone https://github.com/adafruit/zj-58
cd zj-58
make
sudo ./install
```

Your thermal printer may have arrived with a test page in the box or the paper bay. If not, or if you threw that away, you can generate a new one by installing a roll of paper and holding the feed button (on printers that have one) while connecting power, or tapping the button on the back of the "Nano" printer or the "Printer Guts."

Look for the baud rate that's printed near the bottom of the page. This is typically either 9600 or 19200 baud. This is important…you'll need to know the correct value for your printer.
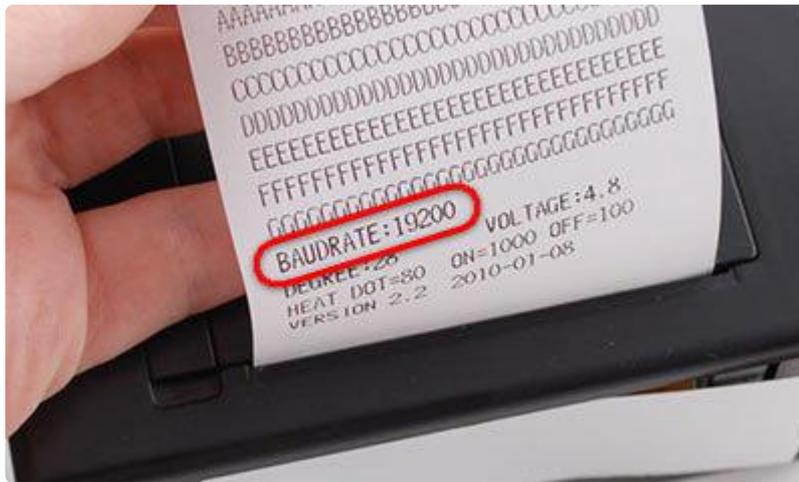
The printer doesn't need to be connected yet. We can prepare the system the same regardless.

To add the printer to the CUPS system and set it as the default, we'll be typing two lines similar to the following (but not necessaryily identical…read on)…

```
sudo lpadmin -p ZJ-58 -E -v serial:/dev/ttyAMA0?baud=9600 -m zjiang/ZJ-58.ppd
sudo lpoptions -d ZJ-58
```

On the first line, change the "baud" value to 9600 or 19200 as required for your printer.

For a USB receipt printer, change the device name to /dev/ttyUSB0

For all other (TTL) printers, use /dev/ttyAMA0 for the device name.

The rest of the line should be typed exactly as it appears above. Likewise for the second line, which needs no changes.

Camera Script

The code that handles the shutter button and moves images from the Pi camera to the printer is included with our version of the CUPS filter software. You'll find it in /home/pi/zj-58/extras/camera.sh (unless you're using a different account or home directory, or downloaded the zj-58 repository elsewhere…the important bit is, look in the "extras" directory for this script.

It's all written as a Bash shell script, no Python or anything else required:

```
#!/bin/bash
```

```
SHUTTER=16
HALT=21
LED=5

# Initialize GPIO states
gpio -g mode  $SHUTTER up
gpio -g mode  $HALT    up
gpio -g mode  $LED     out

# Flash LED on startup to indicate ready state
for i in `seq 1 5`;
do
    gpio -g write $LED 1
    sleep 0.2
    gpio -g write $LED 0
    sleep 0.2
done

while :
do
    # Check for shutter button
    if [ $(gpio -g read $SHUTTER) -eq 0 ]; then
        gpio -g write $LED 1
        raspistill -n -t 200 -w 512 -h 384 -o - | lp

        sleep 1
        # Wait for user to release button before resuming
        while [ $(gpio -g read $SHUTTER) -eq 0 ]; do continue; done
        gpio -g write $LED 0
    fi

    # Check for halt button
    if [ $(gpio -g read $HALT) -eq 0 ]; then
        # Must be held for 2+ seconds before shutdown is run...
        starttime=$(date +%s)
        while [ $(gpio -g read $HALT) -eq 0 ]; do
            if [ $(($(date +%s)-starttime)) -ge 2 ]; then
                gpio -g write $LED 1
                shutdown -h now
            fi
        done
    fi
done
```

A few variables at the top of the script may need editing depending how you assemble your camera…we'll return to this on the "Connections" page.

Configure for Auto-Start

Let's make the camera script launch automatically when the system boots…

```
sudo nano /etc/rc.local
```

Before the final "exit 0" line, insert:

```
sh /home/pi/zj-58/extras/camera.sh
```

It should look something like this:

```
  GNU nano 2.2.6            File: /etc/rc.local              Modified

#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

sh /home/pi/zj-58/extras/camera.sh
exit 0




^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is   ^V Next Page  ^U UnCut Text ^T To Spell
```

If you've located the camera.sh script somewhere else, enter the full path here.

Save the changes to the file and exit the editor.

Once the system's fully configured, you may want to back up the SD card. Like any battery-operated Pi project, there's a small possibility of the filesystem getting corrupted. Many utilities that write SD card images can also read a card to create a backup.

Connections

Camera

Connecting the camera to the Raspberry Pi is explained (with video) on the Raspberry Pi web site (). Note that there is a specific orientation for the cable…the blue tape at each end must face a particular side or it won't work.

Mentioned on the System Setup page, but for posterity: the Raspberry Pi must be configured to access the camera using the raspi-config utility.

You can test the camera using the raspistill command. If it refuses it work, it's usually one of the above two things; cable's installed wrong, or camera's not enabled in raspi-config.

The stock 150mm camera cable should be sufficient for most projects, but if you have an unusual form-factor in mind (e.g. fitting into an existing camera housing), alternate flex cables are available (http://adafru.it/2087).
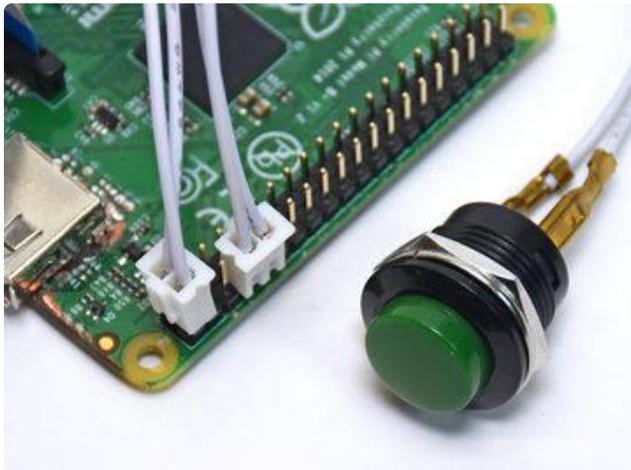
Buttons

There are two buttons on this camera project. The first, as you'd expect, is the shutter release…you press it to take a photo.

The other button initiates an orderly system shutdown before you switch off the power. This button needs to be held for at least two seconds, so it won't trigger accidentally. Once activated, allow 20 seconds or so for the shutdown to complete before switching off power. Don't switch off without a proper shutdown…Linux systems hate that…and it may even corrupt the filesystem on the SD card (then you'd have to start over).
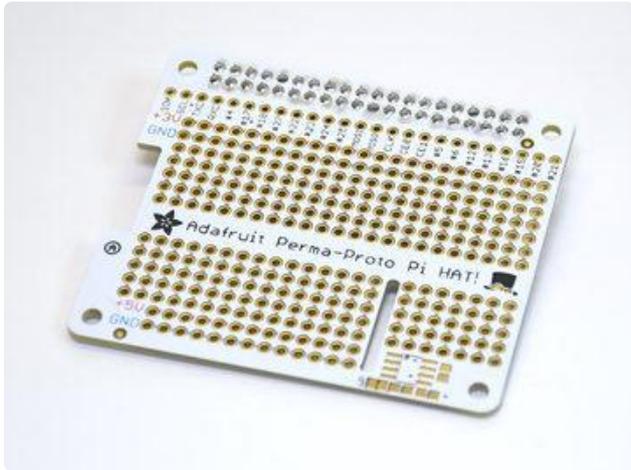
A Perma-Proto Pi HAT (http://adafru.it/2310) is recommend for the most durable connections, and also makes it easier to distribute power and ground to multiple points.

If you're building this as just a temporary project, to be dismantled and re-used for other things later, button quick-connects (http://adafru.it/1152) can be used instead. With a little planning, each plug fits atop a GPIO pin and adjacent ground point.

If using the Perma-Proto approach, do not solder everything together yet! Wires can be soldered at the button end, but leave the other end unconnected until all the parts are fitted into the case.

The wires should be a little longer than the width of your planned case, so they can snake around as needed.

LED

The LED is optional but provides useful status information…lets you know a picture has been taken or that the system has shut down.

This doesn't need to be anything fancy…you might already have a suitable LED in your parts stash. Not a NeoPixel, just a plain ol' through-hole LED, any color. Add a resistor in-line to limit current…minimum 75 Ohms, but a little higher is okay (100 Ohms, 220, it's all good).

Add wire leads long enough to reach across the camera. Insulate any connections with heat-shrink tube.

For a temporary build, you can clip a F-F jumper wire in half, strip the ends, solder this to your leads and press onto pins on the GPIO header.

If using the Perma-Proto board, do not solder the LED wires to the board yet.

One leg of each button and the LED cathode (shorter or "minus" lead) will connect to a ground pin on the GPIO header...there are several scattered around.

The other leg of each button, and the LED anode (longer or "plus" lead) will connect to GPIO pins. These are configurable in the camera.sh script. The default values are:

```
SHUTTER=16
HALT=21
LED=5
```

The default pin assignments were chosen to make it easy to use quick-connects to straddle a GPIO pin and adjacent ground. However, the pins used above are only present on Raspberry Pi boards with the 40-pin GPIO header..."classic" boards (like the original Model B) with a 26-pin header will require selecting different pins.

The defaults were also chosen to steer clear of some pins that might be useful for adding features in the future...
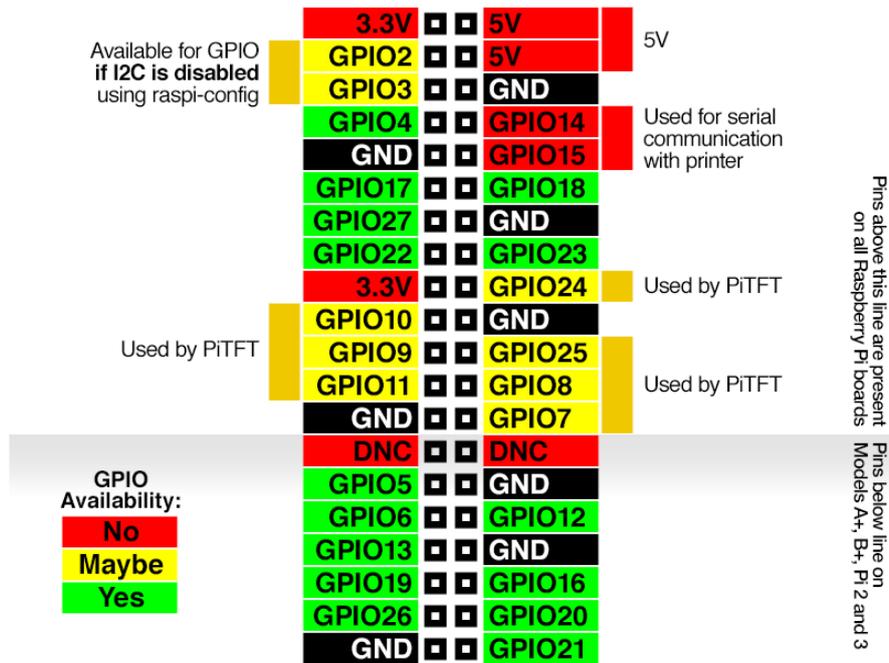
Although this project does not use a PiTFT display, it seems like an obvious direction that some will want to take...so try to avoid using those pins for the buttons or LED.

Printer

The serial connection on the Raspberry Pi can not be reassigned to different pins — it's always on GPIO14 (TX) and GPIO15 (RX). (Also sometimes labeled TXD and RXD.)

When making the connection between Pi and printer, these wires must be crossed. TX from the Pi connects to RX on the printer, and RX to TX. (If using the USB mini printer, this step is bypassed, just use a USB cable.)

The standard and "Nano" thermal receipt printers, plus the "Printer Guts" model, all have their pins labeled. On the "mini" printer, if using the TTL interface instead of USB, green wire is RX, blue is TX (but remember to cross these when making the connections to Pi TX and RX).

Power

Power from the battery pack needs to be split to both the Raspberry Pi and the printer.

If using the Perma-Proto HAT, this is super easy: use the +5V and GND rails on the board; battery pack and printer can both connect there, and these also go to the appropriate points on the GPIO header.

If wiring "manually," you may need to splice two ways off the battery pack to both the Pi and printer.

Four Eneloop NiMH cells nominally provide 4.8 Volts to the system (sometimes a little more, sometimes less)…this is close enough to 5V to keep the Pi happy…and provide adequate current to run the printer. Do not install alkaline cells…this could potentially damage the Pi or anything connected to its USB port, and probably won't have enough current for the printer.

Our 4X AA battery holder has an On/Off switch. If using a different battery holder without a switch, add your own switch in-line on the + wire.

Assemble and Customize

We'll use the box the printer came in as a DIY enclosure!

If you already tossed the box, or if you need a different shape (for example, using a different Pi board or a different battery), not to worry…you can improvise from almost anything. Maybe a slightly larger box that an Adafruit order came in, or some kind of Tupperware-like plastic container. Improvise, go nuts, it's not rocket science.
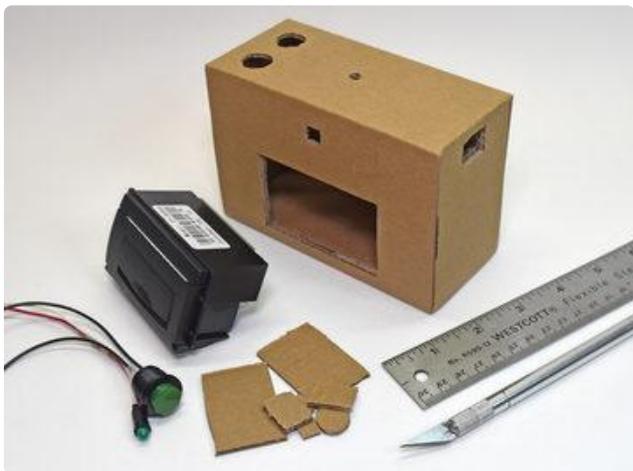
Measuring some parts, or just tracing around others directly as a template, use a pencil to mark where the components will go, then cut out these sections with a hobby knife.

I chose to put the thermal printer at the lower front center, camera board at upper center. The Pi and battery pack will then fit to either side.

Nothing should be soldered to the Perma-Proto board yet. If you've made these connections already, un-solder the wires from the board. Re-do these after everything's in position.
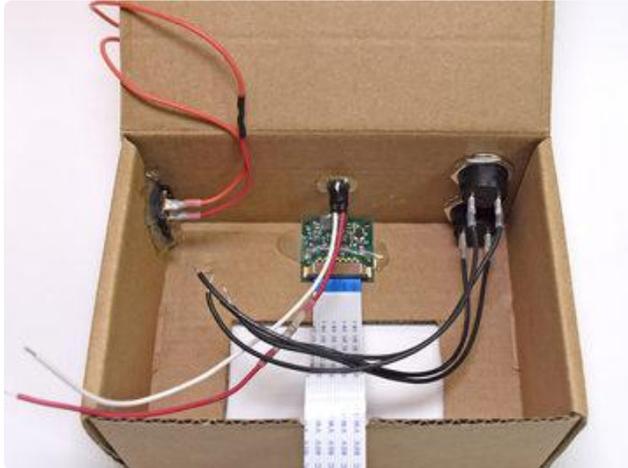
Starting from the outside, feed the wires through each hole and press the buttons and LED into place. The buttons include washers and nuts to hold them in place on the inside, while the other parts can be held with hot glue.

The camera board is pressed in place from behind and held down with small dabs of hot glue…don't go crazy on this part, it may be sensitive to heat.

Since I didn't have the same 4X AA battery holder on hand, I added my own power switch, which you can see at the top-left of the box.
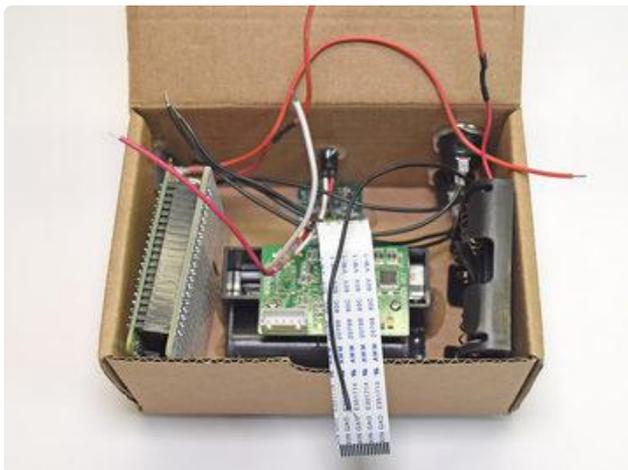
Experiment with parts placement inside the box. I'd planned to have the battery on left, Pi on right, but there wasn't adequate clearance under the buttons. Fortunately I'd connected long wires to everything and it was no problem to swap these positions!

The 4X AA battery holder with switch is a little larger than the battery holder I'm using here. Instead of the right side, it might sit flat against the back of the printer instead.

There's some "electronics origami" as parts are connected and fit into place...
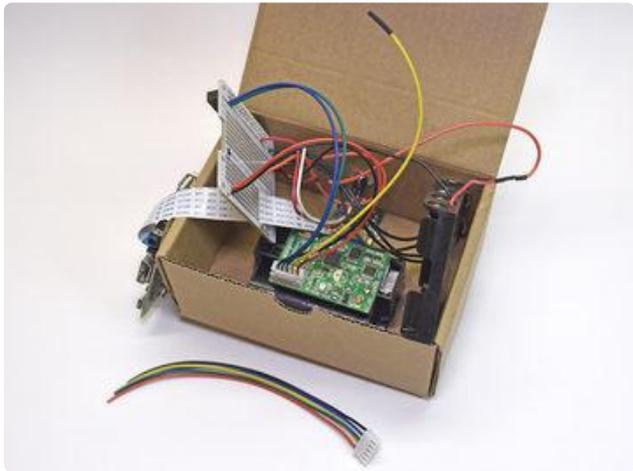
Perma-Proto HAT is removed from the Pi for the time being.

The camera cable needs to snake through the slot on the Perma-Proto HAT, then into the Raspberry Pi.

The printer TTL cable was cut in half, TX and RX connected to the Perma-Proto (remember, cross these to RX and TX on the Pi), red and black to the +5V and GND rails. The end of the unused wire (DTR) was insulated so as not to cause mayhem.
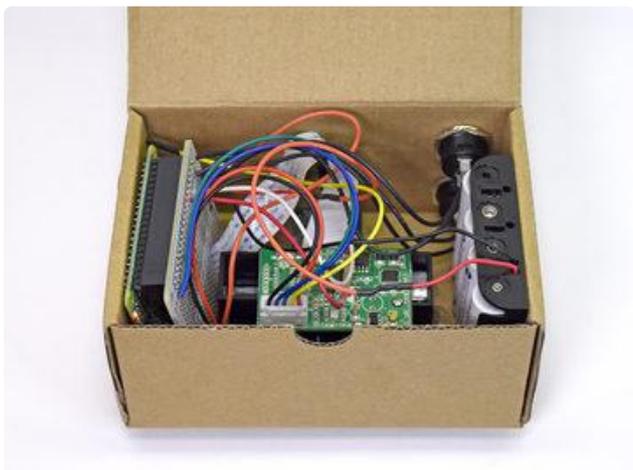
Once everything's connected, then the Perma-Proto HAT can be reattached to the Pi.

Find a stable position for the Pi and battery pack. The former can be held in place with some foam tape or hot glue. The battery pack should be able to be lifted out, so the batteries can be replaced.

Try powering it on. The red LED (power indicator) on the Raspberry Pi should light, then the green LED (SD card access) should flicker.

After 30 seconds or so, the status LED (on GPIO pin 5) should flash a few times, indicating the system is ready. Aim the camera at something and try the shutter button...

Success!

Troubleshooting

No red light when I switch it on!

- Check that the batteries are fully charged and are installed in the correct orientation.
- Check that the red and black wires from the battery pack go to the +5V and GND rails on the Perma-Proto board, or (if wired manually) are split to the equivalent points on both the Raspberry Pi and printer.

No green light!

- Is the SD card installed and properly seated?
- Disconnect the Perma-Proto board and try booting the Pi "bare." If this works, you might have a short or wrong connection somewhere in your wiring.

The printer is spitting paper like mad!

- Disconnect the TTL cable from the printer. Connect a monitor and keyboard to the Raspberry Pi, run the raspi-config utility and disable the serial console.

System boots, but status LED never flashes.

- Confirm that the + (longer) leg of the LED is connected to GPIO5 (or whatever pin you've configured in the script), and the other leg is connected to a ground pin.

Doesn't print!

- The camera might be failing to capture a photo. Confirm that the camera ribbon cable is installed in the correct orientation and is firmly seated in both the Pi and camera board connectors. Remove the Pi from the box, connect a monitor and keyboard, run the raspi-config utility and make sure the camera is enabled.
- Make sure that TX and RX are crossed between the Raspberry Pi and printer. Pi TX should go to printer RX, while Pi RX should connect to printer TX.
- Review the steps on the "System Setup" page. Make sure the CUPS software is installed and that you've added a line to /etc/rc.local to run the camera script automatically at startup.

Prints, but is very light.

- Part of this may be the lighting, especially if shooting outdoors.
- Really dark printing may require more current than the batteries can provide. See notes below about modifications you can make for alternate power supplies (though they won't fit inside this same case).

I used the leftover scrap of cardboard from the printer cutout to fashion a little viewfinder. This is more stylistic than functional...the actual field of view is much wider...but gives people a little hint how to point and work the camera.

You could also use stickers or colored markers to spruce it up...draw a "lens" on the front, or give it a Polaroid rainbow logo. Or just draw some dragons at random, they make everything better.

Custom Enclosures

This project is ripe for all kinds of hacking...especially custom enclosures, perhaps 3D-printed or laser-cut, or retrofitted into an old Polaroid camera.

If you do make a custom enclosure, note that the printer is very orientation sensitive and should only go one of two ways:

- Paper feeding out the top.
- Paper feeding out the front, with the slot nearest the top (not bottom).

Any other orientation, the paper roll will wedge in place and the printer will jam!

Portrait vs Landscape

You're not required to install the camera in the box in the traditional "landscape" orientation…turning the camera board "sideways" is totally fair game, if you want a camera that's optimized for portrait photography without having to turn it. As explained above, the printer component should be kept horizontal, else the paper will jam.

Alternate Power Sources

The 4 Eneloop batteries provide sufficient voltage and current to run the Pi and thermal printer, though the output may be a bit light for some users' tastes. Other power sources are possible but increase cost and complexity, and may require some trial and error…

- A 5 Volt UBEC (http://adafru.it/1385) can provide ample current (3A) by stepping down a higher-voltage battery, such as a bank of 8 NiMH cells, or a 7.4V camcorder battery (we have holders for some Canon, Sony and Panasonic varieties in the shop). Camcorder batteries are a bit expensive to buy just for this project, but might make sense if you already have some around.
- A USB power bank may or may not actually work…it depends, some can't sustain sufficient output current and prints will be extremely light or paper won't feed properly. Sometimes the culprit isn't the power bank, but the USB cable, with tiny conductors that can't deliver much current.

Diffusion Dithering

I like the "ordered" dithering pattern CUPS uses by default…but if you'd prefer diffusion dithering instead, it's a simple change.

First, install ImageMagick:

```
sudo apt-get install imagemagick
```

Then modify the camera.sh script as follows…the "raspistill" line should be changed to:

```
raspistill -n -t 200 -w 512 -h 384 -o - | convert - -remap pattern:gray50 png:- | lp
```

You'll need to reboot the system or restart the camera script for this to take effect.

Not quite the look you're after? ImageMagick has a ton of image conversion and filtering options! ()

Adding More Controls or Other Features

If you're the hacking type and would like to add additional buttons, sensor feaures or control a flash…or even a PiTFT viewfinder…here's a map of the GPIO header again:
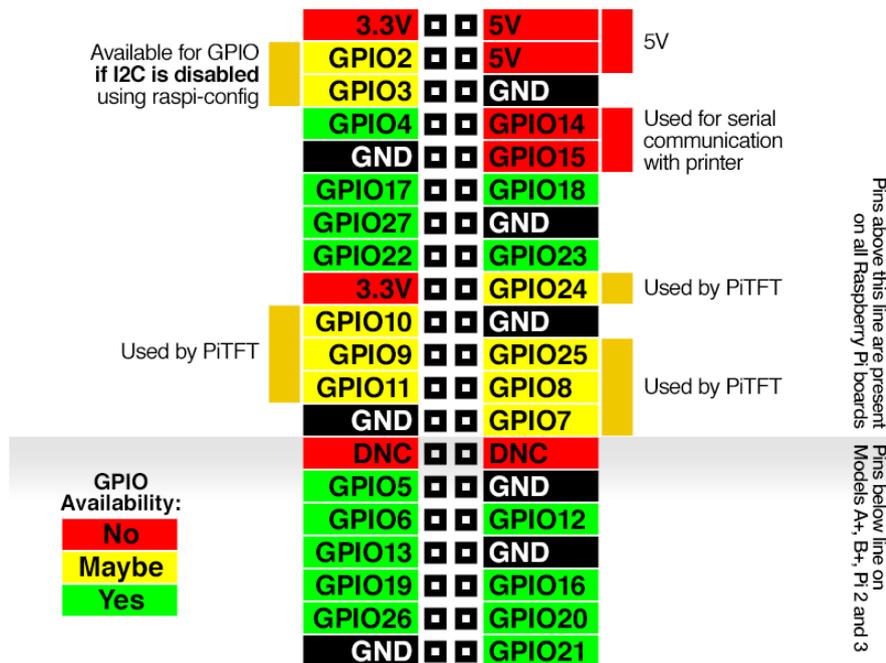
Any of the green GPIO pins are fair game as 3.3V logic inputs or outputs. If you plan to use a TFT display or maybe add one in later, steer clear of the yellow pins…the PiTFT displays require exclusive use of these. GPIO pins 2 and 3 may be available if I2C is disabled using raspi-config…but some sensors or devices may require using I2C.

As presented, our example script does not support a PiTFT viewfinder...this will require your own custom code, perhaps written in Python.