



Raspberry Pi Photo Booth Projector

 September 2, 2016  [Featured](#), [Raspberry Pi](#)

Intro

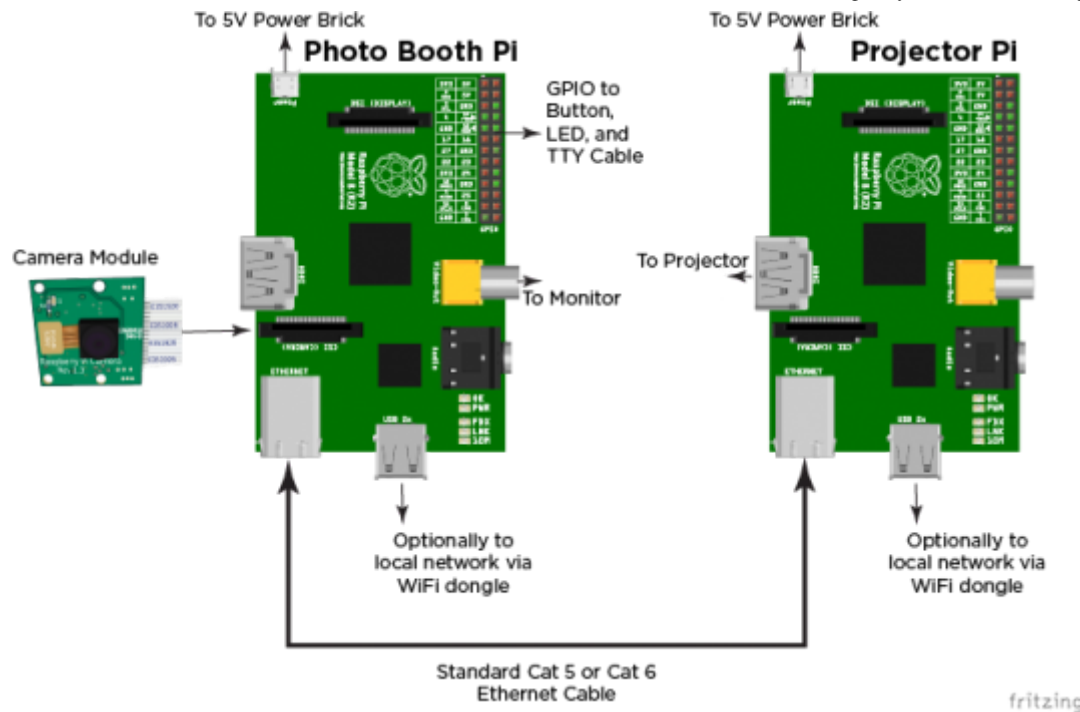
This is a companion application to my [Raspberry Pi Photo Booth](#). Sometimes at events, there is no access to the internet. But, I still want to share photos with the event attendees during the event.

When taking photos, the photo booth itself does show the most immediate photos in the moment to the users. But, without an internet connection to post online, there was no way to share with the larger event group.

So, I [developed a way](#) to use a second Raspberry Pi connected to a projector. It simply creates a slideshow of the available photos in the photo booth. As the event goes on, more and more photos are added to the slideshow.

Note, this is a bit of an advanced tutorial. But let's get started.

Hardware



This schematic shows the basic setup. I wanted the photo booth pi to work as originally intended, with lots of network flexibility. Sometimes I'll use it without the projector. So, I didn't want to have to change IP addresses or anything sneaky just to setup the pi differently.

The short story is connecting the two pi's via a standard Ethernet cable. Then, the projector pi finds the images on the photo booth pi to then build and show the slideshow.

Pi Setup: The Photo Booth

Pi Name

The first thing to do is [build the photo booth following all of the instructions here](#).

```
pi@photobooth:~ $
```

Next, since there are two Raspberry Pi's in this build, it's easier to know which one you are working on if you change their name. [Follow these instructions to set the photo booth pi's name to "photobooth"](#).

File Sharing

We want to share the pics on the photo booth to the other pi. So, we can use Samba to setup the share of /home/pi/photobooth/ to other network computers as /pishare

Follow these instructions on how to setup Samba on the photo booth pi.

When you edit the /etc/samba/smb.conf file, I added the following the the end:

```
[PiShare]
comment=Raspberry Pi Share
path=/home/pi/photobooth
browseable=Yes
writable=Yes
only guest=no
create mask=0777
directory mask=0777
public=no
```

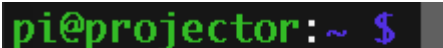
Notice the masks are 0777. That's not the best security protocols. So go ahead and tighten that down as you see fit. I also suggest changing the default password on both pi's.

Now, we'll switch over to the other pi.

Pi Setup: The Projector

Configure a brand new raspberry pi however you prefer. Look into [NOOBS](#).

Then, just like the photo booth, change the name of the pi. Follow these instructions to set the photo booth pi's name to "projector".

A terminal window with a black background. The prompt is 'pi@projector:~ \$' in green and blue text. A grey cursor bar is visible to the right of the prompt.

To get both pi's talking to each other, they need to be on the same network. This could be done with a WiFi router that both are connected to. But, I wanted to have a simple solution of just connecting the pi's with a standard Ethernet cable. Note, each pi has the ability to communicate

over the Ethernet cable without having to wire your own cross-over cable. To get the pi's on the same network, I chose to setup a DHCP server on the projector pi.

DHCP Server

To install the DHCP server, I followed [the Adafruit tutorial for setting up a WiFi Access Point](#), but I just switched out the instances of eth0 for wlan0, since we're using the local wired network instead of the WiFi.

Change the rc.local file to have the right settings when booting every time.

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

The settings should be:

```
sudo ifdown eth0
sudo ifconfig eth0 192.168.42.1
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
sudo iptables -A FORWARD -i wlan0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o wlan0 -j ACCEPT
sudo service isc-dhcp-server start
```

Now your DHCP server should be running. You could test by turning off the photo booth pi, connect the ethernet cable to both pi's, and turning on the photo booth. Does the photo booth pi get the right IP address of 192.168.42.11? Check with ifconfig.

To get the settings correct on every boot of the projector pi, install iptables-persistent.

```
apt-get install iptables-persistent
```

It will save your settings during install.

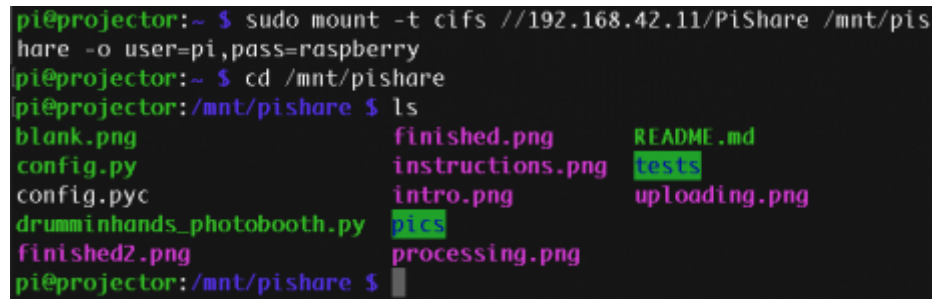
Mounting the Shared Folder

With the photo booth connected to the projector via the local network, and the photo booth serving a shared folder, now we can mount that folder on the projector pi. First, try just typing in the command to the projector pi's terminal. If you changed the default password (which is recommended) then update accordingly.

```
sudo mount -t cifs //192.168.42.11/PiShare /mnt/pishare -o user=pi,password=raspberrypi
```

To test that it worked, navigate to your new share on the projector pi and see if you can see files.

```
cd /mnt/pishare  
ls
```



```
pi@projector:~$ sudo mount -t cifs //192.168.42.11/PiShare /mnt/pishare -o user=pi,password=raspberrypi  
pi@projector:~$ cd /mnt/pishare  
pi@projector:/mnt/pishare$ ls  
blank.png          finished.png        README.md  
config.py          instructions.png    tests  
config.pyc         intro.png          uploading.png  
drumminhands_photobooth.py  pics  
finished2.png      processing.png  
pi@projector:/mnt/pishare$
```

If this is successful, then you're in business. This mount does not happen automatically on each boot up. I had trouble with timing getting the projector DHCP server running before the photo booth pi started up and also getting the photo booth to share its folder with the projector. There was a bit of a catch 22. So, mounting the drive on the projector happens within the python code.

Screen Settings

Now, let's **change the screen settings for the projector**.

```
sudo nano /boot/config.txt
```

For my projector, I changed the settings to:

```
framebuffer_width=800  
framebuffer_height=600
```

```
sdtv_mode=0  
sdtv_aspect=1
```

Python Code

OK, now let's move on to setting up the python code.

Download the [projector code and files from github](#) to the projector pi. I put them in a local file in /home/pi/photobooth/.

Update necessary passwords and file paths in the config_projector.py file.

```
nano /home/pi/photobooth/config_projector.py
```

Then [change the settings to start the python script at bootup](#).

```
sudo nano /etc/xdg/autostart/myscript.desktop
```

Add the following to the file and save:

```
[Desktop Entry]  
Type=Application  
Name=drumminhands_projector.py  
Comment=Start my script  
NoDisplay=false  
Exec=sudo python /home/pi/photobooth/drumminhands_projector.py  
NotShowIn=GNOME;KDE;XFCE;  
Name[en_US]= drumminhands_projector.py
```

After those settings are complete, turn off both of the the raspberry pi's. We'll test the boot up sequence next.

Booting Up at an Event

When you are at an event, the sequence and timing to boot up the two Pi's is important. Use the following:

1. First, plug in all of the cables per the wiring diagram above, but do not turn on any of the computers yet.
2. Then, boot up only the projector pi.
3. Wait until you see the desktop. At this point, the python code should be running in the background. It will wait 5 minutes until the other pi is all situated.
4. During this 5 minute wait, boot up the photo booth pi. It should go straight into the photo booth application.
5. Optionally, take a few photos in the photo booth to get some images in the system.
6. During the 5 minute delay on the projector pi, both pi's will setup their IP addresses, share the folder, mount the drive, etc. Basically, it gets all of the ducks in a row so that you are ready to start the slideshow.
7. Then, the projector will start displaying the slide show on it's own. If it's right at the beginning and there are no photos yet in the system, it will be a boring slide show. So find some people around you and start out getting some good, fun photos.
8. As the event progresses, the slide show will slowly build, showing all of the images.

When and Where to WiFi

So, where do you put the WiFi dongle? Well, you have a few options.

- If you are not posting photos online, then you don't need either pi to use a WiFi dongle. There will be a local network between the two pi's over the Ethernet cable. That's the minimum required.
- You could put the WiFi dongle in the photo booth pi. It will use the wlan0 IP address and get onto your local wireless network, and then to the internet. Yeah. If you choose to not put an additional WiFi dongle in the projector pi, then it will not be on the local wireless network. That's just fine, as long as you do not need to troubleshoot anything. If you do, use the optional TTY cable.
- You could put the WiFi dongle in the projector pi. If you do, the projector pi will then be on the local wireless network, and get internet access via that network. The projector pi will also share the connection to the internet to the photo booth via the Ethernet cable connection. Oooh, exciting. This makes it easy to SSH into the projector pi from a computer on the local wireless network. But still use the TTY cable to change settings on the photo booth pi. Unless of course you just want to SSH from the projector pi to the photo booth pi. That works quite well. This is my default setup.
- Put a WiFi dongle in both pi's. Yeah. Lots of internet access all around.

Now you're done. Enjoy.

Cover photo by Gustavo Medde on Flickr.



Search

Search site...

GO

Navigation

Portfolio

Contact



Share your thoughts

Logged in as drumminhands. [Log out?](#)

Comment

//

Comment



What We Do

We help exhibit experts create interactive media-based experiences so that visitors become more engaged and exhibits drive home interpretive messaging. We make visitors weep, jump for joy, point, share, and ask others to join them in their discovery. We strive to blow visitors' minds and have them bring their friends and family back for more visits.



Who We Are

Drumminhands Design is a group of designers, coders, interpretive thinkers, visitor experience crafters, and technological magicians. Led by founder and principal Chris Evans, we offer a commitment to quality custom media-based software and hardware solutions with easy to maintain parts and pieces.



How We Work

Unlike many focused on technology first, we build through careful messaging followed by correct media choices, driving home interpretive stories. Technology should disappear leaving a sense of awe and magic. We use principles of Universal Design to create inclusive and approachable experiences. We contribute to the overall collaborative and creative exhibit team of experts.

© Copyright 2021. All Rights Reserved.

[Twitter](#) [LinkedIn](#)