ITIS 3110L-L01

The Binaries

The SPi Cam

11/27/2016

**Table of Contents**

Overview…………………………………………………………………………….………..2

Appropriate Documentation..……………………………………………………….………3

Setting up Motion Pie………………………………………………………….…….4

Installing and Configuring NAS……………………………………………….…….9

DNS and Port Forwarding Setup………………………………………………….12

SSH Key Authentication……………………………………………………………18

How to maintain/update…………………………………………………….….….………..20

Summary…………………………………………………………………………….……….22

Names and Signatures………………………………………....………………….……….24

**Overview**

We have used a Raspberry Pi model 3 and have installed the MotionPie image to use the pi cam on the Raspberry Pi as a surveillance camera that will take still images of any motion detected on the camera. Putting in the ip address assigned to the the picam on a web browser shows a live stream video of the MotionPie’s Pi cam. In addition to this, after configuring the MotionPie, we will use a NAS (Network Attached Storage) to mount it on the Pi and a Windows computer, where the saved screen shots that the Raspberry Pi cam takes when it detects motion will be saved at.To add security, the ubuntu NAS server is remote and the only way to access it is through SSH. However, there is a key that is needed and only one computer has that key. Finally we will configure DNS so that the camera can be access from outside the local network.

The project was chosen because using a MotionPie as the primary component, there were many different designs that could have been implemented. By setting up such a system, we hoped to use the skills we have already learned like DNS, SSH, and NFS, while getting more in-depth into topics like NAS. Utilizing these different technology, we believe that to demonstrate our overall understanding of IT infrastructure.

In order to begin, we had to order the MotionPie camera. The Raspberry Pi has a 40 pin General purpose input-output (GPIO) connector to which ribbon cables can be connected to. Then a MotionPie .img file needed to be mounted to an SD card. We chose MotionPie because comes along with all tools we needed to get started. Before setting up DNS, all that is needed is to be on the local network and enter the ip address of the MotionPie and there is a live stream to the camera. From there, there is a GUI interface that allow changes to be made.

We decided to use NAS(Network Attached Storage) because we wanted to have remote access to the pictures the motion pie took when it detected motion. This way everyone on the same network could see all of the pictures taken without the need of going to just one place to access them, but instead they could be accessed by any device on the local network. We feel using a VM of a Linux Distro Ubuntu allows us to gain more knowledge about Linux.

Finally we wanted to the Ubuntu with the NAS to act as a remote server. To do that a user must SSH to the Ubuntu machine. SSH to the Ubuntu samba server was implemented because it adds another level of security. Access to the ubuntu VM requires a key to SSH to it. The NAS should run fine after setup and not need and major changes. We felt limiting access to the NAS configuration files prevents errors and protects it from any unwanted changes. That is also why we added keys to keep users away.

**Appropriate Documentation**

**IP table used in lab**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Description*** | ***Formula*** | ***Used in Lab*** | ***Answer*** |
| Samba /ubuntu | 192.168.0.n | ubuntu | 192.168.0.8 |
| Raspberry pi (client) | 192.168.0.n | Raspberry pi client | 192.168.0.12 |
| Motionpie | *192.168.0.n* | MotionPie access | 192.168.0.9 port :85 |

**Equipment needed**

1. A raspberry pi
2. Raspberry pi cam ( camera)
3. 8gb sd card
4. Internet connection
5. Linux distro debian, ubuntu , mint (we used ubuntu on this project)

**Setting Up MotionPie**

**Download MotionPie img. file**

Downloading MotionPie was really easy as , we got the appropriate software by visiting “<https://github.com/ccrisan/motionpie/releases>” website page and below are the steps we took to write the image on the sd card :

1. Go to <https://github.com/ccrisan/motionpie/releases> and download **Motion Pie** into your SD card
   1. Should be called [motionpie-raspberrypi2-20150719.img.gz](https://github.com/ccrisan/motionpie/releases/download/20150719/motionpie-raspberrypi2-20150719.img.gz)

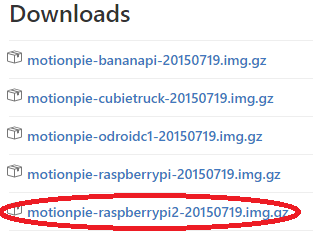


Figure 1 shows the link to download the MotionPie img file

1. Go to Ninite.com and download **7-Zip** under Compression category

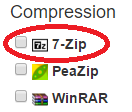


Figure 2 shows the name of the program needed to unzip the “.gz” extension file that you downloaded it early.

* 1. Extract [motionpie-raspberrypi2-20150719.img.gz](https://github.com/ccrisan/motionpie/releases/download/20150719/motionpie-raspberrypi2-20150719.img.gz)
  2. The file should be an image file now

1. Insert your SD card into the computer or laptop’s SD card reader (It should be G:/)
2. Format your SD card by right clicking on it and select format

**Installing the Motion Pie Image onto your SD card**

After dowloading the required image from the website, we downloaded a program that allowed us to mount a bootable image of MotionPie, and below we will show the steps we took.

1. Download WIn32DiskImager
   1. Run the program
2. Select the Blue folder then select MotionPie img file under where you extract the [motionpie-raspberrypi2-20150719.img.gz](https://github.com/ccrisan/motionpie/releases/download/20150719/motionpie-raspberrypi2-20150719.img.gz)

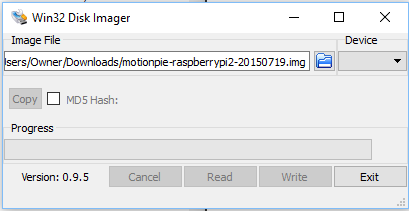


Figure 3 shows the name of the file after it is chosen to be written on the SD card.

* 1. Drop down on Device select your SD card drive (e.g G:/)
  2. Select Write, the click Yes

**Installing the camera**

To attach the camera to the Raspberry Pi we had to:

1. Lift the clip
2. Slip the camera cord in
3. Push the clip back down



The ribbons go into the Raspberry Pi as shown on Figure 4

Next we booted up the Raspberry Pi with the SD card inserted. We had to make sure the pi was connected to the Internet through a wired ethernet connection. In addition to this, we connected it to a monitor using a standard HDMI cable; however, if there is no monitor to check what ip address was assigned, you can go on your home router and check all of the active clients to check which ip belongs to the MotionPie. The reason is because MotionPie does not have a GUI. MotionPie will just run with your network connection and you will have access to it by the internet interface it has.

**Initial setup Motion Pie**

After powering up the Pi with the newly installed MotionPie image installed on it, we were then able to see the MotionPie boot up and locate your IP-Address listed above in the monitor. This IP-Address is important because this will be your way into your GUI interface.

**Transition to another computer**

Now you get onto any laptop and type in the IP-Address you wrote down above into your URL. (e.g 192.168.0.10:8081) The :8081 is the defualt port installed with MoitionPie, you can change this in the admin GUI.

The default login when freshly startup is **admin** or **user**, no password is set at this time. So now we had owner’s permission to change any settings.

Once logged in, our Russian dolls **Molly**, **Holly**, and **Polly** showedon the MotionPie home screen GUI.



Figure 5 shows an image from the MotionPie camera after setup Holly, Molly and Polly

Next you should add some security protection to your pi-camera. To add user authentication to your MotionPie click on the key icon.



The administrative settings can be changed by clicking on the screen

like in Figure 6

We were now be able to see and make change to the MotionPie setup. We make change to your admin and setup password here.

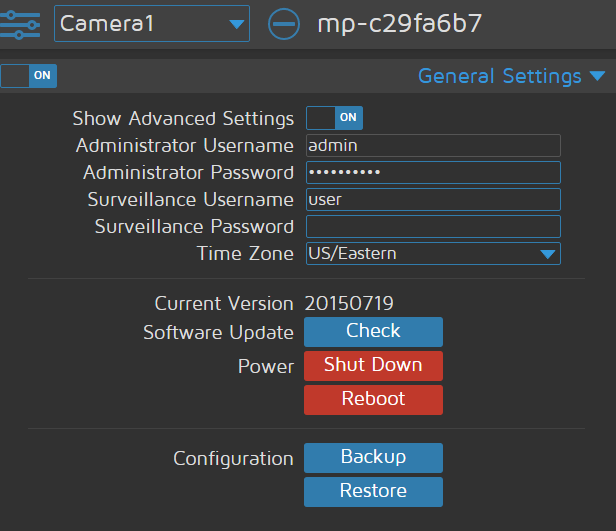


Figure 7 the admin GUI interface

Make sure to click to apply button on the top right corner every time you update anything.

Next, we set-up if you want the MotionPie to capture pictures or video. In our case, we chose to make our MotionPie capture pictures. To set up MotionPie to capture picture make sure you select “**ON**” from the “**Still Images**” section. You can customize the quality of the capture pictures and manage of how long you want the pictures to be kept. The MotionPie was finally ready to be tested out..

**Installing and configuring NAS**

**Install samba**

To install samba all we did was to use a simple command “apt-get install samba “ onto the linux distro we used which in our case was “ubuntu.” For this part we had to make sure we had an internet connection and make sure our ip address was static on our own network.

apt-get install samba

**Configure smb.conf**

“gdbcommands smb.conf ” once inside make sure to make a backup copy of both this files anywhere safe. We made a backup copy to my home directory. This can be done by using the “cp” command learned in this course.

Creating a new “smb.conf” file on “/etc/samba”. After the file was created we edited this file with the following content if anyone else is trying to do it ,include the content that is on \* **bold letters**\*:

|  |
| --- |
| ***[global]***  ***server string = Sebastian media*** *----(can be any name you want to use)*  ***workgroup = WORKGROUP*** *-----------( this is usually any name)*  ***netbios name = sebastian*** *--------------(can be anything as long as less or equal to 15 characters)*  ***security = user*** *-------( its user, which means users on the same network have access)*  ***encrypt passwords = yes*** *---------------( it needs to be the same as shown)*  ***guest account = nobody*** *-----------------(enable to allow guest*  ***name resolve order = bcast host*** *------------( type the same way shown in order for it to work)*  ***include = /etc/samba/smbshared.conf****------( this is the path to the config file of shared folders).* |

**Configure smbshared.conf**

For this step we had to create a file called “smbshared.conf” in the “/etc/samba” folder to make it work. This folder will hold the folders that you want to share to everyone on your network to use. **Important:** make sure the path of the folder that you want to share is correct. For our project we just shared the Pictures folder from the ubuntu machine to everyone on the network.

Content of smbshared:

|  |
| --- |
| ***[Pictures]****-----------------------( name of the folder we used)*  ***comment = ubuntu file share****---------( any name desired)*  ***path = /home/sebastian/Pictures****---- (path of the folder we wanted to share to everyone)*  ***browsable = yes****----------( this allowed the folder to be browsed)*  ***guest ok = yes****---------------------------(allows all users)*  ***read only = no-----------****( users can erase items on the shared folder)*  ***available = yes****------------( this makes the folder available)*  ***public = no-------****( this is set so that only the local network has access to it)*  ***writable = yes****--------------------------( this allows files to be changed or added on the shared folder)*  ***valid users = sebastian --------------****(this is the user allowed to login on the shared folder)* |

After changing this,we saved our changes and once they got done saving the changes on, got out of the folder.

**Adding users with password**

To add a layer of security onto the NAS we set up a user with a password to be able to log in securely onto the shared folders from any machine on the network. We typed “sudo smbpasswd -a sebastian” to basically add “sebastian” as an authorised user and then we also got prompted for a password for that user to use. If anyone is trying to do this the syntax for the command is “**sudo smbpasswd -a [username]**”.

* Next we had to Add “smbusers” file to the “/etc/samba” folder to be able to use this username to be autheticated.

The content of this file should be:

**sebastian = “sebastian”**

If anyone is trying to do this step the syntax to be included on this file should be:

**Username = “username”**

Once this step is done we saved our changes and got out from the file.

* For the next step we had to create rules that would allow the samba server operate with the current firewall on the network so it would allow it and not block it from being used by others on the network . To accomplish this we had to use a simple one-line command:
* **Sudo ufw allow samba**

The command then allowed samba to finally connect to the network and allow the folders to be shared. With other devices that were connected to the same local area network.

**DNS and Port Forwarding Setup**

We created a virtual server/ port forwarding on the local router. Inbound port was to 42132 and the private ip address which is the motionPie with the camera is 192.168.0.10. The local port is 85. We called this server PiSpy.

This is our local router. All we did here is go to the Virtual Servers/ Port Forwarding setting under the firewall table. We have to implement this step into our local router so the outside network can contact our local network through our router using the ip address 192.168.0.10 and only from port 85 on our local side.

|  |
| --- |
| aaris.PNG |

Figure 8 local router GUI interface, setting up the port forwarding server

Next we had to go to [www.no-ip.com](http://www.no-ip.com) to sign up for a host website server to host our local website so that we can access our local website anywhere outside of our network. This step is essential to DNS to work properly, carefully configure the questions asked during this process. The information input on www.no-ip.com should match your local network server that was just set-up on the local router PiSpy (Refer to Figure 8).

|  |
| --- |
| no-ipAcct0.PNG |

Figure 9 shows the no-ip.com’s confirmation page to activate the account

Now that we had our web host “no-ip” account set up so that we can access our local network from anywhere. We now have to set our camera Pi settings to link up with our outside network “no-ip” web server.

For this step, it will required us to use another raspberry pi to configure the setup linkage between our local network and the outside network through the terminal.

Linking the local network server PiSpy and no-ip outside network host

|  |
| --- |
| *pi@raspberrypi:~ $ sudo bash*  *root@raspberrypi:/home/pi# cd /usr/local/src/*  *root@raspberrypi:/usr/local/src# wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz*  *--2016-11-13 21:41:50-- http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz*  *Resolving www.no-ip.com (www.no-ip.com)... 8.23.224.110*  *Connecting to www.no-ip.com (www.no-ip.com)|8.23.224.110|:80... connected.*  *HTTP request sent, awaiting response... 301 Moved Permanently*  *Location: http://www.noip.com/client/linux/noip-duc-linux.tar.gz*  *[following]*  ###Continues on pg. 14  *--2016-11-13 21:41:50--*  *http://www.noip.com/client/linux/noip-duc-linux.tar.gz*  *Resolving www.noip.com (www.noip.com)... 8.23.224.107*  *Connecting to www.noip.com (www.noip.com)|8.23.224.107|:80... connected.*  *HTTP request sent, awaiting response... 200 OK*  *Length: 134188 (131K) [application/x-gzip]*  *Saving to: ‘noip-duc-linux.tar.gz’*  *noip-duc-linux.tar. 100%[=====================>] 131.04K 238KB/s in 0.6s*  *2016-11-13 21:41:51 (238 KB/s) - ‘noip-duc-linux.tar.gz’ saved [134188/134188]* |

Now we download the tar file and then unzip it. Then we install into the directory noip-2.1.9-1. We then enter our outside network host login and password (no-ip login information) to authorize us to have access to our outside web host server.

|  |
| --- |
| *root@raspberrypi:/usr/local/src# tar xf noip-duc-linux.tar.gz*  *root@raspberrypi:/usr/local/src# cd noip-2.1.9-1/*  *root@raspberrypi:/usr/local/src/noip-2.1.9-1# make install*  *gcc -Wall -g -Dlinux -DPREFIX=\"/usr/local\" noip2.c -o noip2*  *noip2.c: In function ‘dynamic\_update’:*  *noip2.c:1595:6: warning: variable ‘i’ set but not used [-Wunused-but-set-variable]*  *int i, x, is\_group, retval, response;*  *^*  *noip2.c: In function ‘domains’:*  *noip2.c:1826:13: warning: variable ‘x’ set but not used*  ###Continues on pg. 15  *[-Wunused-but-set-variable]*  *int x;*  *^*  *noip2.c: In function ‘hosts’:*  *noip2.c:1838:20: warning: variable ‘y’ set but not used [-Wunused-but-set-variable]*  *int x, y, z;*  *^*  *if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi*  *if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi*  *cp noip2 /usr/local/bin/noip2*  */usr/local/bin/noip2 -C -c /tmp/no-ip2.conf*  *Auto configuration for Linux client of no-ip.com.*  ***Please enter the login/email string for no-ip.com dyang29@uncc.edu***  ***Please enter the password for user 'dyang29@uncc.edu' \*\*\*\*\*\*\*\*\*\*\*\****  *Only one host [pispy.ddns.net] is registered to this account.*  *It will be used.*  *Please enter an update interval:[30]* |

We cd in /etc/rc.local. Go into this file and we added in this line /usr/local/bin/noip2

|  |
| --- |
| *root@raspberrypi:/usr/local/src/noip-2.1.9-1# nano /etc/rc.local*  *++++++++++++++++++++++++++++++++++++++++++*  *# By default this script does nothing.*  *# Print the IP address*  *\_IP=$(hostname -I) || true*  *if [ "$\_IP" ]; then*  ###Continues on pg. 16  *printf "My IP address is %s\n" "$\_IP"*  *fi*  ***/usr/local/bin/noip2***  *exit 0* |

To complete the booting procedures we boot all the settings configure to no-ip, we run this command

|  |
| --- |
| *root@raspberrypi:/usr/local/src/noip-2.1.9-1# sudo /usr/local/bin/noip2* |

To check if our web server is running we type in this command

Once you see the result as this one that means your local network is now linked or forwarded to the host web server

|  |
| --- |
| *root@raspberrypi:/usr/local/src/noip-2.1.9-1# sudo /usr/local/bin/noip2 -S*  *1 noip2 process active.*  *Process 2547, started as /usr/local/src/noip-2.1.9-1/noip2, (version 2.1.9)*  *Using configuration from /usr/local/etc/no-ip2.conf*  *Last IP Address set 172.73.140.142*  *Account dyang29@uncc.edu*  *configured for:*  *host pispy.ddns.net*  *Updating every 30 minutes via /dev/wlan0 with NAT enabled.*  *root@raspberrypi:/usr/local/src/noip-2.1.9-1#* |

**Test our web server if it's running**

Instead of using our private local address 172.73.140.142:42132, now anyone anywhere can access our devices and view our cam by typing in **pispy.ddns.net:42132** to their URL.

Type to URL: **pispy.ddns.net:42132**

Result: Anyone on this weblink will be able to see what our live spy cam is streaming. Below we have attached an image showing how it looks when you go on to the specified URL address.

|  |
| --- |
| no-ipUp.png |

Figure 10 shows a successful access to the MotionPie by entering http://pispy.ddns.net:42132

\*Only if or when you want to shut down your webserver

Run this command in your raspberry pi terminal if you want to shutdown your web server

|  |
| --- |
| *root@raspberrypi:/usr/local/src/noip-2.1.9-1# sudo /usr/local/bin/noip2 -K 2547* |

**SSH Key Authentication**

SSH was previously set up on another Raspberry Pi that holds Public Key. For Ubuntu we installed ssh with *apt-get install ssh*. On the Pi we ran the *ssh-keygen -t rsa -b 4096.* Using the *scp ~/.ssh/id\_rsa.pub sebastian@192.18.0.8:* command, we copied the key to the server.

On the Ubuntu the following commands were executed to create an empty directory for authorized keys:

|  |
| --- |
| *mkdir ~/.ssh*  *touch ~/.ssh/authorized\_keys*  *chmod 700 ~/.ssh*  *chmod 600 ~/.ssh/authorized\_keys* |

In order to make the keys the only way to authenticate for ssh, on the client sider make the following changes to /etc/ssh/sshd\_config.

|  |
| --- |
| *# Make sure that we can authenticate using our public/private key pair*  *PubkeyAuthentication yes*    *# Disable logins using standard passwords*  *PasswordAuthentication no* |

The commands resulted in a directory .ssh made, created the file authorized\_keys, and made the directory inaccessible only to user sebastian with *chmod 700* and authorized\_keys is only readable and writable*.* Then ssh needed to be reset on the server using */etc/init.d/ssh restart*.

To test whether or not the key authentication worked, we tried to use” PuTTy” on a Windows machine to attempt to SSH into the ubuntu. The screenshot below shows a failed attempt to SSH into the ubuntu.

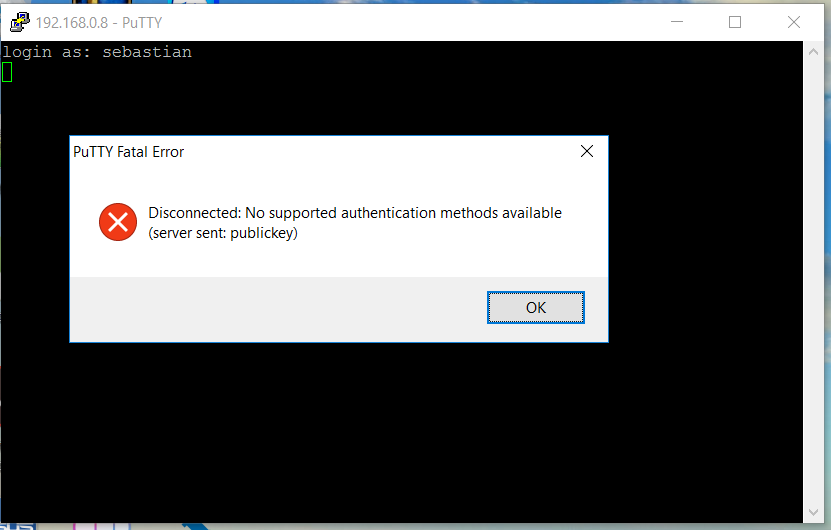


Figure 11 is a screenshot of a failed attempt to SSH into the ubuntu from a non-authorized device.

|  |
| --- |
|  |

**How to maintain/update**

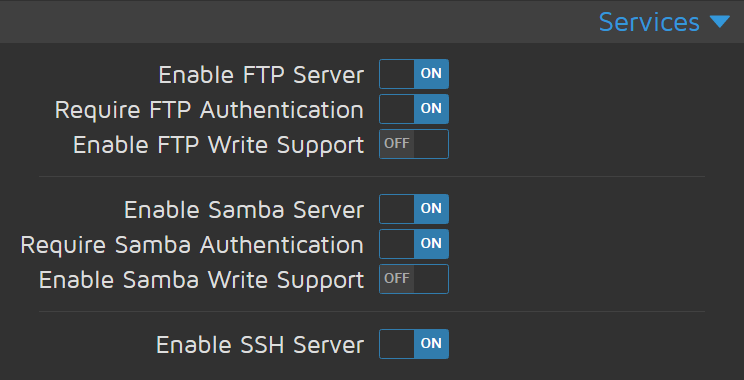


Figure 12 Motion Pie GUI Service settings

There are many ways that we could make alterations to the system. The software that comes with MotionPie offer different services to use to connect. We could use FTP or SFTP for added security. Because there is no CLI for this, any changes would have to done from the GUI. We could have it set so that it can run a command when there is a notification. The MotionPie can run a command.

For maintenance, if the MotionPie isn’t static ip, then it is possible for the ip address to change when the pi powers off. Then if it does change, the settings will need to be changed on the virtual server/port forwarding.

Some of the proper ways to maintain the system would be to regularly change the system administration password to manage the Pi cam, ensure that the NAS (Network Attached Storage) folders are updated and make sure the path has not changed, make sure the authorized machine with the private key public key to the server is always updated and finally make sure that the proper DNS/port forwarding service is up and running by logging in and check the configurations are still the same and match the pi cams. Another way to maintain the system is by making sure MotionPie stays up to date.

To properly change the system administrator password and username on the pi we had to do the following:

* Go to the URL and type the assigned IP address to the cam and the port number
* Sign in as admin user and go to general settings tab to change password.

To check for updates on the system to make sure everyting was up to date. We had to check for the points below

* Go to the URL and type the assigned IP address to the cam and the port number
* Sign in as admin user and go to general settings tab press Check.

To make sure NAS is always up to date with the latest version as well as make sure the folders are in the correct path make sure to follow the steps below:

* sudo apt-get update samba to install latest samba patch
* to configure samba, you can use the editor of your choice to edit /etc/samba/
* Make sure the file named smbshared.conf has the right path for the folders you wish to share.

Making sure that SSH( secure shell) is up to date with the right public key and private key where they belong is crucial as well adding more possible users that have access or removing them if they are not using the service. The command below allow the key to be copied to other users.

* On the server run *scp ~/.ssh/id\_rsa.pub* ***newuser@ipaddress***

**Summary**

**What did we think about the project?**

Overall this was a great project, the group enjoyed working on. “The Spi Cam” as we named it, was a great and very cool project that was not easy to accomplish but in the end we did and it worked as planned. The project itself was very time consuming as we only had three group members making the workload hefty on each member. The Spi Cam project required many parts to successfully work, each member was appointed one of these areas to research on, which in the end turn out to be a very nice learning experience as we all thought each other a little about the areas we had worked on. Every group member loved the spi cam so much that we all said we could further work on it outside of class to make it even better with more sophisticated software..

**What problems did we encounter?**

As with every big project we encountered a few problems to get it work, at the beginning, middle and end. Some of the early problems were trying to make the right motionPie image work, as there were many versions of it. Half way down the project when we had already set up the NAS(network attached storage) to take pictures when detecting any motion and had configured ssh(secure shell) with a public and private key to connect. We had a small issue or had made a small mistake were we didn't set a static ip on the MotionPie. So, by the time the DNS(domain name system) and port forwarding were configured, it had an ip address that had been previously assigned by DHCP, so as a result when we rebooted the system , the DHCP server gave another ip address to motion pie making the DNS part useless. There was also a hardware issues when the pi cam ribbon had fallen out of place on the motherboard, and we spent a couple of hours trying to find out why the camera was not working.

**What did we learn ?**

This was a great project, all of the group members learned a lot from it as we spent countless hours to make this whole entire project to work, and in the end everything went accordingly and it worked just fine. We learned that working on a project is one of the hardest things to do, all of the team members in the group felt the pressure of meeting deadlines just like a real job. Moreover, everyone’s mind is not always on the exact same page, this is why teamwork is such a great skill to have and is needed to be learned to be successful. As a group, we learned that this project could have been harder maybe even impossible if we were to take on this project individually. This project was a great experience for every single member in the group as we gained the knowledge to make our own security cameras or even use a NAS to share media in our own local network

**What would you had done if given more time?**

given the case that we would have had some more time to work on this project, there are more things that we would have liked to add on the pi cam . For example, SMTP would have been an implementation we would have added if time allowed, so MotionPie would send out e-mails in the event the camera detected any activity. Other cool thing we would have incorporated in our camera if given more time would have been the integration of night vision with the motion detection.

**Names and Signatures**

Sebastian Mier\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Arrington Mills\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Derek Yang\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_