[Date]

PI WIRELESS SERVER

**CSE461: Digital System Design Project**

**PI WIRELESS SERVER**

**Group No: 05**

**Group Members:**

Geetanjali Oishe -13101211

Rafid Ahmed-13101209

Biprojit Roy-13101207

**Table of contents**

1. Introduction …………………………………………………………………..3
2. Motivation ……………………………………………………………………3
3. Equipment ……………………………………………………………………4
4. Setup ………………………………………………………………………....4
   1. HOSTAPD and DHCP installation 4 …………………………………....4
      1. Set up DHCP server 5..……………………………………………….5
      2. Set up wlan0 for static IP 7.……………………………………….....7
      3. Configure Access Point 7 …………………………………………....7
      4. Configure Network Address Translation 8 ………………………....8
      5. Update hostapd 9 …………………………………………………….9
   2. Setup mini DLNA 10 …………………………………………………...10
      1. Installing 10 ………………………………………………………...10
      2. Connecting the hard drive 10 ……………………………………....10
      3. Mounting the drive on startup 11 …………………………………..11
      4. Configuring MiniDLNA 13 ………………………………………...13
   3. Automating TSR ………………………………………………………..14
   4. Attendance automation …………………………………………………15
5. Working Procedure …………………………………………………………17
6. Result ……………………………………………………………………….17
7. Conclusion ………………………………………………………………….17
8. Reference …………………………………………………………………...18

**Introduction:**

PI WIRELESS SERVER is basically a raspberry pi based project. The product outcome by the project comes with some very useful features which will make life easier and exciting. The features are:

1. **Attendance automation:** By this feature teachers will not have to take attendance manually. When our device is turned on it will cross match the mac address of corresponding student which is previously saved in a text file. After cross matching if the mac address matches it will give attendance. There will be certain time for taking attendance.
2. **TSR access:** By this feature students will have access to the TSR in classroom. There will be a time allocation for files such as certain files will be opened for certain times.
3. **Online media streaming:** By this feature users can stream media such as videos, music, pictures and others which is stored in the memory of raspberry pi.
4. **Internet Sharing:** This product can create a Wi-Fi zone. After connecting to the Wi-Fi the users will be able to connect to the internet.

There have been some previous works regarding some of our features. Such as online media streaming, internet sharing. But as whole the project provides a unique solution to automation.

Actually this project is an outcome of the urge to a solution we face. On daily basis there are some problems we see or we face such the wastage of time giving attendance, nothing to do in boring journeys, no lab to access TSR. To solve these problems we wanted to build a digital system which will make thing easier.

**Motivation:**

As previously said the story behind this project is the urge to solve some problems and make things more efficient. In the same time this product will save time, make life easier and entertain us. Here are some reasons which will describe the motivation of the project and how people will be useful:

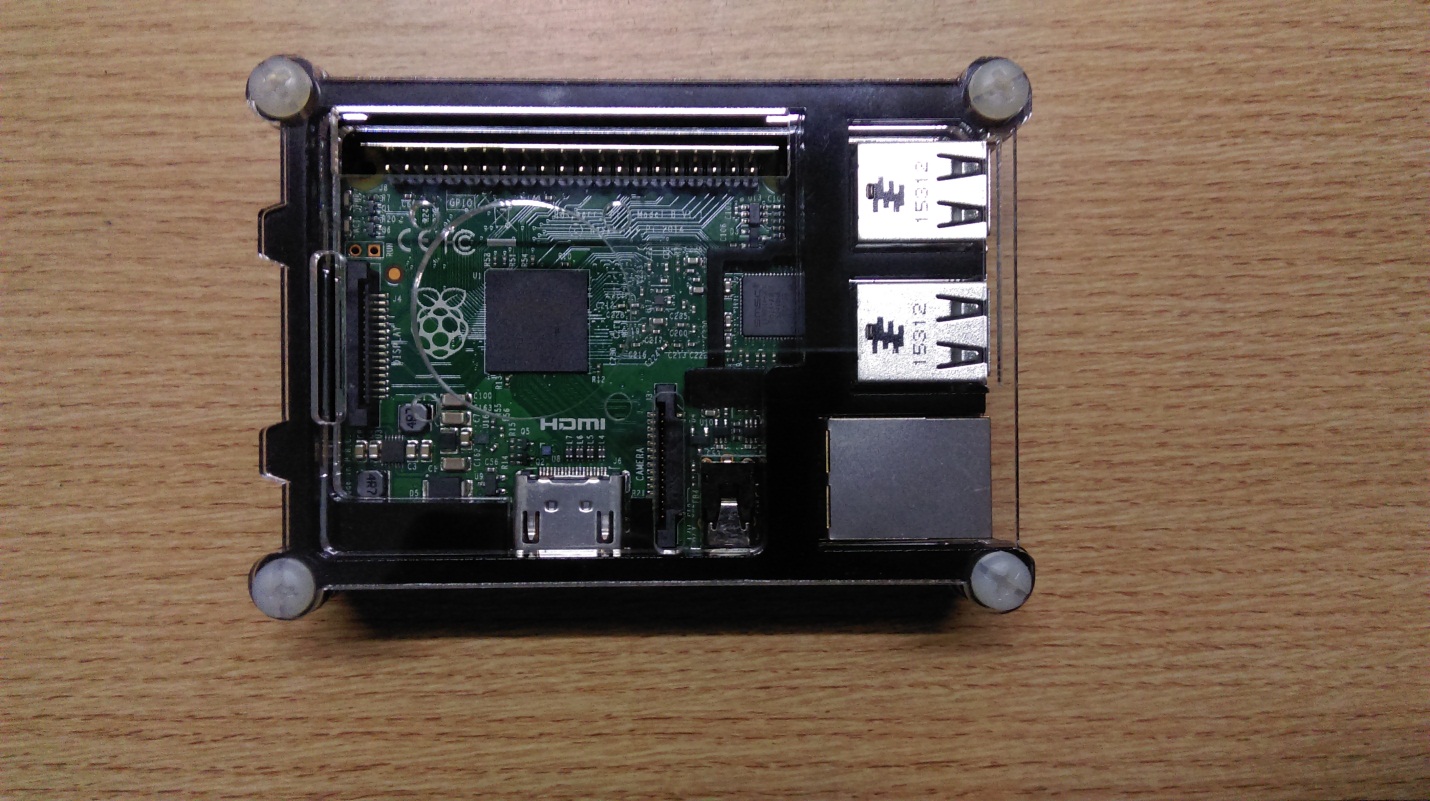
1. In this era of modern science and technology why should teacher waste their valuable times taking attendance. This should have been automated. So to solve this problem we created a feature to our project which will take the attendance automatically when students will connect to the Wi-Fi of our device. It will save time and energy wasted for the attendance.
2. Why should students always have to go to the computer lab to access the TSR. It kills time. Besides, sometimes there are not enough seats in labs. As a result in the time of need they have to wait. So to make students life easier we added the feature by which students will have access to the TSR in the class time. This will also help teacher to share teaching materials with the student during class time.
3. On boring journeys we have nothing to do. We just get bored. To make the boring journeys entertain able we have come with the solution of media streaming. Users will stream media from servers and enjoy those just connecting to the Wi-Fi. Journeys will not be that boring anymore.
4. You don’t need to make hassle to just connect to the internet. By our device user will have internet access just by connecting to the Wi-Fi.

When you need a product as effective and efficient definitely price will go up. But in our project we have cut down the total expense as low as possible. But the expense is bearable for the purposes we have built this.

**Equipment:**

As said earlier our project is a raspberry pi base project we used raspberry pi as the microprocessor. Additionally to achieve the features we have added some other devices. Here are the devices we have used:

1. **Raspberry Pi:** Raspberry Pi can be described as a low cost mini sized computer. It has to be connected with a keyboard, mouse and a monitor. We used it to do the computing works of our digital system. We used python as our programming language.



1. **Wireless network adapter:** We have used a wireless network adapter to create the Wi-Fi zone.



1. **Micro SD card:** All computer needs memory. As said earlier raspberry pi can be called a mini computer there is a need of memory. So for that memory we used a 8 GB micro SD card.



1. **Pen drive:** For the external memory we used pen drive. In its memory the attendance and media files will be there.



1. **Internet connected Ethernet cable:** As we will create Wi-Fi zone and enable media streaming we need a internet connection in our device. For that we need a internet connected Ethernet cable to connect to the internet



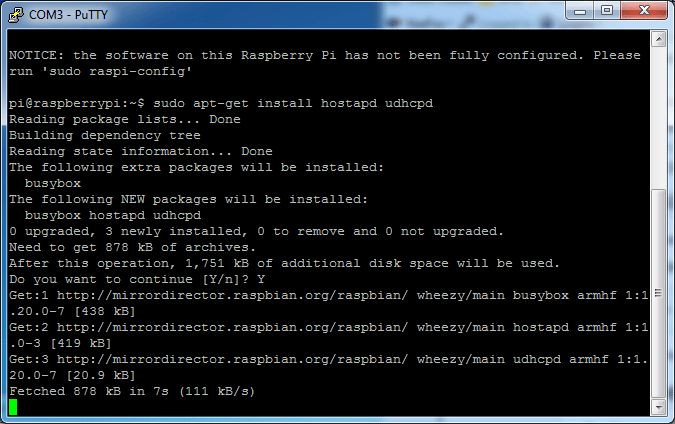
**Set-up**

Firstly we boot up the raspberry pi with raspbian OS. Then we inserted the router in a usb port. As initially, the router is not active, we had to follow and install some protocol as well as configure some files to make it up and running. Below we describe the necessary steps to configure:

**(a) HOSTAPD and DHCP installation**

First we install the software onto the Pi that will act as the 'hostap' (host access point). Internet access is needed for this step so make sure that Ethernet connection is up!





**i) Set up DHCP server:**

Next we will edit **/etc/dhcp/dhcpd.conf**, a file that sets up our DHCP server - this allows wifi connections to automatically get IP addresses, DNS, etc.



We edit the file to this form



Then scrolled down to the bottom and add the following lines



Run



and scrolled down to **INTERFACES=""** and updated it to say **INTERFACES="wlan0"**

**ii) Set up wlan0 for static IP:**

Next we will set up the **wlan0** connection to be static and incoming. run **sudo nano /etc/network/interfaces** to edit the file

We find the line **auto wlan0** and add**#** in front of the line and in front of every line afterwards. Basically we just remove any old **wlan0** configuration settings; we'll be changing them up.

Add the lines



After allow-hotplug wlan0 - see below for an example of what it should look like. Any other lines afterwards should have a **#** in front to disable them



Assign a static IP address to the wifi adapter by running

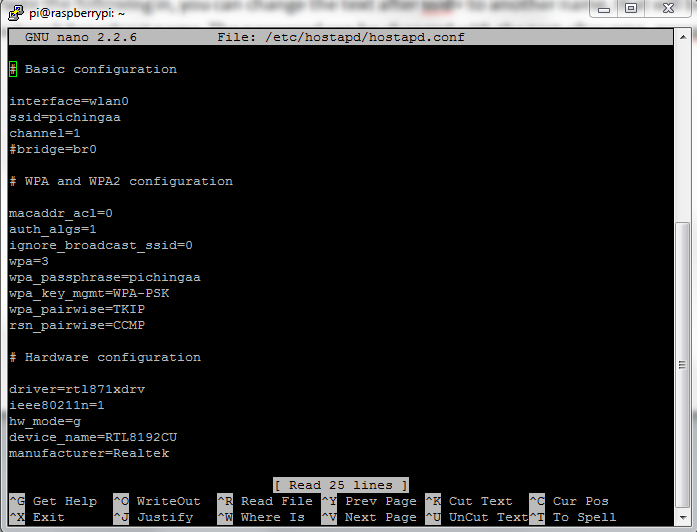


**iii) Configure Access Point:**

Now we can configure the access point details. We will set up a password-protected network so only people with the password can connect.

Create a new file by running **sudo nano /etc/hostapd/hostapd.conf**

We write the following in, the file can be changed by editing the text after **ssid=** to another name, that will be the network broadcast name. The password can be changed with the text after **wpa\_passphrase=**



Now we will tell the Pi where to find this configuration file. Run



We find the line **#DAEMON\_CONF=""** and edit it so it say **DAEMON\_CONF="/etc/hostapd/hostapd.conf"**

We have to remove the # in front to activate it!

**iv) Configure Network Address Translation:**

Setting up NAT will allow multiple clients to connect to the WiFi and have all the data 'tunneled' through the single Ethernet IP.

We run the following commands to create the network translation between the ethernet port eth0 and the wifi port wlan0



To make this happen on reboot we run



We run **sudo nano /etc/network/interfaces** and add



to the very end

**v) Update hostapd:**

Before we can run the access point software, we have to update it to a version that supports the WiFi adapter.



Now build hostapd:



After a while, control should be given back to the terminal.



This last step will move the created hostapd binary to **/usr/local/bin** add a startup script and create a configuration file in **/etc/hostapd/hostapd.conf**. We edit this configuration file and start the hostapd service:



**Finishing up!**

OK now that we know it works, time to set it up as a 'daemon' - a program that will start when the Pi boots.

We run the following commands



We can always check the status of the host AP server and the DHCP server with



To start the daemon services. We have to verify that they both start successfully (no 'failure' or 'errors')

Then to make it so it runs every time on boot



**(b) Setup mini DLNA**

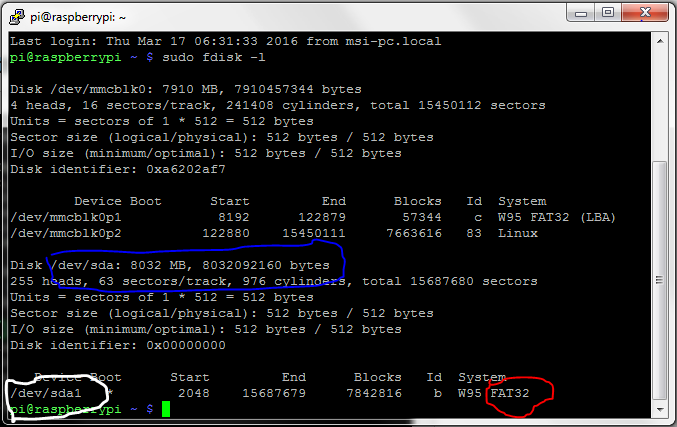
**i) Installing:**

First we run this command



Before we can start our media server we need some media of course. So what we are going to do is make it so that our media hard drive is mounted on start-up.

**ii) Connecting the hard drive**

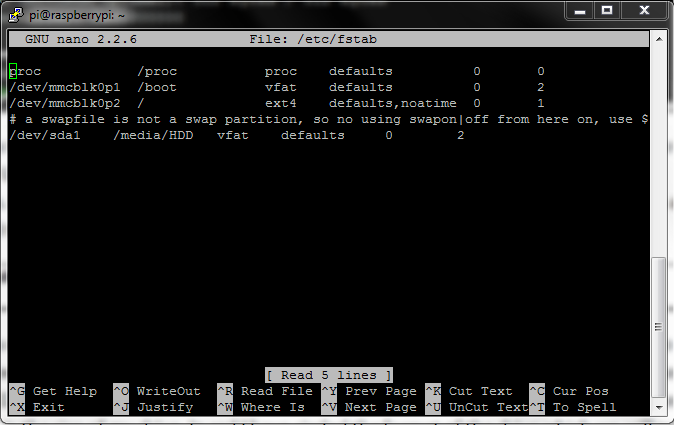


To do this the first thing we need to do is plug in our media drive. We make sure you plug it into the powered USB hub and not directly into the raspberry pi because sometimes that can cause problems. Once we have done this we need to go back to putty or whatever SSH client you are using and type in:



What this does is it shows use important information about the drives that are connected to our raspberry pi. In the picture we have circled the name of our drive in white. In our case it was /dev/sda1. We have circled in green says that the drive /dev/sda has 8 GB which is the size of our drive. In red we have circled the format of the drive which we will need in the next step. We are going to need to know the name of the drive and the format of the drive in the following steps.

**iii) Mounting the drive on startup**



We need to have our media drive to be mounted on startup so that we can access its contents. To do this we are going to need to make a folder to mount it to. We can do that by using this command:



What this command does is makes a folder called HDD in the media directory. So once we have made this folder we need to give it read write permissions. We can do this by using this command:



This command command tells the folder HDD that it has all permission. This means that it has read and write permissions which is what we wanted.

Now we need to edit the fstab file. This is the file that the raspberry pi operating system refers to when it is looking to see which dives to mount at startup so we need to put our media drive in that file. We can do that by using the command:



We go to the bottom of this file and we are going to add this line:



The first part where is says /dev/sda1 is the is the name of the hard drive that we want to add. Remember from the previous step. The next part is the place where we are going to mount it to. Then we have the format of the hard drive. In this case it is fat32. And finally the 0 and 2 at the end are permissions.

Now if we reboot the raspberry pi and move into the directory /media/HDD, we should be able to see all the files on your hard drive.

**iv) Configuring MiniDLNA:**

To start configuring MiniDLNA we need to edit the config file. This can be done by using this command:



Once we have that file open we are going to need to change that part that looks like this:



to this:



and this:



to this:

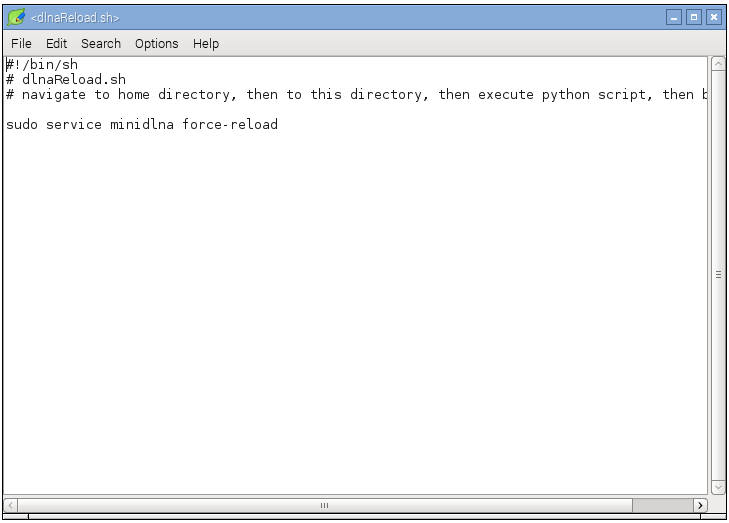


In the line above where I have put LazarusPit can be whatever we want.

Now that we have configured MiniDLNA we have to refresh it. To do this we can run the following commands:



We prepare a shell script to be run at startup for reloading DLNA each time pi starts



**(c) Automating TSR**

Install samba protocol





Set password



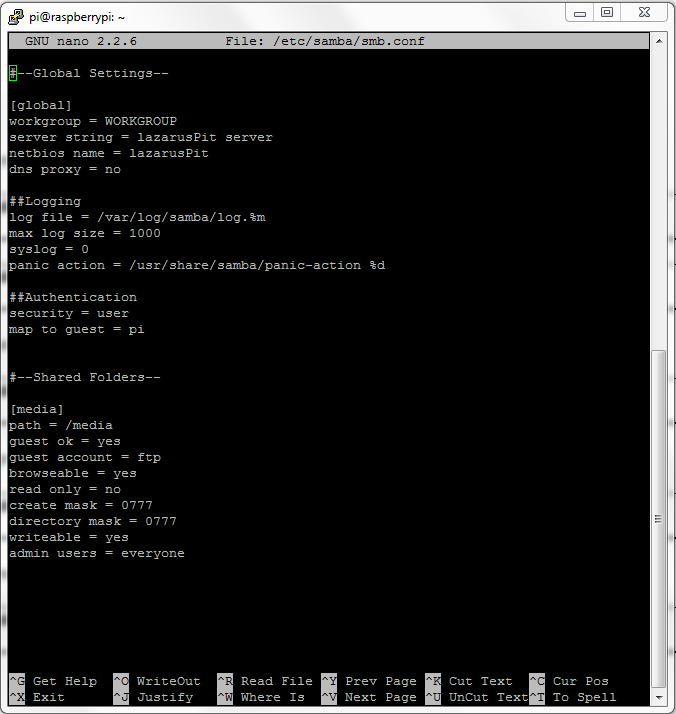
Remove previous smb configuration file



Create new smb configuration file



Set basic parameters and authentication

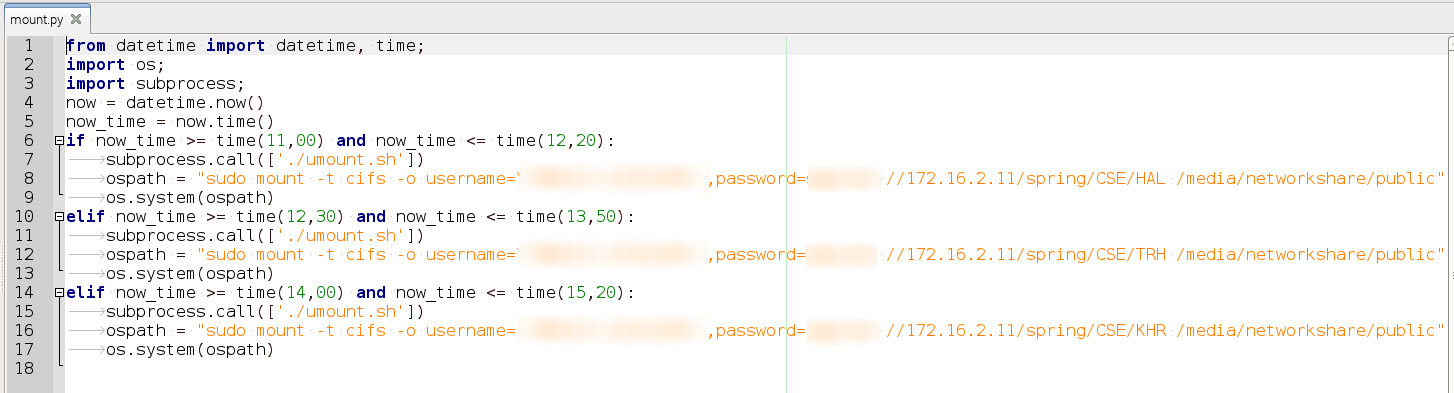


Restart samba process



We develop a python program to constrain the appearance of a TSR folder after a certain time

Then we make a shell script that runs this program at startup

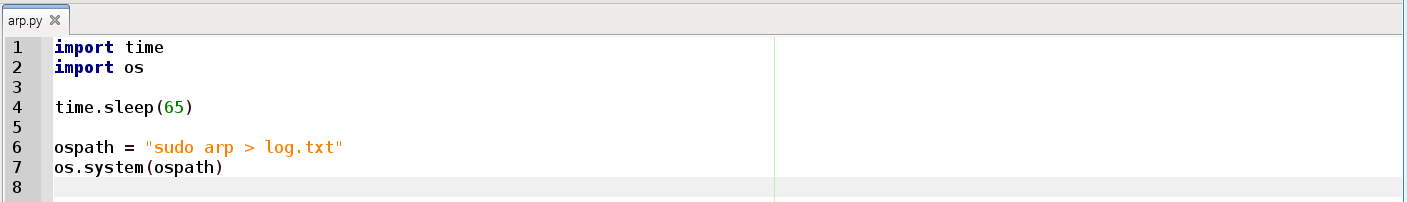


We have to give user id and password appropriately and specify the path of the folder in TSR for broadcast

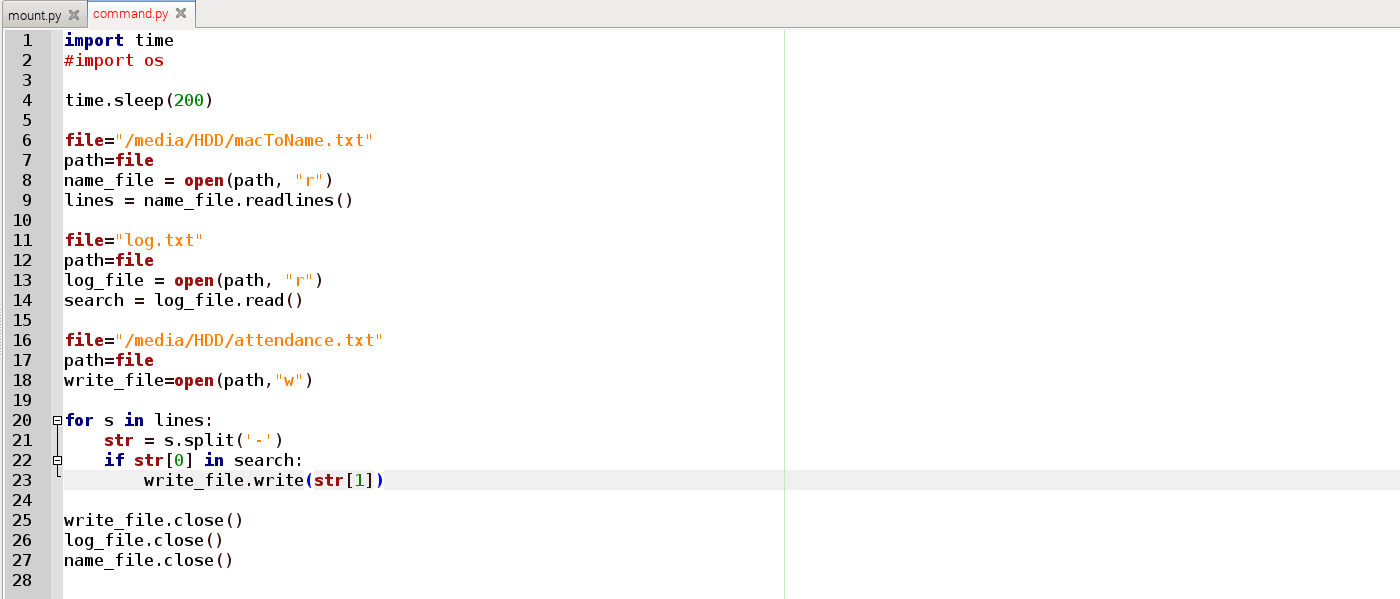
**(d) Attendance Automation**

First we stored a text file that contains students name and corresponding mac address of their devices. When we power up the raspberry pi, it uses ARP to grab the mac addresses connected to it and cross match found mac addresses in the previously stored text file. Then it makes a new text file writing the name of the students whose devices mac address were found.

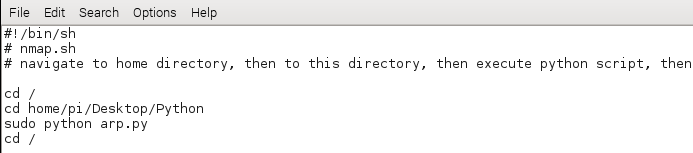
This program does the ARP and saves information in a log file

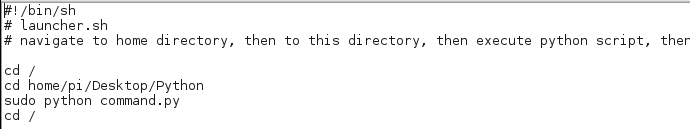


Here is the python program for doing cross matching the log file with the file containing stored student information



Script files are written to run these programs at startup





Now the raspberry pi can take attendance and save it as a text file in attached usb storage.

**Working procedure:**

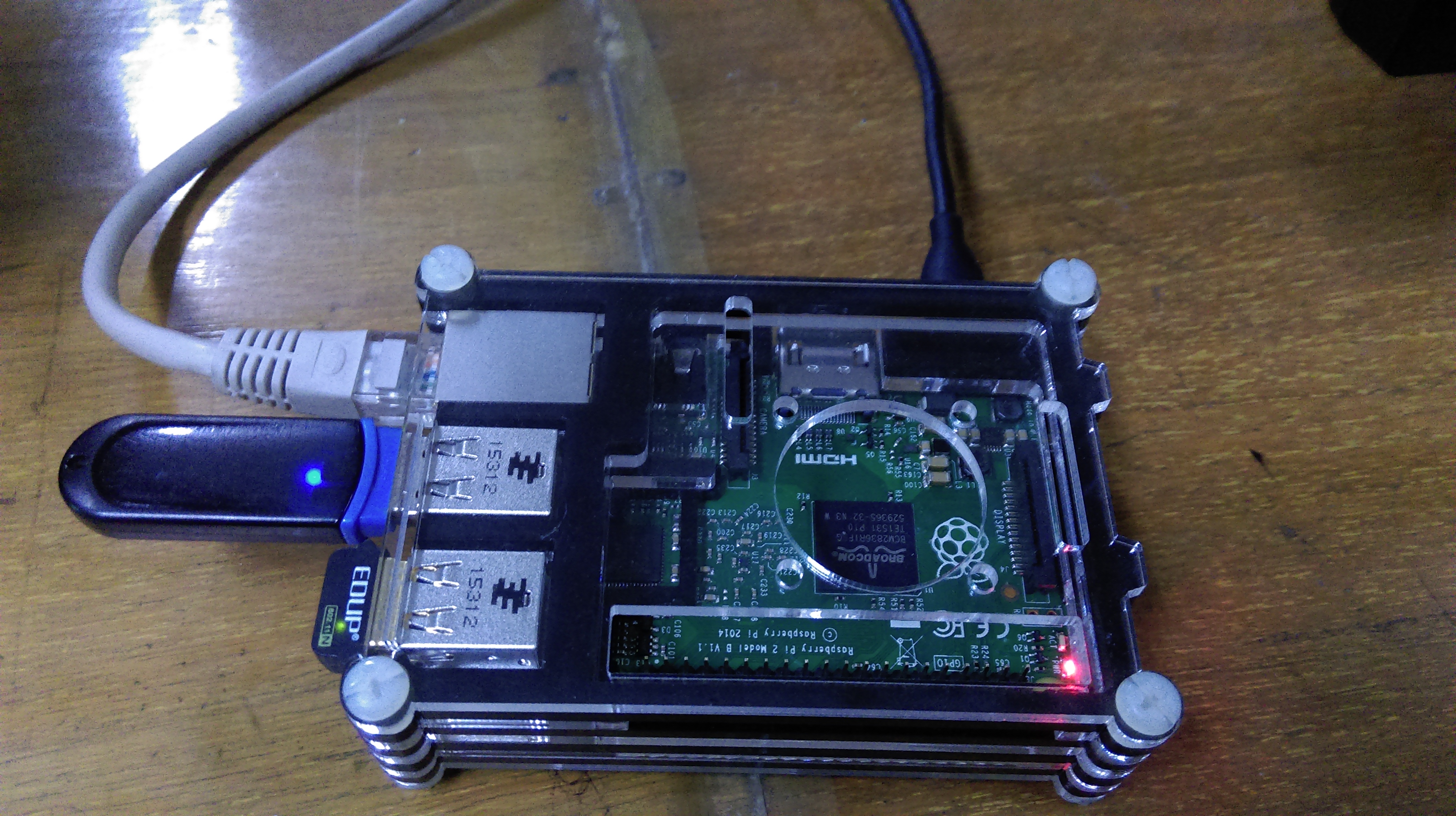
The process and set up may seem difficult but the working procedure is absolutely easy. With a simple direction any person having the basic idea of operating a device can use its function. Here is a brief idea about how it works:

1. **For attendance**: User need to save the mac addresses and their corresponding student id/name in a specific text file. After that whenever students turn on Wi-Fi it will match and take the attendance and save it in a file in external hard drive.
2. **For accessing TSR**: An IP will be provided. After connecting to the device using that IP users will access TSR.
3. **For media streaming:** Connecting to the device user will find the location of the multimedia by using a given IP address.
4. **For network sharing:** If the device is connected to an internet connected Ethernet cable the users will find its name on their Wi-Fi list. After giving the password they can use the internet.

So, after all it seems easier to use all those features.

**Result:**

After testing the results we found that our project gives almost 100% accurate result. It takes the attendance correctly. It provides media streaming, internet sharing and TSR access.



**Drawbacks**: It cannot be said that it is a drawback but due to unavailableness of very powerful wireless network adapter we had to use a medium range network adapter.

**Conclusion:**

As mentioned earlier our main motive was to solve problems and make things more efficient we tried our level best to bring out the accurate result. In almost all the cases we became successful. But we faced certain problems to accomplish the goal. First of all for buying all the necessary equipment we had to look for various places. As raspberry pi is not totally popular in our country it is not so popular. Then, raspberry pi 2 was very new to us all. We all with our members had to study a lot for setting up the raspberry pi. At last, we would say we will try to add some more effective feature to our project to make it more useful.

**Reference**

**Project**

https://www.youtube.com/watch?v=YgAgcHdPDdM

**Adding local proxy**

https://raspians.wordpress.com/2015/02/26/adding-local-proxy-to-raspberry-pi/

**DHCP problem**

http://sirlagz.net/2013/02/10/how-to-use-the-raspberry-pi-as-a-wireless-access-pointrouter-part-3b/

**hostapd**

https://learn.adafruit.com/setting-up-a-raspberry-pi-as-a-wifi-access-point?view=all

http://www.jenssegers.be/43/Realtek-RTL8188-based-access-point-on-Raspberry-Pi

**DLNA**

http://www.instructables.com/id/Raspberry-Pi-Media-Server-MiniDLNA/step5/Configuring-MiniDLNA/

**SAMBA**

https://www.youtube.com/watch?v=jr2Jz8km2BU

**Running python in crontab**

http://www.instructables.com/id/Raspberry-Pi-Launch-Python-script-on-startup/

**Accessing Shared networking**

https://rasspberrypi.wordpress.com/2012/09/04/mounting-and-automounting-windows-shares-on-raspberry-pi/

http://geeks.noeit.com/mount-an-smb-network-drive-on-raspberry-pi/

* No numbering or colon in points
* Picture in equipments
* Result.