# Running Raspberry Pi in Headless Mode

## Setting up Raspberry Pi for CSMWireless

1. Begin with Raspberry Pi turned on, and configured as mentioned in the set up packet distributed in class.
2. Open a terminal. This is the 4th icon on the top, or the black box with '>_' in it.
3. Type ‘ `cat /sys/class/net/wlan0/address` ’. This should print your MAC address, which we will need in the next step.
4. On another computer/device, go to netreg.mines.edu . Acknowledge the message at the bottom.
5. Fill in the information on this page. Your operating system will be “Raspbian”. The MAC Address is what we found in step 3.
6. Type in your MultiPass information and register your device.
7. After 5 minutes, type ‘ `sudo reboot now` ‘ into your Raspberry pi (minus the quotes).
8. Now you can click the wifi button on the top right hand side, and join CSMWireless.

## Enabling SSH on your Raspberry Pi

1. Go to the Raspberry Pi Configuration Menu (Menu Icon > Preferences > Raspberry Pi Configuration)
2. If you have not already done so, change your password. This is critical as the default password for Raspberry Pi is always the same, and you are opening your box up for anyone to mess with. We want it to be password protected. The default is ‘ raspberry ‘.
3. Go to the interfaces tab, and enable SSH.
4. Click OK. It may ask to reboot, this is okay. Go ahead and reboot.
5. Go back to a terminal window, and type ‘ `ifconfig -a | grep "inet addr"` ‘
6. Here you should see your IP Address. There may be two lines - use the one that is longer and more complicated. Typically the second one.

## Remotely Connecting to your Raspberry Pi.

1. First we need to connect to the Raspberry Pi from our personal computer. This differs between Mac, Linux, and Windows.

    Mac/Linux:
    1. Open up a terminal window. For Mac, this can be found in launchpad, or hold CMD and press the spacebar, type “`Terminal`” and type enter.
    2. Now you will type “ `ssh pi@XXX.XXX.XXX.XXX` “ inserting your IP address where the X's are.
    3. You may get an authentication warning, type ‘ `yes` ‘ and hit enter.

4. Type in your password that you changed to earlier in this process and hit enter. It will not seem like you are typing, this is a normal Unix feature. If you feel you made a typo, just hit backspace a bunch of times and try again.

Windows:

1. Install PuTTY from http://www.putty.org/
2. Launch PuTTY and type your IP Address into the destination server/hostname box.
3. Port number should be 22. This is the port for all SSH connections.
4. Click Connect. A dialog may pop up asking about adding the IP address as a host. Click yes. It will ask for a username and password. Username (unless changed) should still be pi. Password will be the new password you created for your Pi.
5. A terminal window should open in its place, and you are now connected to your Raspberry Pi.

2. Now lets test our connection. On your computer, type "`touch ~/temp.txt`" and hit enter.
3. Back on the Pi, type " `cd ~; ls` " and hit enter. You should see a " `temp.txt` " file there.

# Finding IP Address on Bootup

Because the way CSM has configured the wifi, the IP Addresses are not static. So if you unplug your Raspberry from MZ-022 and go try and plug it in over at Brown, the IP Address will have changed, and you can no longer access your Pi unless you hook it up to a monitor and find the IP address again. This defeats the whole purpose of working "headless". Instead, we will add a script at boot up that will send us an email with our IP address. So instead of finding it the hard way, when it turns on, it will send us an email and we just use that in PuTTY or Terminal just like we did before.

1. Back on your personal computer, download the attached file "`pifind.py`". Move this file to your Desktop. Open with Notepad, OpenEdit or another text editor (not MS Word or Pages), and edit in your email information. Your fromaddr will be your personal email (or a dummy email you can set up as well on gmail so you don't have your password to your real email on your Pi), the toaddr will be any email you can access, and the username and password will be your username and password to the "fromaddr" you entered. **Note: I have not tried this with our MyMail accounts. I would recommend making a dummy Gmail account. The "fromaddr" MUST be a GMAIL account.**
2. Save the file. Put this file onto a flashdrive and transfer that to /home/pi/ on your Raspberry Pi. If you know how, you can scp the file to your Raspberry Pi. For Mac/Linux, this command would look something like (without <> or quotes) " `scp <path_to_file>/pifind.py pi@<Raspberry_IP_Address>:/home/pi/` "
3. Once the file is on your Raspberry (Make sure this file is in the /home/pi/ directory), you can make it executable. SSH back to your Pi. Type "sudo chmod +x /home/pi/pifind.py". Hit enter. Try and test it! Type "/usr/bin/python /home/pi/pifind.py" and hit enter. Check your email!
4. Now we need to make it run on boot up. Type " cd /etc/ ". Hit enter. Lets open up write permissions on this file we are going to edit. Type "sudo chmod 777 rc.local " Hit enter. Open up "rc.local" in a text editor. On the Pi you can right click and click Text Editor.
5. Under all the comments, we need to add our script. First we need to type " sleep 60 ". This will give the Raspberry Pi some time to connect to the wireless network before we try to send an email.

6. Next, on a new line, we can add " /usr/bin/python /home/pi/pifind.py & ". This starts our email script, and the & puts it in the background. This is important, if the email script gets hung for some reason, it will not cause the rest of the Pi to stop booting.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
sleep 60
/usr/bin/python /home/pi/pifind.py &
exit 0
```

7. The last line should still be exit 0. Your rc.local should look like the image below (if there is some default things about printing IP Address, you can leave it, comment it or delete it. Doesn't make a difference):

8. Now see if it works! Type " sudo reboot now ". You should get an email. Under the wlan0 category, use the inetaddr as your IP address when you ssh to your Raspberry Pi.

# Using VNC on your Raspberry Pi

You are set for this course with SSH as described above. You can simply write all of your scripts/Python on the text editors through the command line, however that is a little painful and most of us like to use GUIs instead. Below I will show you how to port your entire display back to your laptop or desktop and you can interact with it as if you plugged it into a monitor.

1. Find a VNC viewer for your machine. I know that Mac OS has it built in, and I will describe how to use it later. For Windows/Linux, you may have to install on your own. The lab machines should have VNC Viewer. Feel free to find a tutorial for this online, it is pretty common. Just look up VNC Viewer and install it.
2. SSH to your Raspberry Pi as described above.
3. Lets make sure everything is updated and upgraded. Type " sudo apt-get update " hit enter. Type Y if asked. You will also need to type q at some point. Look for things to stop happening on the screen. Then type "sudo apt-get upgrade". Type Y if asked. This may take a long time. Our Raspberry Pi's have been sitting on a shelf for a while.
4. Now we want to install a VNC server on our Raspberry Pi. In the terminal, type "sudo apt-get install tightvncserver". Hit enter. Type y, enter if it asks.
5. Once it is finished, start the server by typing "tightvncserver". This will need to be started everytime we boot the Pi. If you'd like, you should be able to put it at the bottom of your /etc/rc.local file like we did with pifind.py. Note: doing this will always run tightvncserver, which does use CPU resources and might slow down other programs running.

6. Open up your VNC Viewer, and start a new session (File > New Connection). This step may differ depending on the VNC Viewer you use. Info you will need: Your normal IP Address we have used this whole setup, as well as Port 5901 for VNC if it asks. You can type anything for “friendly identifier”. If there is not a port box, you will need to type your IP address, followed by :5901 in the IP address/hostname box.

   With Mac OSX, you can open up a Finder window, and on the top toolbar, click “Go > Connect to Server”. You can now type vnc://**<IP Address of Raspberry Pi>**:5901 in the server address. Click connect and type in your Pi password.

That is it! You should see a window with your Raspberry Pi show up. Any questions or weird messages, feel free to email me at sheastyer@mymail.mines.edu. Try googling the error first, there are a lot of resources out there for linux commands and there is a good chance you forgot to take out the “<>” or something similar. If you do email me, make sure you include a screenshot or copy of both the command you entered as well as the error message.