LAB 01 – GETTING STARTED WITH RASPBERRY PI (RPI) BOARD

GOAL

The goal of this Lab is to explore the embedded system using a Linux-based system such as raspberry pi.

WHEN TO USE RPI

The RPi is perfectly placed for the integration of **high-level software and low-level electronics** in any type of project. Whether you are planning to build an automated home management system, robot, multimedia display, Internet of Things (IoT) application, vending machine, or Internet-connected work of interactive art, the RPi has the processing power to do whatever you can imagine of an embedded device.

WHEN NOT TO USE RPI

The RPi platform is not ideal for project developments that are likely to be commercialized. The Raspberry Pi platform largely utilizes open source software (there are some closed-source blobs used with the GPU), but it is not open source hardware. Schematics are available for RPi boards, but there is a lack of documentation on the hardware used. In addition, the Broadcom bootloader license3 explicitly states that its redistribution in binary form is only permitted if it will "*… only be used for the purposes of developing for, running or using a Raspberry Pi device.*" It is unlikely that such a license would transfer to a product of your own design.

The focus of the RPi Foundation is on education, and product commercialization is far from that brief. If you are planning to build an embedded Linux project that is to be commercialized, you should examine the BeagleBone platform, which is entirely open source and is supported by strong Texas Instruments documentation. If you are interested in learning embedded Linux using the BeagleBone platform, contact the instructor as he has the same labs/resources available for the BeagleBone platform.

PREREQUISITES

To follow this Lab you need:

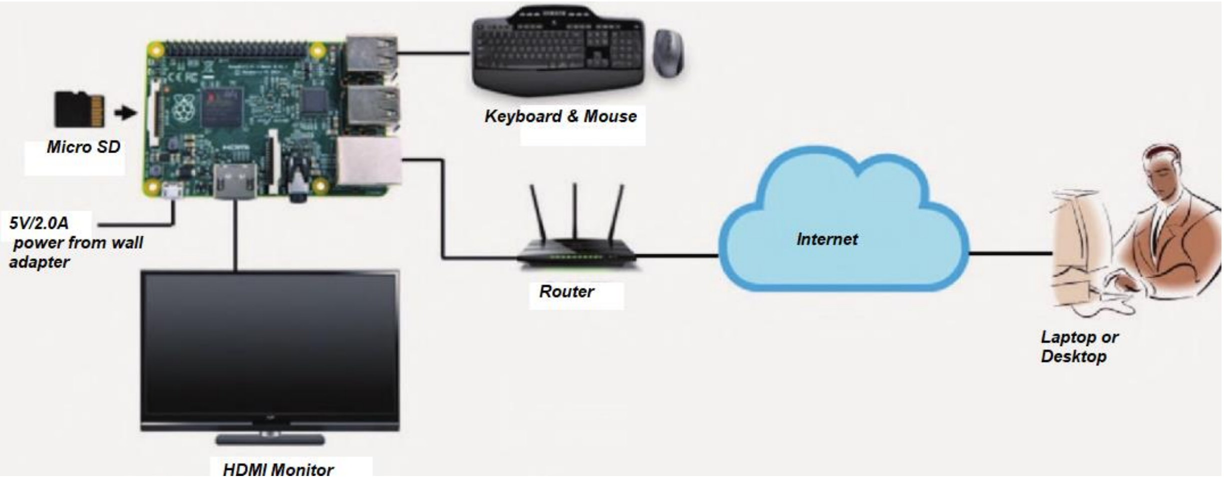
1. Raspberry Pi board
2. Micro USB cable
3. 8 GB Micro SD card with the NOOBS OS image
4. MicroSD card reader
5. A Window-Based PC, or a PC provided with a recent Linux installation (e.g., Ubuntu Desktop 14.04 LTS) or a virtual machine hosting a recent Linux installation;
6. Telnet/SSH terminal software (e.g., Putty download PuTTY)
7. (**Optional)** USB-to-Serial debug module for Raspberry Pi or USB to TTL adapter
8. (**Optional)** A Micro SD card reader attached to the PC/virtual machine
9. (**Optional**) Micro HDMI cable
10. (**Optional**) USB Mouse
11. (**Optional**) USB Keyboard
12. (**Optional**) HDMI Monitor/TV/VGA Adapter

BOARD'S NOOBS OS SETUP

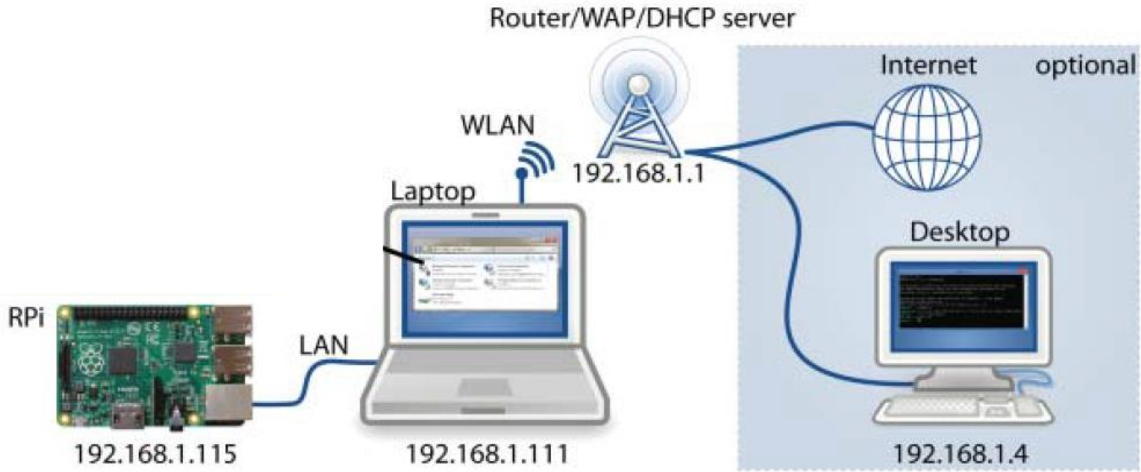
The NOOBS OS installation is recommended for your RPi. The tutorial at https://learn.sparkfun.com/tutorials/sd-cards-and-writing-images is available to give you a basic understanding the SD cards and how to write different images to the SD cards of your choice. Then, obtain NOOBS OS from raspberrypi.org and unzip the contents of the NOOBS zip file to the empty, formatted microSD card.

LABORATORY SETUP- NETWORK CONNECTION

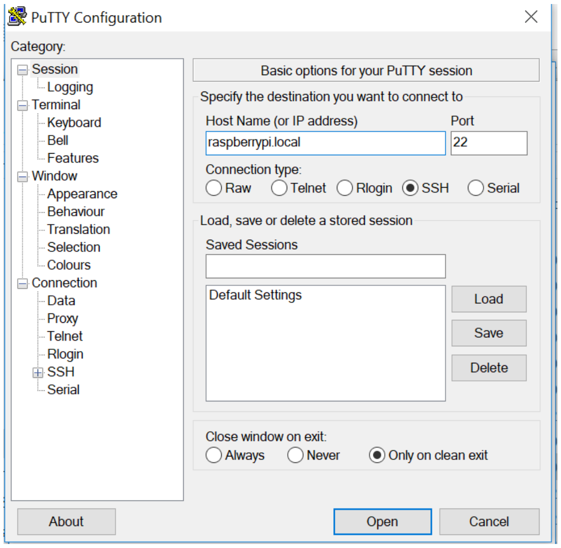
Option 1: connecting to wired network via a router



Option 2: sharing a network with laptop/desktop computer's network



Open PuTTY configuration software. Enter the Host Name **raspberrypi.local.** Select port 22 and connection type: ssh, then click the Open tab. A PuTTY interface console will appear. Enter pi to login to the RPi and enter the password raspberry.



Specify the destination you want to connect to

Host Name (or IP address): raspberrypi.local   Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

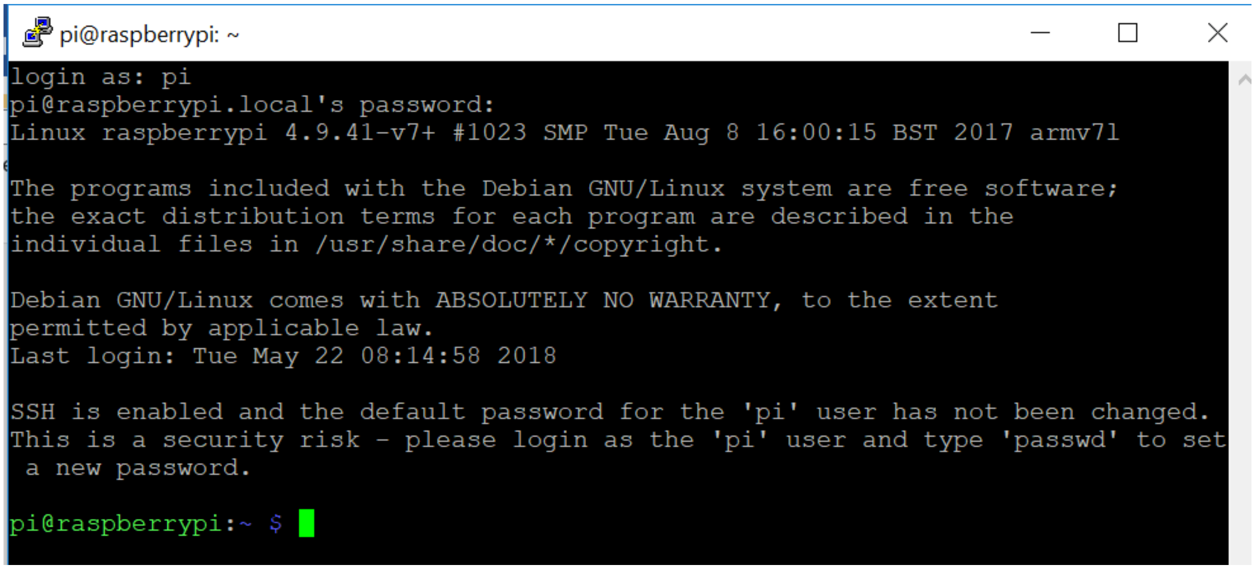
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

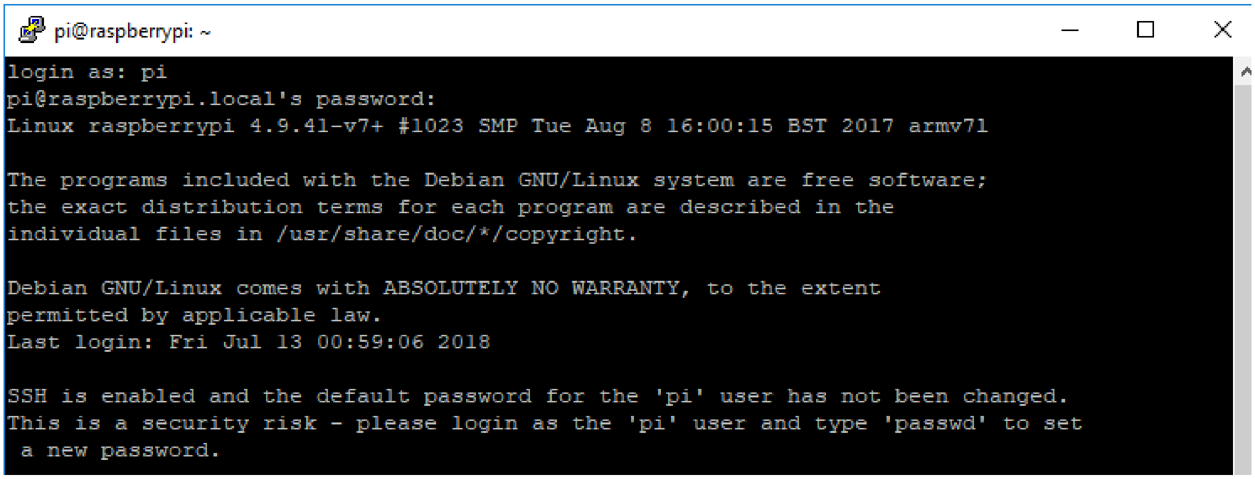
Open PuTTY configuration software. Enter the Host Name **raspberrypi.local.** Select port 22 and connection type: ssh, then click the Open tab. A PuTTY interface console will appear. Enter pi to login to the RPi and enter the password raspberry.



```
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 22 08:14:58 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:~ $
```

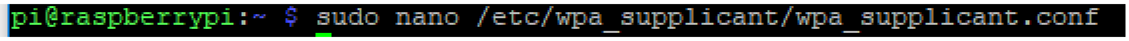
WIFI CONNECTION TO STOUTSECURE NETWORK

Use PuTTY to connect your RPi. Login to your RPi using the login name pi and the password (default: raspberry) as shown below.



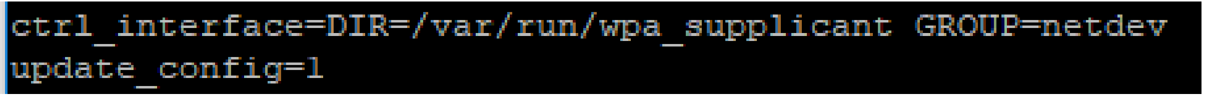
```
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul 13 00:59:06 2018

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
 a new password.
```

To begin, we'll need to enter in a text editor to enter some network information. You will need to be in root and use sudo for this as shown.

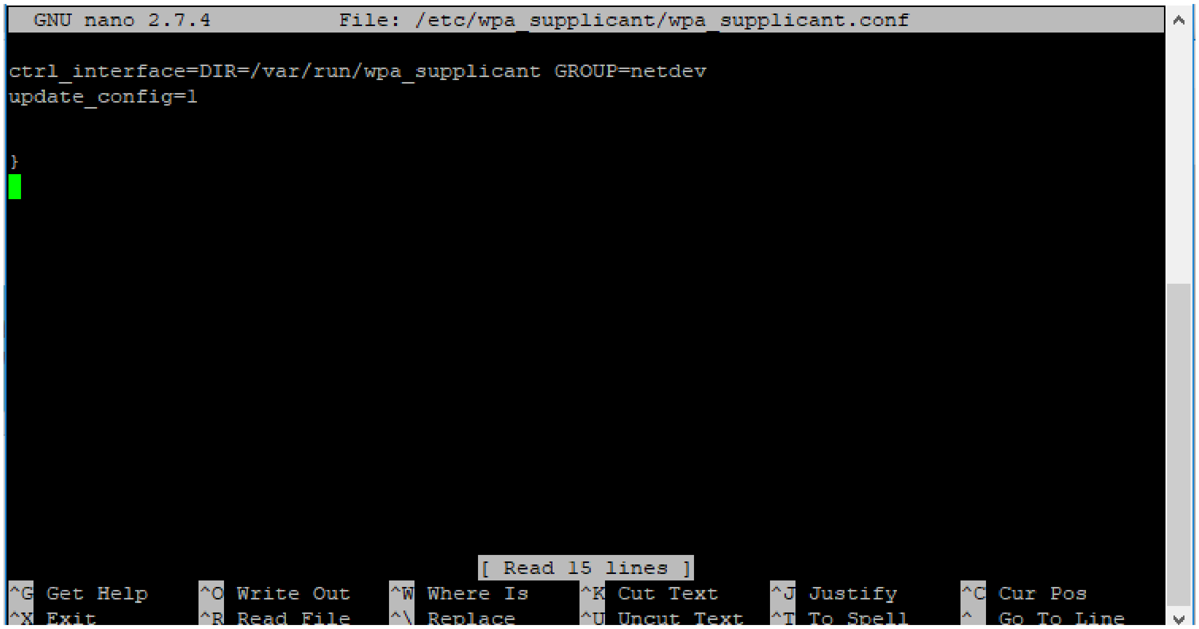


```
pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Once you are in, check if this is on the top of the page. If it is not, enter it in

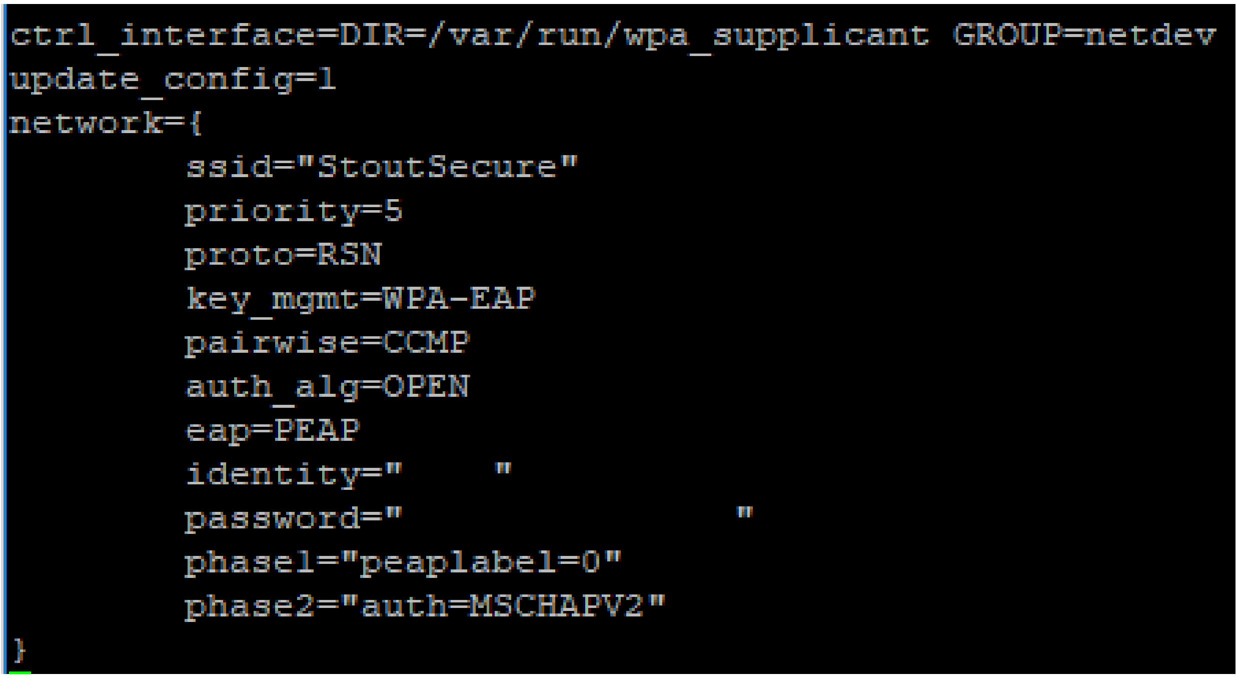


```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
```



```
  GNU nano 2.7.4            File: /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1


}

                              [ Read 15 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^  Go To Line
```

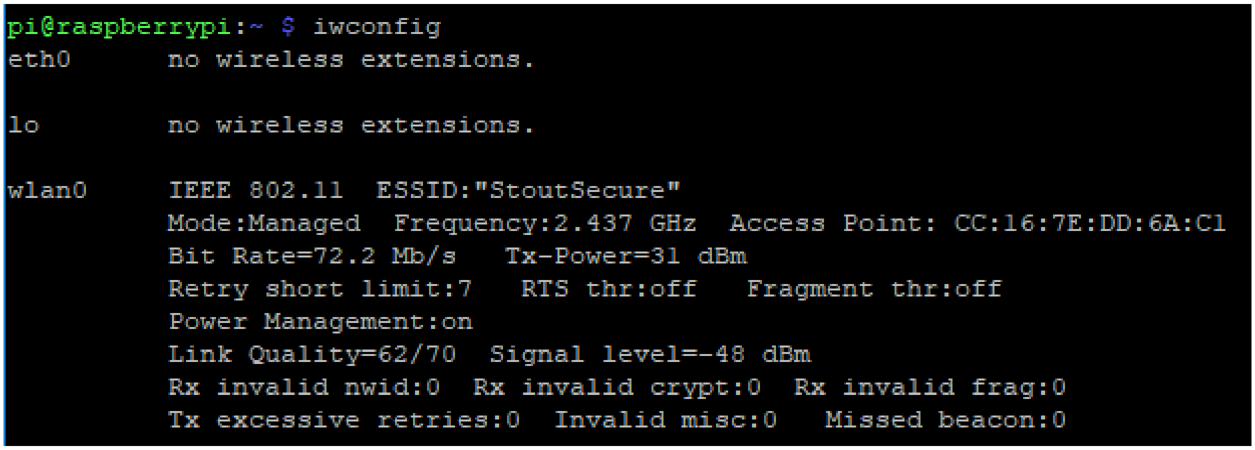
From there, this is where we enter the network configuration.



```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
        ssid="StoutSecure"
        priority=5
        proto=RSN
        key_mgmt=WPA-EAP
        pairwise=CCMP
        auth_alg=OPEN
        eap=PEAP
        identity="    "
        password="               "
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
}
```

Enter your username inside the " " for the identity and the same for a password. Don't put @my.uwstout.edu with it. Once you are done, hit "CTRL-O" to save and then "CTRL-X" to exit. Reboot the RPi and you should be connected to Stout Secure network.

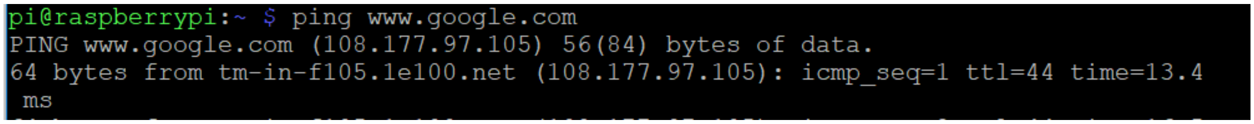
To test if you are connected to Stout Secure, type in this command iwconfig. It should show that you are connected to Stout Secure with wlan0.



```
pi@raspberrypi:~ $ iwconfig
eth0      no wireless extensions.

lo        no wireless extensions.

wlan0     IEEE 802.11  ESSID:"StoutSecure"
          Mode:Managed  Frequency:2.437 GHz  Access Point: CC:16:7E:DD:6A:C1
          Bit Rate=72.2 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=62/70  Signal level=-48 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Enter the Linux command date to verify if your RPi connects to the network successfully.

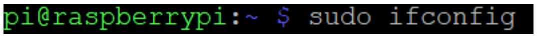
You can also use the command ping www.google.com to test if your RPi receives any data from the network.



```
pi@raspberrypi:~ $ ping www.google.com
PING www.google.com (108.177.97.105) 56(84) bytes of data.
64 bytes from tm-in-f105.1e100.net (108.177.97.105): icmp_seq=1 ttl=44 time=13.4
 ms
```

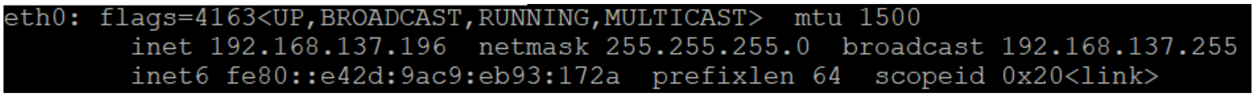
FINDING RASPBERRY PI IP ADDRESS (WHEN A NETWORK CABLE IS USED)

If you're using your Pi with a network cable, you can type ‘sudo ifconfig’



```
pi@raspberrypi:~ $ sudo ifconfig
```

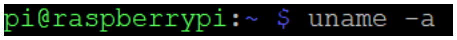
you will find eth0 with IP address (e.g., 192.168.137.196)



```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.137.196  netmask 255.255.255.0  broadcast 192.168.137.255
        inet6 fe80::e42d:9ac9:eb93:172a  prefixlen 64  scopeid 0x20<link>
```

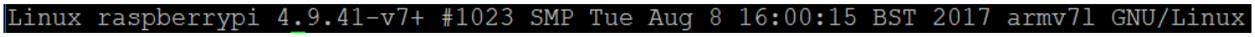
FINDING WHICH LINUX VERSION YOU ARE RUNNING

uname -a



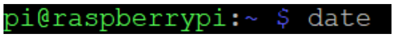
```
pi@raspberrypi:~ $ uname -a
```

You will see that message such as (Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv71 GNU/Linux)



```
Linux raspberrypi 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 armv7l GNU/Linux
```

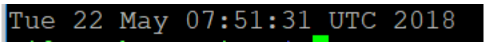
FINDING TIME AND DATE



```
pi@raspberrypi:~ $ date
```

pi@raspberrypi:~ $ date

You will see that message such as (e.g., Tue 22 May 00:43:25 UTC 2018)



```
Tue 22 May 07:51:31 UTC 2018
```

RASPBERRY PI SOFTWARE UPDATE

Begin by updating the repository package list:

pi@raspberrypi:~ $ sudo apt-get update

Wait for the Pi to be updated. When you’re done, type:

pi@raspberrypi:~ $ sudo apt clean

This will discard any unneeded files that have been downloaded as part of the upgrade. Finish by restarting:

pi@raspberrypi:~ $ sudo reboot

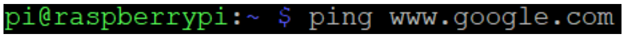
Or,

pi@raspberrypi:~ $ sudo shutdown -h now

and turn off the power and turn on the power on to reboot your Pi.

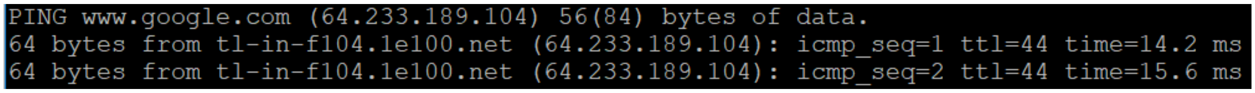
PING DATA FROM INTERNET

While the Raspberry PI is connected to the internet, you can use ping command to acquire data from the internet to view if data is received by the RPi.



```
pi@raspberrypi:~ $ ping www.google.com
```

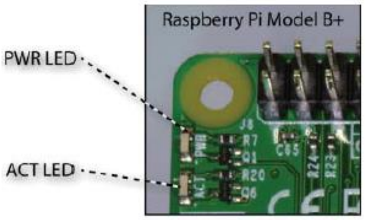
You will see that message such as,



```
PING www.google.com (64.233.189.104) 56(84) bytes of data.
64 bytes from tl-in-f104.1e100.net (64.233.189.104): icmp_seq=1 ttl=44 time=14.2 ms
64 bytes from tl-in-f104.1e100.net (64.233.189.104): icmp_seq=2 ttl=44 time=15.6 ms
```

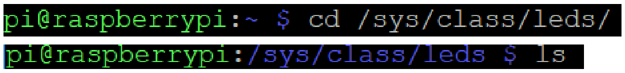
INTERACTING WITH ON-BOARD LED

The behavior of the RPi onboard userLEDs can be altered —the LEDs on the top left corner of the RPi board



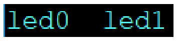
RPi board power LED (led0) and activity LED (led1)

You can change the behavior of these LEDs to test if we can override the led0 by writing 1 and 0, but you will temporarily lose this useful activity and power status information. *Sysfs* is a virtual file system that is available under recent Linux kernels. It would be useful to briefly explore the mechanics of how sysfs can be used to alter the behavior of the onboard LEDs. Using your SSH client, you can connect to the RPi and browse to the directory /sys/class/leds/



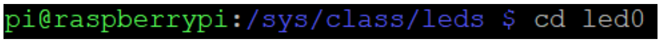
```
pi@raspberrypi:~ $ cd /sys/class/leds/
pi@raspberrypi:/sys/class/leds $ ls
```

The command should return two values: led0 led1



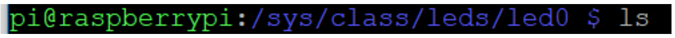
```
led0  led1
```

You can see the two LED sysfs mappings: led0 and led1. You can change the directory to alter the properties of one of these LEDs. For example, to alter the behavior of the ACT LED (led0):



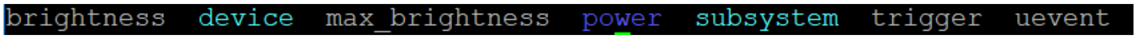
```
pi@raspberrypi:/sys/class/leds $ cd led0
```

Then type ls command.



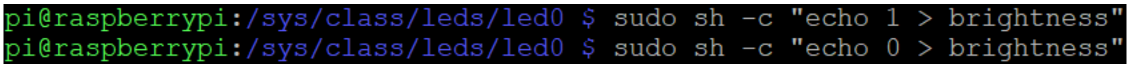
```
pi@raspberrypi:/sys/class/leds/led0 $ ls
```

It should return the following values.



```
brightness  device  max_brightness  power  subsystem  trigger  uevent
```

Here you see different file entries that give you further information and access to settings. You can turn the ACT LED fully on or off using:



```
pi@raspberrypi:/sys/class/leds/led0 $ sudo sh -c "echo 1 > brightness"
pi@raspberrypi:/sys/class/leds/led0 $ sudo sh -c "echo 0 > brightness"
```

where **sudo sh -c** is used to execute a shell command from a string command that requires superuser access.

SHUTDOWN AND REBOOT

Warning: physically disconnecting the power without allowing the Linux kernel to unmount the micro-SD card can cause corruption of your file system.

One final step to discuss in this lab is the correct shutdown procedure for your RPi, as improper shutdown can potentially corrupt the ext4 fi le system and/or lead to increased boot times due to file system checks. Here are some important points on shutting down, rebooting, and starting the RPi:

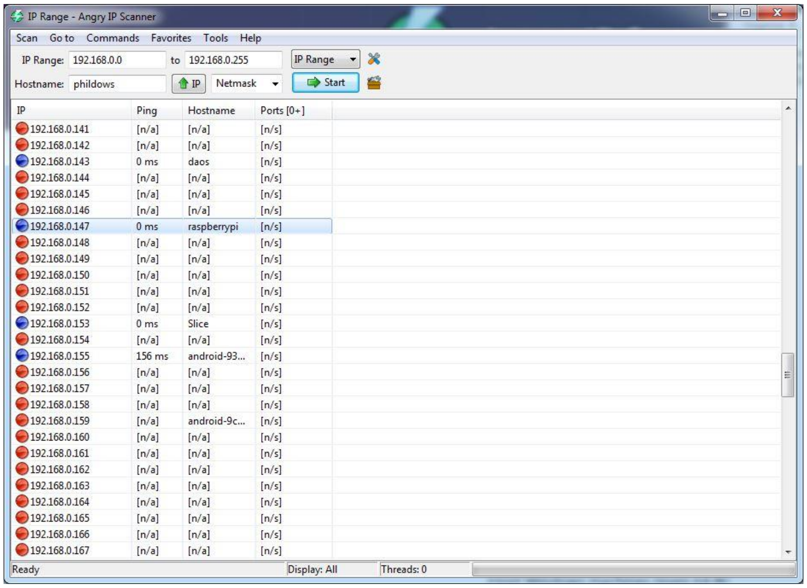
- Typing **sudo shutdown -h now** shuts down the board correctly. You can delay this by five minutes by typing **sudo shutdown -h +5**.

Observe the activity LED light- yellow color on the RPi board. When the LED turns off, you can safely remove the power cord and other cables from your RPi.

- Typing **sudo reboot** will reset and reboot the board correctly.

FINDING RASPBERRY PI IP ADDRESS FROM YOUR LAPTOP/DESKTOP COMPUTER

Finding your Raspberry Pi can be done by doing an "IP Scan" across your whole network. Most home networks will have about 254 possible combinations of IP address, so looking at them is an easy task for a computer. We use something called Angry IP Scanner http://angryip.org, this runs on Linux, MacOS X, or Windows and lets you scan your entire local network to find devices- including any Pi's you might be running. When you run Angry IP Scanner it should automatically pick a sensible IP Range for you- all you need to do is press Start and keep an eye out for your Raspberry Pi.



IP Range: 192.168.0.0 to 192.168.0.255

Hostname: phildows

Netmask

IP	Ping	Hostname	Ports [0+]
192.168.0.141	[n/a]	[n/a]	[n/s]
192.168.0.142	[n/a]	[n/a]	[n/s]
192.168.0.143	0 ms	daos	[n/s]
192.168.0.144	[n/a]	[n/a]	[n/s]
192.168.0.145	[n/a]	[n/a]	[n/s]
192.168.0.146	[n/a]	[n/a]	[n/s]
192.168.0.147	0 ms	raspberrypi	[n/s]
192.168.0.148	[n/a]	[n/a]	[n/s]
192.168.0.149	[n/a]	[n/a]	[n/s]
192.168.0.150	[n/a]	[n/a]	[n/s]
192.168.0.151	[n/a]	[n/a]	[n/s]
192.168.0.152	[n/a]	[n/a]	[n/s]
192.168.0.153	0 ms	Slice	[n/s]
192.168.0.154	[n/a]	[n/a]	[n/s]
192.168.0.155	156 ms	android-93...	[n/s]
192.168.0.156	[n/a]	[n/a]	[n/s]
192.168.0.157	[n/a]	[n/a]	[n/s]
192.168.0.158	[n/a]	[n/a]	[n/s]
192.168.0.159	[n/a]	android-9c...	[n/s]
192.168.0.160	[n/a]	[n/a]	[n/s]
192.168.0.161	[n/a]	[n/a]	[n/s]
192.168.0.162	[n/a]	[n/a]	[n/s]
192.168.0.163	[n/a]	[n/a]	[n/s]
192.168.0.164	[n/a]	[n/a]	[n/s]
192.168.0.165	[n/a]	[n/a]	[n/s]
192.168.0.166	[n/a]	[n/a]	[n/s]
192.168.0.167	[n/a]	[n/a]	[n/s]

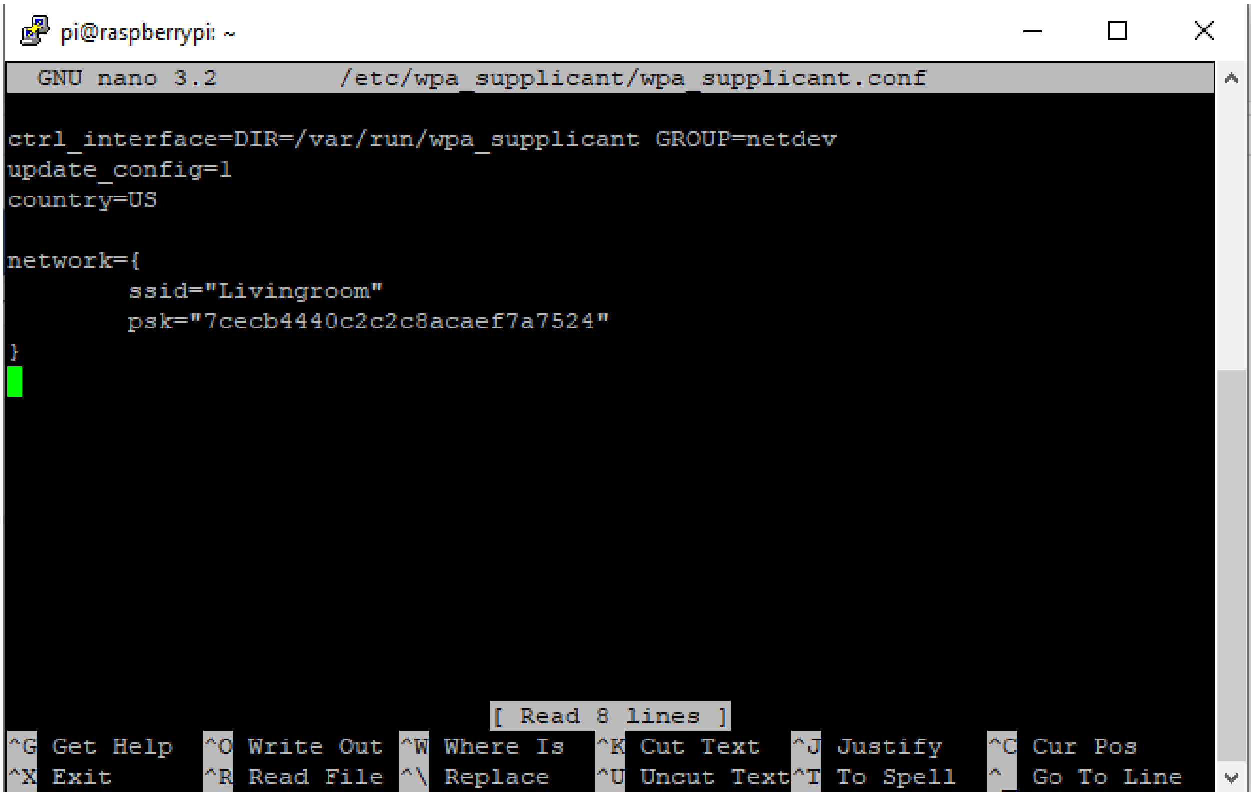
Ready

Display: All

Threads: 0

As you can see above, Angry IP Scanner found my Pi on the local network with its default hostname of "raspberrypi". Your Raspberry Pi may have a different IP address depending on whether it's connected to WiFi or Ethernet, and that address might even change from time to time. If you ever find yourself unable to connect, you can always double-check!

Setting up a network at your home:



```
  GNU nano 3.2           /etc/wpa_supplicant/wpa_supplicant.conf

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=US

network={
        ssid="Livingroom"
        psk="7cecb4440c2c2c8acaef7a7524"
}

                                [ Read 8 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

-END