EECS 2031A 2018

Lab #1 Imaging your Raspberry Pi

EECS 2031 Lab #1

Imaging your Raspberry Pi

This lab will introduce you to your Raspberry Pi and the process of imaging it. The primary goal of this lab is to get your hardware working. At the end of the lab you will have your very own Linux (DEBIAN) computer that you will use in the course. Feel free to customize it as you wish – its yours to play with. Also remember that in the worst case you will have to return to this document to re-image the hardware.

In order to complete this lab you will require your own copy of the necessary hardware. This lab assumes that you are using the nominal hardware described for the course. If you are using some other hardware you will have to adjust the procedures given here. Details on the components involved in the Raspberry Pi Zero can be found here. The

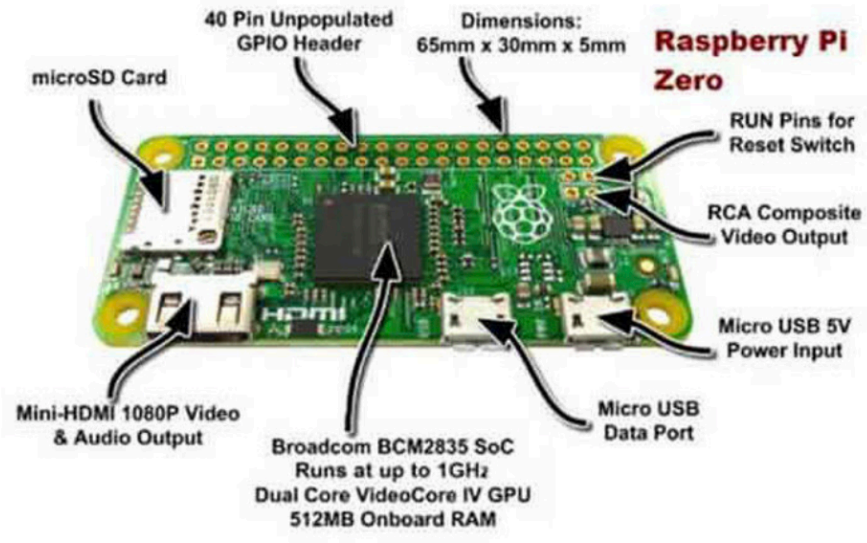


figure below shows the basic structure including the location of the power connector.

In order to complete this lab and obtain a grade for it you will have to demonstrate to your TA that you have successfully imaged your Raspberry Pi, have updated the software infrastructure so that it runs, is connected to the internet and that you have compiled and run the 'hello world' program. This lab is different from later labs as this lab is basically a simple recipe that you need to follow.

As with all labs in this course, the lab is intended to be experiential and collaborative. Feel free to ask for help from your fellow students and the TA's. Feel free to help others. The goal here is to understand the process and to end up with a working Linux Box running on your Raspberry Pi that connects to the internet using Air York.

Imaging the micro SD

You will need a PRISM account and a micro SD card with an adapter. We have prepared a modified version of the file 2018-06-27-raspian-stretch.img file that can be found on raspberrypi.org web site. The only modification has been through the addition of enabling a console on the UART connection (see here if you wish to do this yourself). You will also need a USB-SD Card reader/writer. The monitor in PRISM has a number of these to loan you. Or you can use your own. Note that there is nothing special about the hardware or software being used here and if you have your own Linux/OSX/Windows machine at home you can complete this entire lab on your own and bring the solution in to show the lab monitor to get the grade. One observation though, the disk image file is rather large so you may want to download it while on campus.

To begin the imaging process, log into PRISM using your PRISM credentials.

Borrow a microUSB reader/writer from the lab monitor. You may have to wait a bit here as there are only about 30 readers, so as soon as you are finished with the reader/writer please return it. The lab monitor will require you to leave some identification as security while you do this (your student card will suffice).

Insert your microsUSB card in the microUSB reader/writer.

Insert the microUSB reader/writer into one of the USB ports in your PRISM computer. Note: There may be ports attached to the monitor. If you can identify an open port on the actual computer this will be better.

You now have to find the system device associated with your microSD card. In the terminal window type

```
ls -l /dev/sd? | grep $USER
```

This should output exactly one line of output. If it does not, ask your TA for help. The line will be of the form

```
brw-rw— 1 foo disk 8m 16 Aug 20 09:37 /dev/sdb
```

The word foo will be replaced by your user name and the date may be different. The critical piece of text is at the end, here /dev/sdb. The actual value of this text can change — hence you running this command to find this text.

Now type the following commands in the terminal window

```
cd /tmp
wget http://dl.eecs.yorku.ca/2031/eecs2031a.zip
unzip -p eecs2031a.zip | dd of=/dev/sdb bs=4M conv=fsync
status=progress
```

(This last line should all be typed on one line.) Where the string /dev/sdb is replaced with whatever string you saw at the output at the end of the previous command. This command will take some time to complete. It will output a statement that the write has finished. It will look similar to

```
4802805760 bytes (4.8 GB) copied, 24.004525 s, 200 MB/s
0+73599 records in
0+73599 records out
4823449600 bytes (4.8 GB) copied, 496.076 s, 9.7 MB/s
```

After another minute the prompt will re-appear. At this point type

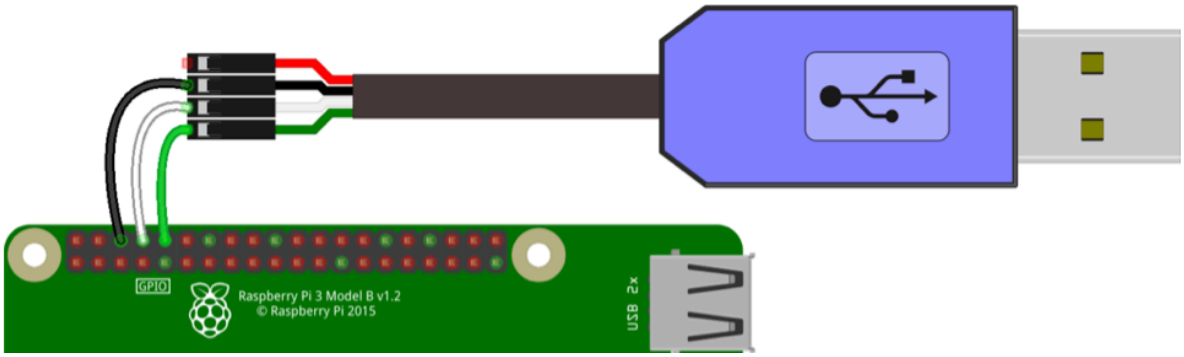
```
rm eecs2031a.zip
```

It is now safe to remove the microSD USB reader/writer from the PRISM computer, remove the microSD card from the reader/writer and return the reader/writer to the monitor to retrieve your ID. Insert the microSD card into your RaspberryPi.

Note: You can use any computer to do the imagining (the use of a PRISM machine not mandatory). Instructions for imaging from different OS platforms can be found on learn.adafruit.com.

Connecting your Raspberry Pi W Zero to a PRISM computer

This involves connecting the UART cable to your Raspberry Pi. This is a delicate task and if you get it wrong can lead to failure of your Raspberry Pi, or the UART connector, or both. So not something to mess up. The wiring you are looking to complete is shown below



Note that the red connector is not connected, and that the wires go to the outer rail. Orient your Raspberry Pi has the pins starting near the mounting hole in the upper left. Check your connection. Now plug the USB into an open USB port on your computer. (A PRISM workstation will work fine). Plug the micro USB power adapter into the power (see image earlier in this document for the location).

Resist the urge to plug the USB power adapter into the wall for a moment. Rather, first verify that all electrical connections are correct. Once you are convinced that your wiring is correct, plug your power adapter in. You should see a green LED flash on the Raspberry Pi. On the PRISM machine (instructions for other hardware options can be found here), open a terminal and execute the command

```
screen /dev/ttyUSB0 115200
```

(there will be no output, yet). Screen provides a terminal that connects to the USB port that is connected to your Raspberry Pi. Watch your PRISM workstation screen window. Eventually the green LED on the board will go solid you should see the screen show a login message and you can log in. If you don't see any output after the LED goes solid green, you may have hit enter to get the connection to start going.

```
Raspbian GNU/Linux 9 raspberrypi ttyS0
raspberrypi login: pi
Password:
Last login: Wed Jun 27 01:22:47 UTC 2018 on tty1
Linux raspberrypi 4.14.50+ #1122 Tue Jun 19 12:21:21 BST 2018 armv6l

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.
pi@raspberrypi:~$
```

Log into your machine. The default user name is pi and the default password is raspberry

If you see an output like that above, congratulations: You have successfully imaged your Raspberry Pi. Hopefully, you will never have to do this again in the course. But if for some reason you end up corrupting your microSD card you will have to follow these instructions again.

Note: To quit out of the screen program running on the PRISM computer type the sequence ^a^k (that is holding down the control key type a and then k). The screen program will ask you if you really wish to quit. Do so,

and you will be returned to the PRISM terminal window. You can reconnect to the console on your Raspberry Pi by running the screen command again.

Configuring your Raspberry Pi OS

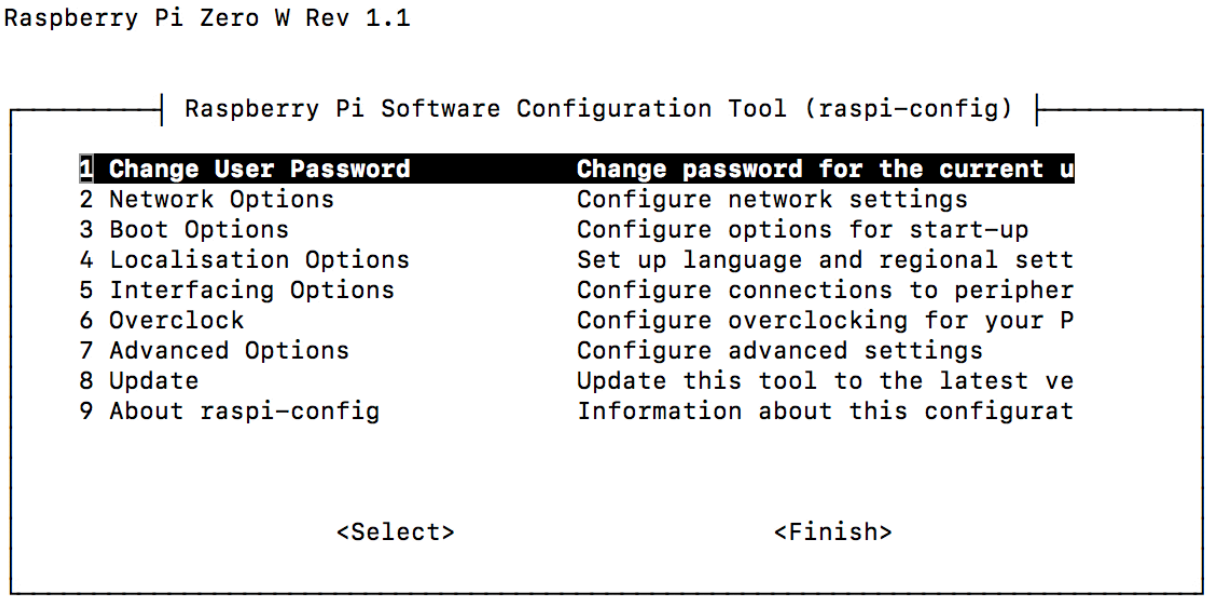
As shipped the Raspberry Pi OS is not complete and the OS is configured in a particular way. Some of these defaults are not ideal and you will change them now.

After logging in to your Raspberry Pi, execute the command

```
sudo raspi-config
```

This will run the command raspi-config as root. You will be prompted for the pi account password in order to run the raspi-config command as root. (sudo=execute a command as another user, here root).

The raspi-config tool uses the arrow keys, escape, and return keys to navigate a simple menu.



The tool is intended to be self-explanatory, and you should use it to

- Change the user password for the pi account to something else.
- Use the Network options to give your machine a different name.
- Use the advanced options tool to expand the file system.

Once you have made all these changes exit out of the raspi-config tool and reboot the system. Log back in using your new password.

Enabling WiFI

By default wifi is not enabled on your Raspberry Pi. Enabling wifi involves providing to the operating system instructions about what network to connect to and what settings to use.

Networking on a UNIX box can be incredibly complex. Here we are only interested in getting the WIFI card on the Raspberry Pi to talk to a small number of WIFI access points. The following commands will prove useful

- ifconfig – this tells you how your network connections are configured. In particular, you can use this command to test if your device is connected to the internet.
- sudo wpa_cli -i wlan0 reconfigure - reconfigures the wlan0 network connection and tells you if the wireless network configuration file is syntactically correct.
- ping 8.8.8.8 - this tries to connect to a well known machine on the Internet with address 8.8.8.8. If it works, great. If it doesn't, then you have learned something. You quit out of ping by typing ^c (control and c keys together).

The collection of wireless networks that the Raspberry Pi knows about is defined by the file /etc/wpa_supplicant/wpa_supplicant.conf. For each network that you wish to add you need to add an appropriate definition. The Raspberry Pi will connect to whichever one is available. (It is possible to do more sophisticated things than this, see the manual for details.)

Note: There is almost a 100% probability that you will not get this right on the first go. And if you do get it right – congratulations – you will get it wrong next time. So the first thing to do is to **make a copy** of the current wpa_supplicant.conf file. That is, in the wpa_supplicant directory execute

```
sudo cp wpa_supplicant.conf wpa_supplicant.conf.bkp
```

This makes a copy of the file as it is. So if you ever mess this up you can do

```
sudo cp wap_supplicant.conf.bkp wpa_supplicant.conf
```

And start again. Remember how long it took to image your microSD? This one line will save you having to do that again when you get this wrong.

There are three common types of wireless networks you might connect to. An unsecured network, a network secured with a site-wide password (typically something you would do at home), and a system like that requires a username and password. There are actually different protocols for negotiating these different systems, but that is not important right now.

Each network that you might encounter requires to be defined. You can use the really simple editor nano to do this. That is, to execute

sudo nano wpa_suppliant.conf

To edit the file.

For an unsecured network ‘UnsecuredReally’, add the following text at the end of the wpa_supplicant.conf file

```
network={
  ssid="UnsecuredReally"
  key_mgmt=NONE
}
```

Note that the blanks for indenting are not necessary but make it easier for you to read and that there is no punctuation after each line.

For a network “SiteWide” with a site wide password (typically wep- or wpa-based protocols), one would add at the end of the wpa_supplicant.conf file

```
network={
  ssid="SiteWide"
  psk="password"
}
```

Where the line psk= defines the password for the network. One obvious concern here is that you have stored your network password in a readable format in this file. This is critically important for your AirYork password (below). You will not want to have this information displayed on the screen at any time.

Finally, networks such as AirYork have a multi-phase security process and a correspondingly complex network definition.

```
network={
  ssid="AirYorkPLUS"
  key_mgmt=WPA-EAP
  eap=PEAP
  identity="PASSPORT-YORK-USERNAME"
  password="PASSPORT-YORK-PASSWORD"
  scan_ssid=1
}
```

Here PASSPORT–YORK-USERNAME is your passport name and PASSPORT-YORK-PASSWORD is your AirYork password.

Once you have made changes to your wpa_supplicant.conf file, execute

```
sudo wpa_cli -i wlan0 reconfigure
```

If there is a syntax error, this will tell you if there was one. Unfortunately, it will not tell you if you have entered a valid network configuration. Once you have a syntactically valid configuration, reboot (sudo reboot now) and once you log in, check your network connection. Note: It can take a minute or two after rebooting for your network connection to become enabled.

Hint: Debug the syntax with a fake password. Once its syntactically correct, insert your correct password. Your AirYorkPlus credentials are also used in a number of other places at York.

Updating the Raspberry Pi OS

Many UNIX's, including DEBIAN, use the apt tool to manage updates to system software. apt stands for advanced packaging tool, and more details on it can be found here. Basically, apt maintains a database on your machine of what packages have been installed using the tool and access web-based resources to learn what packages are available and

their current status. You use the apt tool to update your current packages, to install new system packages, and to remove packages you no longer need. The key task at the moment is to update your Raspberry Pi so that it is running the most recent version(s) of the standard software. To do this you execute the following commands

```
sudo apt-get update
sudo apt-get remove -purge wolfram-engine
sudo apt-get autoremove
sudo apt-get upgrade
```

These commands must run as root as they do things to the underlying OS. For these commands to work you must have access to the Internet. The update commands 'updates' your local machine's understanding of the current status of various software packages. The upgrade command 'upgrades' your machine to bring it into line with that understanding. The update command can consume substantive disk resources and can download a large amount of data. You will be asked a number of questions during the upgrade process - basically, do you want to install these packages. Answer 'y' to all questions. Installing some packages may require your Raspberry Pi to be rebooted. Note: It may take up to 30 minutes for your first upgrade to complete. Bring a book to read for your first lab.

Compiling and running the hello world program

Finally, compiling and running a program. Change your directory to your home directory and use the nano editor to create a file hello.c that contains the following

```
# include <stdio.h>
```

```
int main(int argc, char *argv[])
{
  printf("Hello world\n");
  return 0;
}
```

(hint, copy and paste this into the file). To compile and run this, type

```
gcc -Wall -ansi -pedantic hello.c -o hello
./hello
```

And your program should output Hello World

Logging out and shutting down your Raspberry Pi

To log out of your Raspberry Pi use the command

    logout

This does not power the machine off, rather it just logs you out. As you are likely to be the only user of your machine you are probably logging out to turn it off. The safest way of turning off your Raspberry Pi is to actually have the machine shut down gracefully, syncing files to the disk. The command to do this is

    sudo shutdown -h now

Which runs (as root) the command shutdown, with the argument h (halt) with the time now. So if you have logged out, log back in and execute this command to power down your Raspberry Pi.

LINUX tools introduced in this lab

- login - login to linux
- logout - logout from linux

- sudo - run a program as another user (typically root)
- shutdown - shutdown your computer
- raspi-config - configure your Raspberry Pi
- gcc - compile a c program
- ls - list files
- cd - change directory
- mkdir - make a directory
- nano - an incredibly simple editor
- ifconfig - configure a network connection
- ifup - check the status of a network

Grading

To obtain a grade for this lab you must show your TA your Raspberry Pi running the hello world program. You can do this in your lab, during any of the TA's office hours before your next lab, or at the very start of your next lab. Late solutions will not be accepted.