SOFE 4610U:
Internet of Things

Lab #1: Integrating sensors to a microcontroller board

Table of Contents

1. Objectives

- Learn the Fundamentals of Raspberry Pi 3 setup and get familiar with Osoyoo Esp8266 Arduino IoT kit along with HC Bluetooth module
- Gain experience of setting up a development environment
- Learn the foundation about Linux systems

2. Important Notes

- Work in groups of **four or three** students
- All reports must be submitted as a PDF on blackboard, if source code is included submit your report as PDF and source files as an archive (e.g. zip, tar.gz)
- Save the submission as <lab_number>_<first student's id> (e.g. lab1_100123456.pdf)
- If you cannot submit the document on blackboard then please contact the TA with your submission: David Lennick <david.lennick@ontariotechu.net>

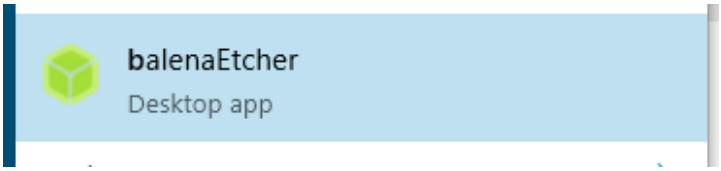
3. Lab Activity

3.1 Image setup (optional)

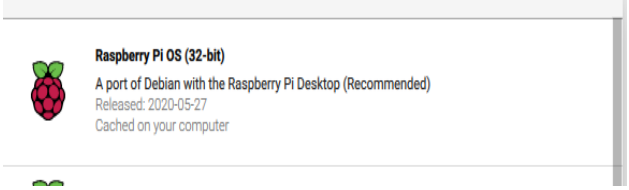
This image setup is only for student who is willing to practice how to download the image file on Raspberry pi. The Raspberry Pi is deployed with buster image. The process is quite straight forward, therefore It's not required to take the lab time to perform this activity.

1. You can setup the Raspberry Pi 3 in windows. The image file can be found the link https://www.raspberrypi.org/downloads/. The raspberry pi imager for Windows is used to download the raspberry pi image.

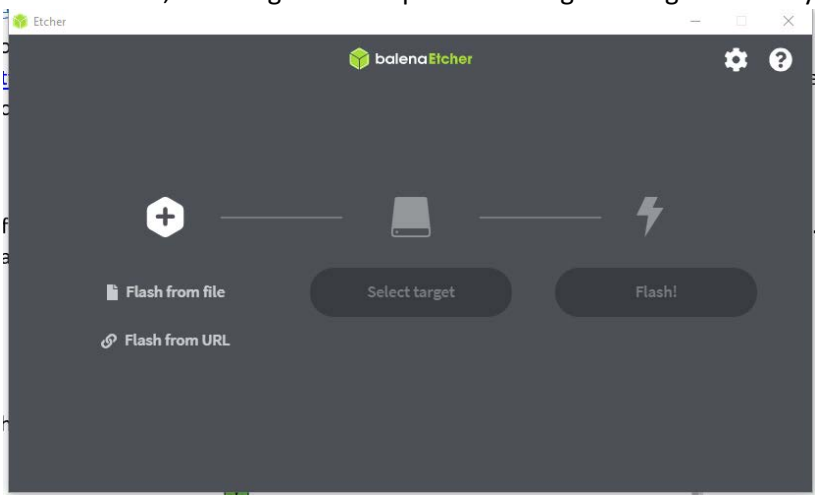
2. After download the balenaEtcher installation file, run it. Follow the on-screen steps to install.



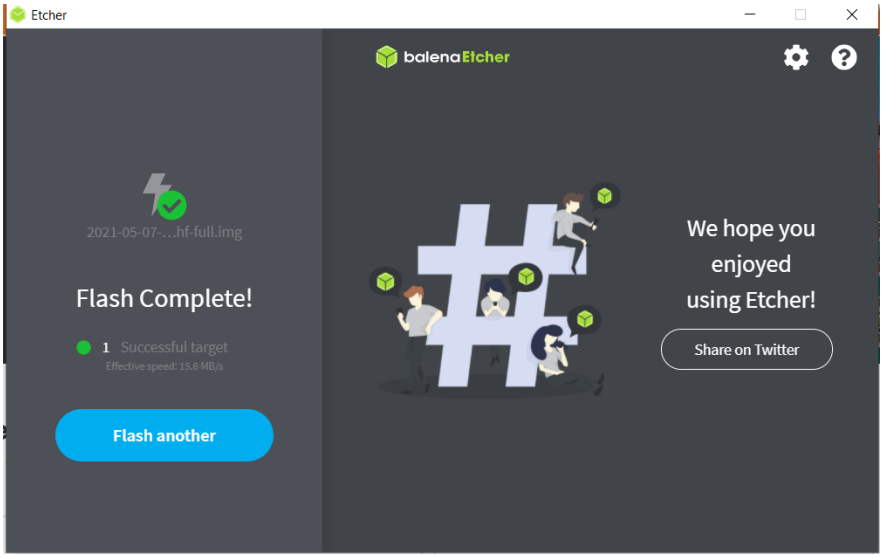
3. Download the following file



4. Click on Flash, the image will complete the image writing on SD for you.



5. The installation should take around 15-20 minutes, then you should see the following after verfication



The following setups have categorized with monitor and monitor-less

3.2 Raspberry Pi Network setup

1. Run the raspberry pi with HDMI connected to the monitor with a keyboard and mouse. Open the terminal, run the following command

```
$sudo iwlist wlan0 scan
```

It will generate a list the available network list that contain your network ssid.

For example, if you are on campus, you should see ESSID: CAMPUS WIDE as ESSID.

2. To gain access to the network for the Raspberry Pi, type the following command in the terminal

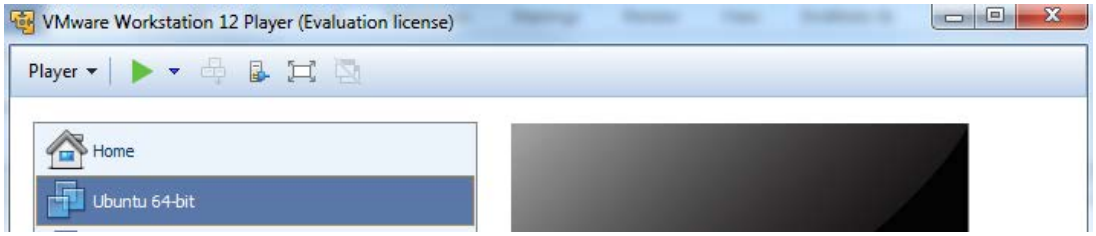
```
$sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

3. Add the following at bottom of the file

```
1.network={
2.ssid="SSID"
3.psk="WIFI PASSWORD"
4.}
```

3.3 VMWare setup

1. Begin by downloading **VMWare player** to your computer and installing it if your host computer is running Windows. (Virtual Box is another option for this lab).
2. Enter your email to use the evaluation version which is free for all academic use. Next, download the latest 64-bit Ubuntu.iso image file, and install Ubuntu in VMWare.
3. During the installation, you will be prompted to specify memory and virtual hard disk creation, ensure that you select the default settings by assuming that you have more than 2 GB memory spare on your hard drive.
4. Once installed, you will see the Ubuntu from the left side panel



*If you are using Ubuntu as your host machine, you are welcome to use virtual environment as well, so you don't have to worry about editing your path and changing files on your host machine

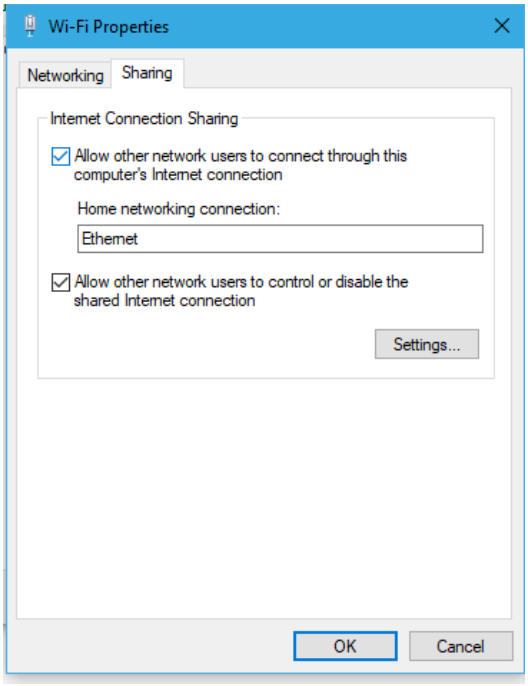
3.4 Windows host machine setup

To have stable Raspberry Pi connection, there are virtually two essential ways to hook up Raspberry Pi to the host machine. In this case, we will use Windows 10 as the host machine. The stable connection between the host and raspberry pi is essential to our raspberry pi development. However, if the host machine is Linux Ubuntu system, you could use ssh command in the terminal to connect with Raspberry Pi.

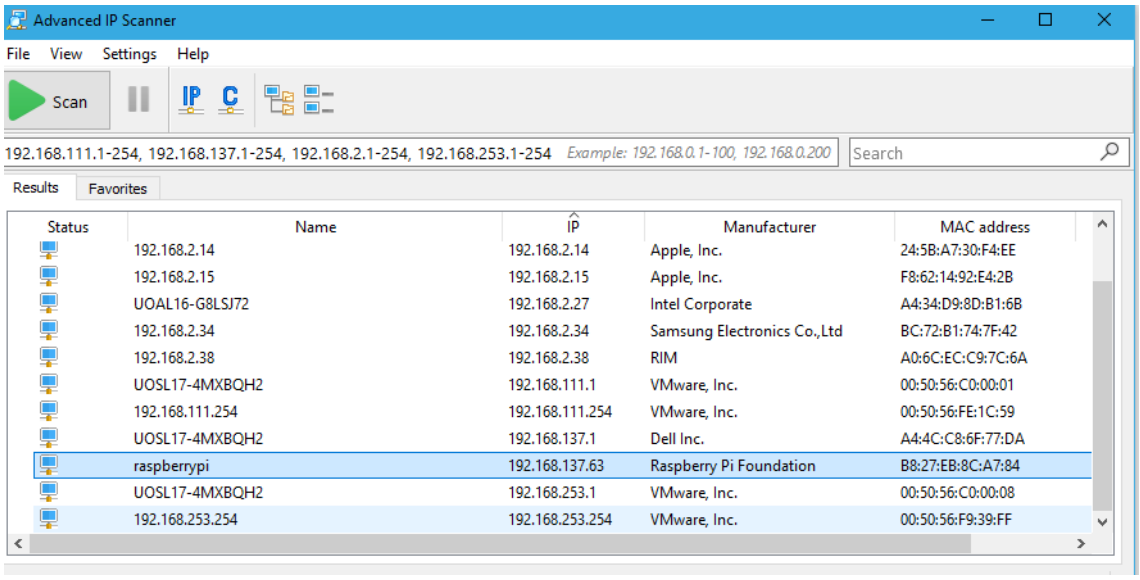
3.4.1 Method 1: Connect Pi to laptop via Ethernet cable

1. Connect the Raspberry Pi as shown with the Ethernet cable connect to your host machine

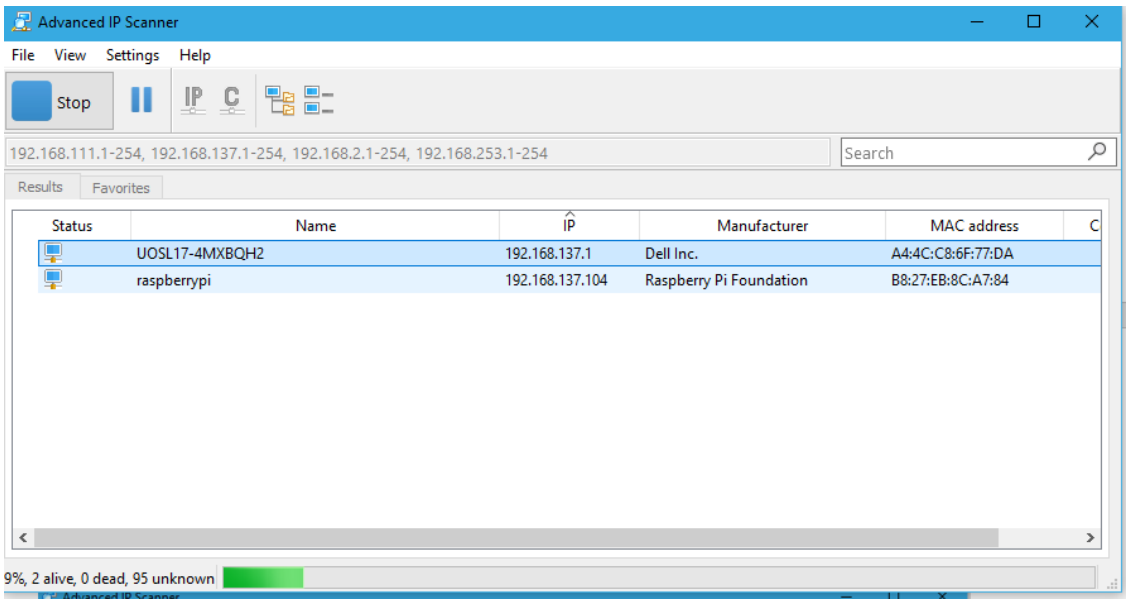
2. Go to WIRELESS change the sharing in the properties, use “Ethernet” (Make sure your windows 10 has been updated, if there is an update, you might experience network connection issues)



3. Find out the Pi’s IP address
   a) Use HDMI cable connects to a monitor
      Inside the pi terminal type “hostname – I” to find the IP address
   b) On the host machine, use “Advance IP scanner” (downloaded in the same folder) to scan nearby device, as shown below



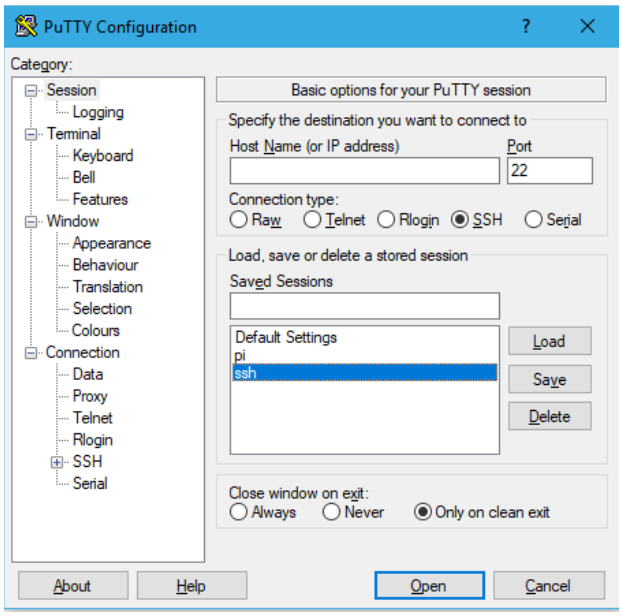
Try to find Pi's IP after the first time using Advance IP scanner, as follows, it appears that your laptop stays as 192.168.137.1, and pi changes to 192.168.137.104 (once use the internet sharing, the IP address of host laptop is always 192.168.137.1)



*Note: If you still couldn't connect to the pi, you will have to restart the raspberry pi device.

3.4.2 Method 2: Putty

Use putty, then use ssh to connect to the pi as shown




```
$sudo apt-get update //update packages
```

Use wired connection in school network or at home, you should be able to obtain the IP address of the Raspberry Pi (normally given to you before you are sitting in the lab or remotely logging in)

Use putty connection ssh shown above to connect with the raspberry Pi.

Once connected with Raspberry Pi, check the raspberry version

```
1.  $lsb_release -a
2.  $cat /etc/os-release
```

3.5 Raspberry Pi Linux Environment Basic

There are many websites introduce Raspberry Pi. In this experiment, we use Raspberry Pi as the platform especially Linux Environment to perform some programming. As previously demonstrated, the Pi can be remotely connected via a Windows machine, VM, Mac or any Linux machine as well.

In order to transfer files freely between the host machine and Raspberry Pi, you could download the WinSCP for the windows host, or use command lines from Linux machines. You will do some basic research on how to transfer file between Linux environment, and you will provide your research result in the lab deliverables along with the screen shots.

In this experiment, the user graphic interface (GUI) is not recommended. It's why you should use the terminal windows to connect the Raspberry Pi remotely after obtaining the IP address. It's many basic Linux commands to practice before or after the labs, and it's highly recommended that you should do that throughout the term.

A basic Linux command file is posted in the lab session on Canvas.

3.6 OSOYOO ESP8266 Arduino IoT kit and Bluetooth Module

This and following several experiments will be utilized OSOYOO ESP8266 Arduino IoT kit as the primary tool for Arduino programming and BLE setups and programming.

The detailed list could be found via this URL
https://www.amazon.com/s?k=arduino+wifi+kit&rh=p_4%3AOSOYOO%2Cp_78%3AB08P2LQPQ8&ref=nb_sb_noss_2

The primary components will be utilized will be Arduino R3, WIFI Shield and DTH11 sensor. More details about the Arduino introduction will be explained in the following experiment, however for whoever is already familiar with Arduino hardware and software setup, you can start out with care especially when you mount the Arduino WIFI shield on top of the Arduino. Please make sure that you DO NOT blend the pins.

HC05 Bluetooth module is another major component will be utilized for BLE programming.

HC-05 Bluetooth Module

3.7 Cross Compile between Raspberry Pi and VM Ubuntu

Write a simple C program print the word “Hello” under Ubuntu Linux environment and generate the executable file as the following steps.

* Step 2-7 on your host computer (Linux). Step 1 and 8 on Raspberry Pi

1. In the Raspberry Pi, find out the CPU version

```
$cat /proc/cpuinfo

$cat /proc/device-tree/model
```

2. Download the toolchain on Ubuntu by execute the following command

```
wget https://github.com/raspberrypi/tools/archive/master.zip
```

3. Now adding the path to the environment

```
$echo $PATH

Vim ~/.profile
```

4. Add the path to the end of the file, where /home/lwu is my home address, you should have your own.

```
PATH=”$HOME/tool/arm-bcm2708/gcc-linaro-arm-linux-gnueablihf-raspbian-x64/bin:$PATH”
```

```
source ~/.profile
```

5. Write the Hello.c //print Hello

```
#include <stdio.h>
int main() {
   // printf() displays the string inside quotation
   printf("Hello");
   return 0;
}
```

6. Use regular gcc command to compile the and code, the a.out (or whatever you name the executable file is a x86-64 file). Now compile the Hello.c code with arm toolchain.

```
sudo apt install gcc-arm-linux-gnueabihf

$arm-linux-gnueablihf-gcc Hello.c
```

a.out is generated

7. Transfer file to Raspberry pi using scp command under VM Ubuntu or using Winscp application on Windows platform

```
$scp a.out pi@ip address:/home/pi
```

store the a.out file to /home/pi.

You might need to be the super user to transfer the file, otherwise Pi4 will reject the connection.

8. Then on the raspberry pi, you should be able to execute ./a.out to print the string “Hello”.

4. Lab Deliverable

We have completed lab activities, and you should be familiar with cross compiling program. You should be able to work with both C and Python languages to write programs. Please submit the followings in your lab report for evaluation. Please include the steps and screen shots.

- Putty connection between your localhost and Raspberry Pi. (Steps and screen shots)
- Provide the Linux commands and example how to transfer file between Linux machine and Raspberry Pi.
- Cross compile between your local host and raspberry pi. (Steps and screen shots)

Please describe the difficulties you run into and explain how you tackle the problems and resolve them.

5. References

[1]
https://www.ti.com/lit/ds/symlink/cc2650.pdf?ts=1598042791485&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC2650%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%

2526utm_campaign%253Depd-null-null-GPN_EN-cpc-pf-google-wwe%2526utm_content%253DCC2650%2526ds_k%253D%257B_dssearchterm%257D%2526DCM%253Dyes%2526gclid%253DEAIaIQobChMI07D-rdnh6gIVgYvICh3DdAu4EAAYASAAEgLOX_D_BwE%2526gclsrc%253Daw.ds