Lab 0: Intro to running Jupyter Notebook on a Raspberry Pi

Nick Antipa, Li-Hao Yeh, based on labs by Jon Tamir and Frank Ong

January 24, 2018

This lab will walk you through setting up your Raspberry Pi to run a Jupyter Notebook server. By the end of this lab, you will be able to write code from a Jupyter Notebook in a browser on your laptop, then execute the code on the Raspberry Pi.

1 Quick intro

A Raspberry Pi is a tiny, inexpensive computer that uses ARM processors, just like most smart phones. The Linux-based operating system is installed on a micro SD card. We have provided SD cards with a functioning operating system, as well as a full installation of Python 3.5 and Jupyter Notebook. In this lab, we begin by using a USB-to-serial converter to open a terminal window on your Pi. From that terminal, you will secure your Pi with private passwords, connect it to WiFi, and establish an adhoc WiFi network. Once this is complete, you will be able to connect to your Pi wirelessly and write Python code from your own computer. In future labs, we will connect hardware to the Pi, but this lab focuses on simply setting up and executing code. Let's get started.

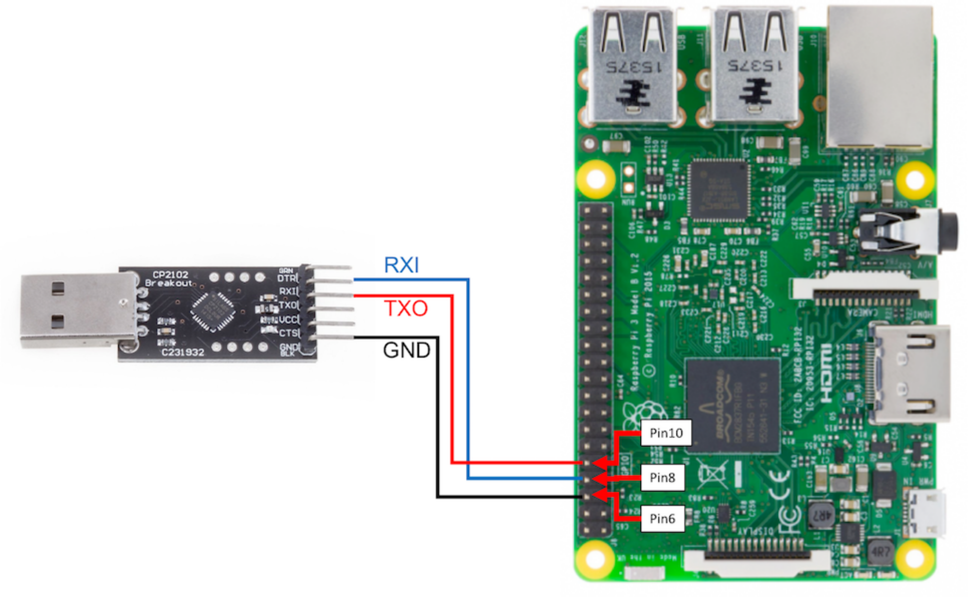


Figure 1: Wiring diagram for CP2102 to Raspberry Pi 3

2 USB-to-serial connection

The first step is to establish communication between your laptop and your Pi. We will do this via a serial connection over the CP2102 USB dongle.

1. Start by downloading and installing the drivers for your appropriate OS from `https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers`.
2. Next, use the provided jumper cables to wire the CP2102 to your pi. Connect RXI to Pi pin 8, TXO to Pi pin 10, and GND to Pi pin 6. See figure 1 [1]
3. Plug in the CP2102 to your laptop's USB-A port. Do not power up the Pi yet.

[1] Here is a website with more details if needed: `workshop.raspberrypiaustralia.com/usb/ttl/connecting/2014/08/31/01-connecting-to-raspberry-pi-via-usb/`

3 Open a terminal on the Pi

Next, we will establish a terminal window on the Pi. This will allow you to modify the settings of the Pi, including securing it with a private password. Do not forget your system password! If you do, we will have to format your SD card and you will need to repeat lab0.

1. Insert the micro SD card into the Pi
2. Use the provided USB-A to micro USB cable to power the pi. The micro-USB connects to the Pi and the USB-A to your laptop. The Pi will boot, indicated by the blinking green light.

3.1 Directions for Mac OS X

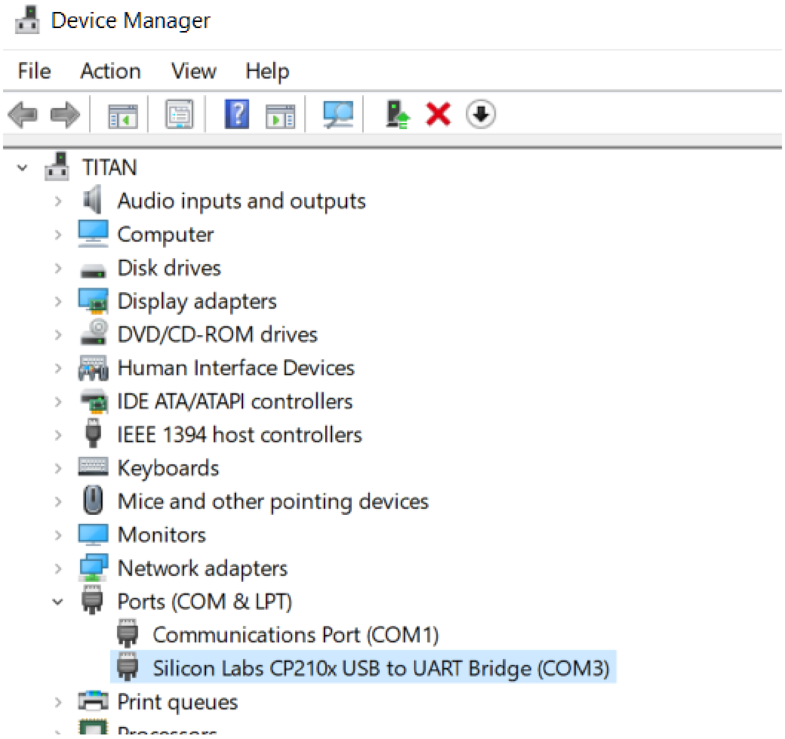
1. open Terminal (or your favorite terminal application)
2. Connect to the Pi by typing the command
   `screen /dev/cu.SLAB_USBtoUART 115200`
   If your Mac user name is longer than 24 characters, this will not work and you will need an alternate serial app.
3. The user name is `pi` and the default password is `EE123Rocks!`

You should now be logged into the Pi. If you have to reboot, leave this window open and it will automatically reconnect. If you close the window and try to connect again with the same command, it may not work. In that case, try `screen -r` to reopen the original session.

3.2 Directions for Windows

We will use a free program called PuTTY to open a serial terminal to the Pi.

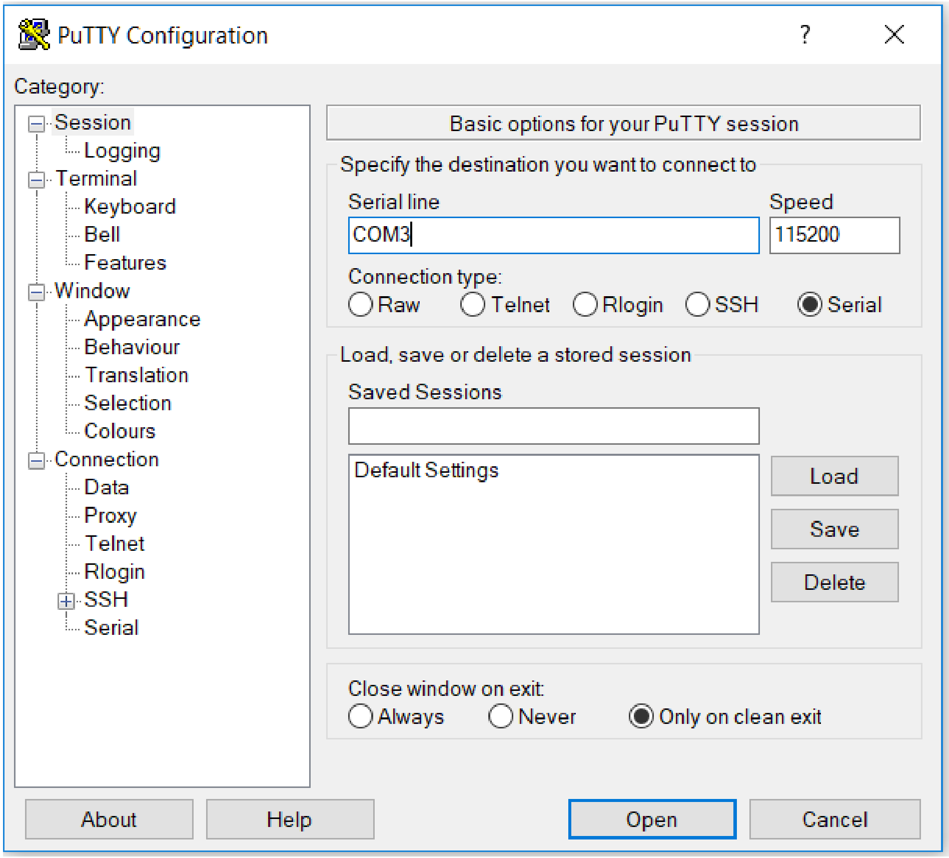
1. Download and install PuTTY from here, using default options: `https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html`
2. Open the Device Manger (search ”Device Manager”)
3. Under Ports (Com & LPT), find the Silicon Labs device and note its number in the form of COM# (COM3 in this example)



TITAN
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 host controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Communications Port (COM1)
  - Silicon Labs CP210x USB to UART Bridge (COM3)
- Print queues

Figure 2: Device Manger

4. Launch PuTTY
5. Select ”Session” in the left panel, and select the ”serial” radio button on the right.
6. in ”Serial line” enter your COM number (COM3)
7. Enter 115200 in the ”Speed” box. You window should look like Figure 3
8. Click ”Open”
9. Once the terminal window opens, press return once on your keyboard



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM3

Speed: 115200

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

Figure 3: PuTTY configuration

10. login with user name `pi` and password `EE123Rocks!`

11. If you were successful, you should see the command prompt.

4 Passwords

In this section, you'll set secure passwords for your Pi as well as the Jupyter Notebook server running on the Pi.

4.1 System Password

First, we need to change your password so others can't login to you your Pi. From your terminal window on the Pi (which should look like this: `pi@raspberrypi:~$`

From here, we need to expand the filesystem. Your SD card is compressed and has no usable space. After we complete this step, you will have 30 GB to work with.

1. open the Raspbery Pi configuration window using `sudo raspi-config` [2]
2. select "Advanced Options"
3. select "Expand Filesystem". It will say "Root System has been resized"
4. Press right arrow twice to select `<finish>`
5. when it asks if you want to reboot, say yes.
6. wait until your terminal window returns to the login screen, and enter user name `pi` and password `EE123Rocks!`

Now you have a fully functional Pi! Let's personalize and secure it. First, you'll reset the system password to something secure. This will keep people out of your Pi when you put it on the network (you will do this later).

1. `sudo raspi-config`
2. Select option 1, `Change User Password`
3. Set a secure password that only you and your lab partner know.

. Your system password is now set. Test it by rebooting and logging in with the same username as before, but your new password. **Do not forget this password! If you do, your SD card will need to be reformatted!**

4.2 Jupyter Notebook Password

This is also a very important step! You will need to set the password for your Jupyter Notebook. Using a simple password, or giving your password will be equivalent as giving full control of your Pi. We have provided a script to make this super easy for you. Simply type the command `python ~/set_password.py`, then enter a secure password. This is the password you will use to login to Jupyter Notebook from another computer.

[2] `sudo` is a command that gives you admin rights, and `raspi-config` is an interactive script for personalizing your Pi.

5 Setting up the network

You won't be able to use nice features of Jupyter Notebook like plotting over the serial connection, so in this section, you'll be guided through various methods for wirelessly networking with your Pi from another computer.

5.1 Connecting to WiFi

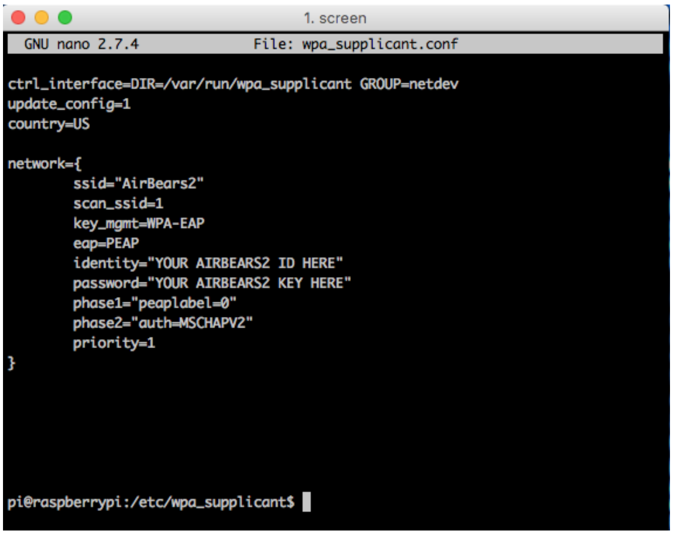
Connecting to most WPA-2 or unsecurred networks can be done easily via `raspi-config`. First, you need to put the Pi in manged Wi-Fi mode. This is done using a script conveniently provided by your amazing teaching staff. **This will not work for Airbears2; see section 5.2**. In WiFi mode, you will be able to connect from your laptop to the Jupyter Notebook server that is running on the Pi. Both the Pi and your laptop will still have internet.

1. Reboot in managed Wi-Fi mode by typing `sudo ~/wifi`
2. After the Pi reboots, log back in
3. Edit your dhcpcd.conf file (this stores DHCP/static IP info)
   `sudo nano /etc/dhcpcd.conf`
4. Comment out these two lines at the end of the file:
   `interface wlan0`
   `static ip_address=192.168.4.1/24`
   If you want static IP, uncomment these and enter static IP info. This should have been commented out by default.
5. Launch `raspi-config` (don't forget to use `sudo`)
6. Select `Network Options` → `Wi-Fi`. Type the SSID (name) of the network, then its password, then exit raspi-config.
7. Test that your are online by pinging any website you want `ping www.google.com`

5.2 Connecting to Airbears2 (optional)

Connecting to Airbears2 is optional. However, if you choose to use it, it will make it more simple to connect to the Pi and the internet while on campus. The process is a bit more complicated and requires editing a file that stores your Wi-Fi network information. This step is optional. Only proceed if you want to get your Pi online while on Campus. This will require storing your Airbears2 key in plain text, so you need to be OK with your partner seeing your key. You can always change your Airbears2 key later if need be. If you want to skip this, move on to section 6.1.

Your Wi-Fi network info is stored in a plain text file. The file provided already has a blank spot for Airbears2, and all you need to do is edit the file. Do this by typing `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` . [3] You'll see a section that looks like this:



```
  GNU nano 2.7.4                File: wpa_supplicant.conf

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

network={
        ssid="AirBears2"
        scan_ssid=1
        key_mgmt=WPA-EAP
        eap=PEAP
        identity="YOUR AIRBEARS2 ID HERE"
        password="YOUR AIRBEARS2 KEY HERE"
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
        priority=1
}




pi@raspberrypi:/etc/wpa_supplicant$ ▌
```

Replace the appropriate text, but do not remove the quotes! Exit by pressing `ctrl-X` . Enter `Y` when it asks `Save modified buffer?`, then hitting `return` .

5.3 Ad Hoc Network

According to Wikipedia, and ad hoc network "is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks." This method will allow you to wirelessly connect a laptop to your Pi without *any* other networking infrastructure. This is handy if you want to do your DSP labs on your Pi away from your home network. Your kind teaching staff have made this really easy! Note that the Pi will no longer have internet connectivity in this mode. **This method may not work for Windows users. If you have problems, proceed to section 5.4 if you have issues**.

1. execute `sudo ~/adhoc YourCustomSSID` (but replace YourCustomSSID with something unique)

2. after the Pi reboots, open your laptop's Wi-Fi and look for a network called EE123YourCustomSSID. Do not connect to this network at this time.

[3] nano is a text editor. You may use vi if you want to

5.4 Access Point (aka Managed) Mode

Using Access Point mode, your Pi will act like a router, enabling you to connect to it as you would a normal WPA2-securred WiFi network. This is similar to Ad Hoc mode, but is more secure and works with Windows. As with Ad Hoc, you will lose internet connectivity on the Pi while in this mode.

1. Update the SSID and WPA PSK (password) by editing hostapd.conf:
   `sudo nano /etc/hostapd/hostapd.conf`
2. Change the SSID to something unique by editing the line
   `sid=EE123ChangeThisName`
3. Set the password (to something secure!) by editing
   `wpa_passphrase=EE123Rocks!`
4. Save the file (ctrl-X, Y, return)
5. Boot into managed mode using `sudo ~/managed`
6. After reboot, log in.
7. You should now see a WPA-2 securred WiFi network named whatever you named it. Connect using your super secret password.

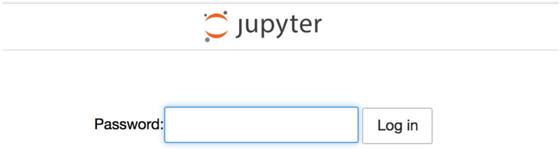
6 Connecting to the Jupyter Notebook Server

Your Pi is already setup to automatically start a Jupyter Notebook Server when it boots. In this section, you will see how to connect to this server from your laptop browser.

6.1 Ad Hoc/Managed Mode

With either Ad Hoc or Managed methods, you can connect to your Pi using a laptop without needing a Wi-Fi network. This is the most robust way to connect, but you will have no internet connectivity on *both your laptop and Pi.* **Ad Hoc may not work with Windows laptops. Use Managed mode if you have issues with Ad Hoc** To use Ad Hoc or Managed mode:

1. make sure your Pi is booted into adhoc or managed mode (see 6.1 and 5.4. Connect your laptop to the ad hoc network. Note: you will lose internet connectivity once you've done this.
2. In a browser window on your laptop, type `https://192.168.1.1:5555`
3. Your browser will probably tell you this is an unsecure, terribly dangerous thing to do. But you know better, so tell your browser you know what you're doing and proceed.



4. Enter the Jupyter password you set earlier into the prompt that looks like this:

You now have a Jupyter Notebook window on your laptop that can see code in the directory `~/EE123` on the Pi. It is empty, but you can put any code you want here in the future. N.B. at this point, you could remove the CP2102 and use ssh to connect to the Pi by typing `ssh pi@192.168.1.1`. We recommend that you create a folder for each lab, so your notebook and files don't get messy. In this case, create the directory Lab0

Finally, on your laptop, download the Lab0 notebook file. **For Mac**, from a terminal on your laptop, move it to the Pi by typing

```
scp [path_to_lab0] pi@192.168.1.1:~/EE123/Lab0/
```

Alternatively, from a terminal connected to the pi, you can use the command `wget http://file.url` to download the file using the commandline.

On Windows, PuTTY comes with a command line program called PSCP From a command prompt (search "CMD" to start), the syntax is `pscp source_file destination`. For example, to move a file from your PC to your Lab0 folder on the Pi in ad hoc mode, type

```
pscp [PathToFile] pi@192.168.1.1:/home/pi/EE123/Lab0
```

. You may also be able to transfer using a USB thumb drive, but we have not tested this so you're on your own!

6.2 Wi-Fi Mode

With your Pi and your laptop on the same wireless network (this is important), go to the command prompt on the pi. Type `hostname -I` to get the IP. Follow the instruction in 6.1, but replace 192.168.1.1 with your Pi's IP (you still need the :5555 at the end). You can also move files using `scp` or `pscp`, just replace 192.168.1.1 with the Pi's IP. Note: this may or may not work on Airbears2, but it should work fine on your home Wi-Fi.

That's it! You can now open lab0 and run your Python code as normal. When you are finished with the lab, shutdown the PI by typing `sudo shutdown -h now`

Important: Always shutdown the pi using the command `sudo shutdown -h now` **BEFORE** disconnecting the power. Failure to do so, may corrupt your file system.