Lab 1: Intro to Running Jupyter Notebook on a Raspberry Pi

Suma Anand, Alan Dong

based on previous work by

Gautam Gunjala, Josh Sanz, Li-Hao Yeh, Michael Chen, Nick Antipa, Jon Tamir, and Frank Ong

January 27, 2021

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.

0 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 microSD card. We have provided microSD cards with a functioning operating system, as well as a full installation of Python 3.7 and Jupyter notebook.

In this lab, we will walk you through three stages of setup:

1. Power on and connect to your Pi via Bluetooth
2. Configure your Pi to connect to your home Wi-Fi network
3. Configure Jupyter notebook

In future labs, we will connect hardware to the Pi, but this lab focuses on simply setting up and executing code. You can also check out this video to see a step-by-step guide. Let's get started!

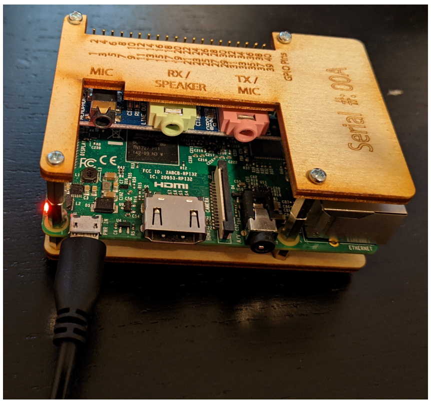


Figure 1: Raspberry Pi assembly with sound extension connected to power

1 Power on and connect to your Pi

1.1 Insert your microSD card

Take the microSD card out of the SD card adapter and insert it into your Pi's microSD card reader. This video shows how to do this. Make sure you do this BEFORE powering on.

1.2 Connect to power

The second step of this lab is to verify your hardware setup and connect the Pi to power. The Pi doesn't have built-in audio capabilities, so we've provided you with a sound extension board. It should be already attached to the Pi. Verify that it is attached, then connect the Pi to power via the micro-USB power cable. You should see a red light come on at the left once you have successfully connected to power (See Figure 1).

1.3 Connect over Bluetooth

First, you need to SSH into the Pi in order to configure it to connect to Wi-Fi. The simplest way to do this is over a Bluetooth connection. After you have successfully connected to the Pi's Bluetooth Personal Area Network (PAN), you will be able to SSH into the Pi at `172.16.0.1`.

Instructions for Mac:

1. Open `System Preferences → Bluetooth.`
2. Turn Bluetooth on and wait for the Pi's hostname (`raspberrypi`) to appear.
3. Click `Connect` to pair with the Pi.
4. `control+click` on the device and select `Connect to network.`
5. Connect to the pi via ssh by opening a Terminal window and running `ssh pi@172.16.0.1.` The password is `EE123Rocks!`

Instructions for Windows:

1. We will use a free program called PuTTY to SSH to the Pi. Download and install PuTTY from here, under "Package files".
2. Right-click the start button and choose `Settings → Devices`. After making sure Bluetooth is enabled at the top, select `Add Bluetooth or other device → Bluetooth`. Wait for your Pi's hostname (`raspberrypi`) to appear in the device list. Click on it to pair.

2a. Exit out of the device list window to go back to the "Bluetooth & other devices" settings and select `Devices and printers` from the menu on the right side or bottom (you may need to scroll).

2b. Alternatively, you can right-click the Bluetooth icon in the Windows taskbar and choose `Join a Personal Area Network` from the menu. Note that you may need to click the arrow to display more icons, and the Bluetooth icon will not be visible if Bluetooth is not enabled.

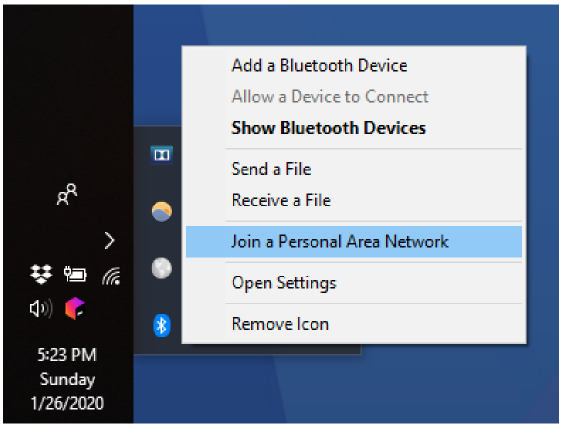


Figure 2: Navigating to Bluetooth devices from Windows taskbar (step 2b)

3. You should see a list of devices in a new window. Look for your Raspberry Pi in this list - its name will match the hostname you selected. You may need to wait for the list of devices to populate. When you have found it, right-click its icon and choose `Connect using → Access point`. A dialog box should appear and indicate that the connection was successful.

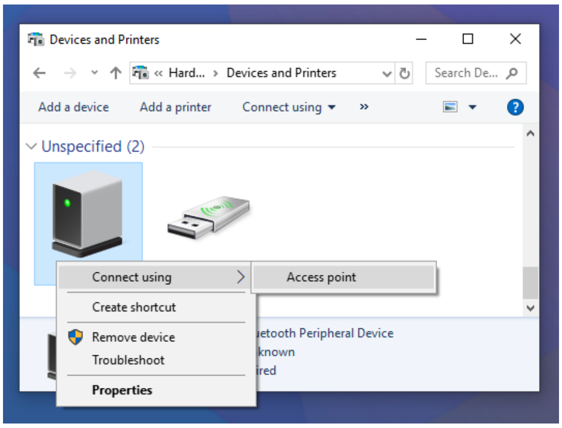
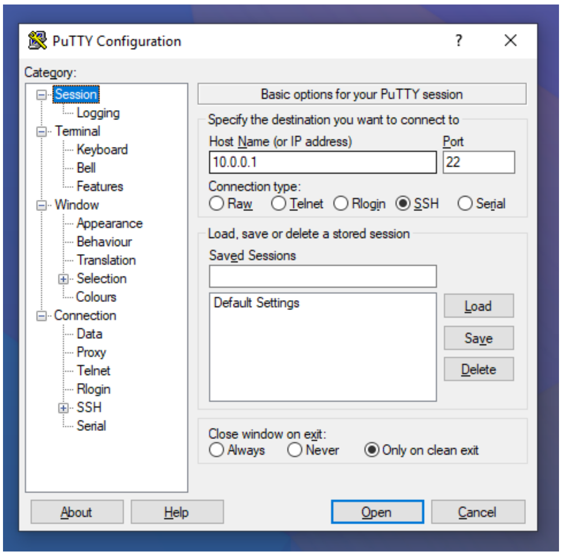


Figure 3: Establishing a Bluetooth connection to the Raspberry Pi (step 3)

3. Open PuTTY, click Session in the menu on the left side, and enter “172.16.0.1” in the box under `Host Name (or IP address)`. The connection type should be `SSH` by default. Once your window looks like Figure 4, click Open.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 10.0.0.1

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

Figure 4: Access the Raspberry Pi via SSH (step 3) - NOTE: enter 172.16.0.1!

4. A terminal window should open. When prompted, enter the username and password: the default user name is `pi` and the default password is `EE123Rocks`!

Linux Disclaimer

Although your brilliant course staff has confirmed that it is possible to complete much of the setup with a Linux machine, the variety of Linux environments and hardware support in the wild means that we cannot debug every problem that crops up and still maintain our sanity. If you wish to complete this lab with a Linux machine you are On Your Own™, but if you ask very nicely a staff member may be able to give you advice.

1.4 Debugging: USB-to-serial connection

If Section 1.3 worked, ignore this section and proceed to Section 2!

If you are unable to connect to your Pi over Bluetooth, you can connect serially over the CP2102 USB-to-UART bridge. This is a LAST RESORT - try the Bluetooth connection again, and do the serial connection if that doesn't work.

1. Shut down the Pi by typing `sudo shutdown –h now`. Always shutdown the Pi using the command `sudo shutdown –h now` BEFORE disconnecting the power or you may corrupt your file system. Then, disconnect the power.
2. Start by downloading and installing the drivers for your appropriate OS here.
3. Next, attach the sound extension to the Pi and then use the provided jumper cables to wire the CP2102 to your Pi (see Figure 5).
   a) CP2102 RXI → Pi pin 8
   b) CP2102 TXO → Pi pin 10
   c) CP2102 GND → Pi pin 6

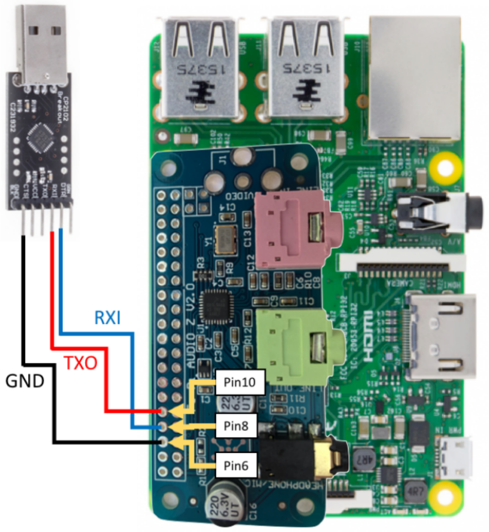


Figure 5: Fe-pi sound extension and CP2102 chip

Note that on some CP2102 chips, the TXO and RXI pins are in a different order.

3. Plug in the CP2102 to your laptop's USB port. Do not power up the Pi yet.

Use the provided micro-USB charger to power the Pi. Note that the USB serial bridge does NOT provide power to the Pi, and power is required. The Pi will boot, indicated by a blinking green light.

Directions for MacOS

1. Open Terminal (or your favorite terminal application).
2. Execute the command `screen /dev/cu.SLAB_USBtoUART 115200` to connect to the Pi.

   If your Mac user name is longer than 24 characters, this will not work and you will need an alternate serial app.
3. After starting screen, press return on your keyboard again or you will be stuck on an empty screen.
4. The user name is `pi` and the default password is `EE123Rocks!`

You should now be logged into the Pi. Your terminal window should say something like `pi@raspberrypi:~\$`. 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.

Directions for Windows

1. Open the Device Manager (search "Device Manager" or right-click the Start Button).
2. Under "Ports (COM & LPT)", find the Silicon Labs device and note its number in the form of COM# (e.g. COM3 in Figure 6)
3. Launch PuTTY.
4. Select "Session" in the left panel, and select the "Serial" radio button on the right.
5. Enter your COM number (e.g. COM3) under "Serial line", and enter 115200 under "Speed". Your window should look like Figure 7.
6. Click "Open". Once the terminal window opens, press Enter once on your keyboard or you will be stuck with a blank terminal window forever.
7. Login with username `pi` and password `EE123Rocks!` If you were successful, you should see the command prompt (which should look like `pi@raspberrypi:~\$`).

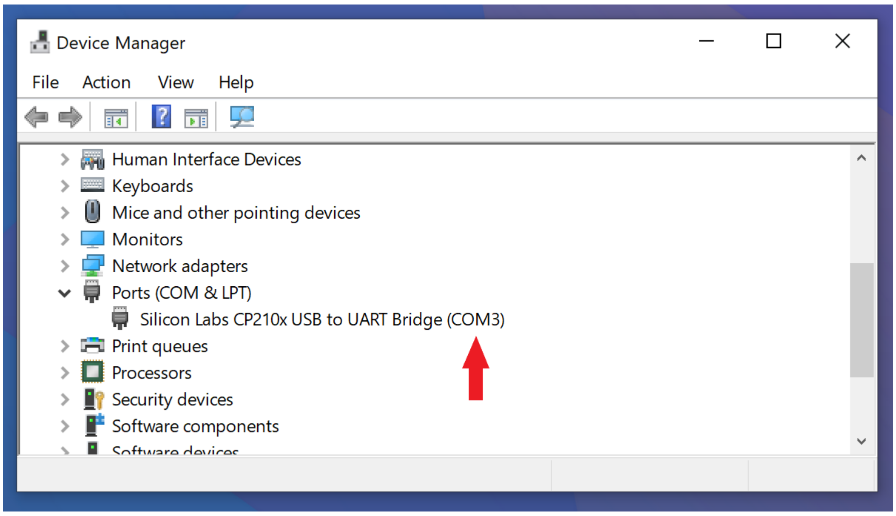
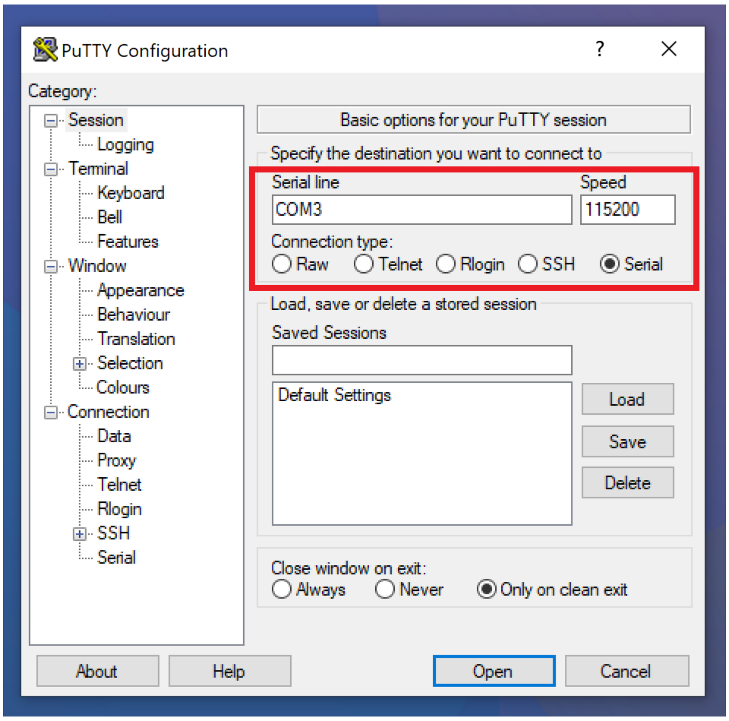


Figure 6: Device manager with recognized CP2102 USB-to-serial bridge



Specify the destination you want to connect to

Serial line: COM3

Speed: 115200

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

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

Figure 7: PuTTY setup window

2 Configure your Pi

This section will walk you through how to configure your Pi's file system, connect your Pi to the Internet, and change your password.

2.1 Filesystem expansion

Now that you can access your Pi's terminal, the first thing you need to do is expand your Pi's filesystem. When you first receive your Pi, the microSD card is compressed and has no usable space. After we complete this step, you will have approximately 30 GB to work with.

1. Open the Raspberry Pi configuration manager using `sudo raspi-config`[1]
2. Select Option 6, `Advanced Options` → `Expand Filesystem`. It will say "Root System has been resized. . . "
3. Press `<Ok>` , then press the right arrow key twice to select `<Finish>`
4. When asked if you want to reboot, select `<Yes>`
   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.
5. Wait until your pi is no longer connected. Re-connect via Bluetooth, and enter username `pi` and password `EE123Rocks!` once again.

You now have a fully functional Raspberry Pi! The next steps will help you set up, personalize, and secure your device.

[1] `sudo` is a command that gives you administrator privileges, and `raspi-config` is an interactive script for personalizing your Pi.

2.2. Connect to Wi-Fi

2.2.1 Connect to your home Wi-Fi network

You won't be able to use all the features of Jupyter notebooks (e.g. plotting) over the serial connection, so in this section, you'll be guided through how to connect your Pi to your home Wi-Fi network.

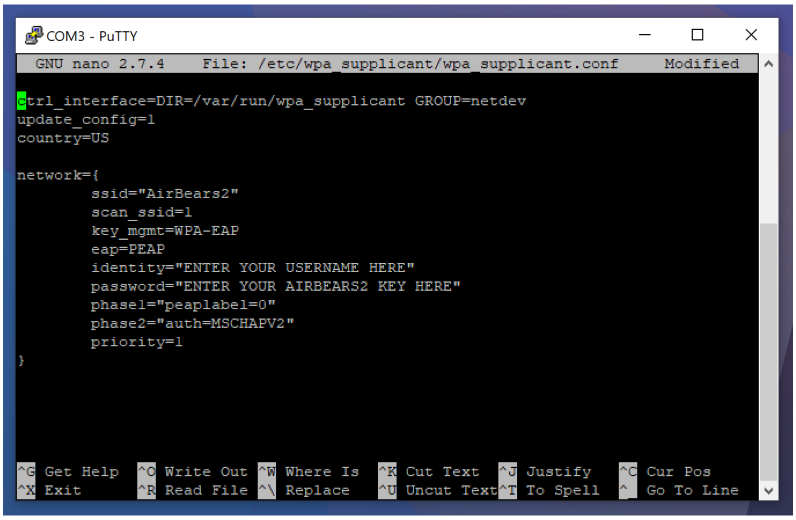
Connecting to most WPA-2 or unsecured networks can be done using `raspi-config`. First, you need to put the Pi in managed Wi-Fi mode. This is done using a script conveniently provided by your spectacular teaching staff.

1. Reboot in managed Wi-Fi mode by typing `sudo ~/wifi`
2. After the Pi reboots, connect through Bluetooth and run `sudo raspi-config`
3. Select `1 System Options → S1 Wireless LAN`. Type the SSID (name) of the network followed by its password, and then exit `raspi-config` by selecting `<Finish>`. Note: This will not work for AirBears2. See Section 2.2.2.
4. Test that you are online by pinging a website (e.g. `ping www.google.com`)

2.2.2 Connect to AirBears2 [Optional]

Connecting to AirBears2 is optional, but you may find it useful for connecting 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.

First, make sure your Pi is in Wi-Fi mode. Your Wi-Fi network info is stored in a plain text file that must be edited. This can be done by running the following command:
`sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`[2]
Your terminal should look like Figure 8. Replace text in the lines `identity="...` and `password="...`, but **do not remove the quotes!** Exit by pressing `Ctrl-X`. Press `Y` when prompted with `Save modified buffer?`, and then press `Enter`. Reboot your Raspberry Pi before testing your internet connection.



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

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="ENTER YOUR USERNAME HERE"
        password="ENTER YOUR AIRBEARS2 KEY HERE"
        phase1="peaplabel=0"
        phase2="auth=MSCHAPV2"
        priority=1
}

^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
```

Figure 8: Plain text file containing network information

[2] `nano` is a text editor. You may use `vi` if you wish.

2.3 Change your passwords and hostname

It is extremely important to keep your Pi secure with strong passwords. There are two passwords to set: the login password and the password to access the Jupyter Notebook. They can be the same password, but make sure they are secure. But first, let's change the login password of your Pi from the default.

2.3.1 Login password

You need to change your login password so others can't login to your Pi (especially when connected to a network).

1. Run sudo raspi-config
2. Select `1 System Options → S3 Password`, then select `<Ok>`
3. Set a secure password that only you know.
4. A screen will come up saying "Password changed successfully". Press `<Ok>` and then `<Finish>` to exit raspi-config.

2.3.2 Jupyter notebook password

Next, set the password for your Jupyter notebook. We have provided a script to make this super easy for you. Execute the command `jupyter notebook password`, and then enter a secure password. This is the password you will use to login to your Jupyter notebook from another computer.

2.3.3 Change your hostname [Optional]

Your Pi will come with a default hostname of `pi`, but you can change it. You should see a serial number engraved into the wooden casing of your Raspberry Pi, e.g. "Serial #001". You can change your hostname from the default to e.g. rasp001 using the serial number of your device.

1. Run `sudo raspi-config`
2. Select `1 System Options → S4 Hostname` , then select `<Ok>`
3. Enter your new hostname (e.g. `rasp001`, note that the hostname is case insensitive) and select `<Ok>` . Your hostname will change on the next reboot.
4. Select `<Finish>` and reboot.

Your new hostname and login password are now set. After rebooting, your terminal should look like `rasp99 login:` . Enter username `pi` and your new password. **Do not forget this password! If you do, your SD card will need to be reformatted!**

3. Configure Jupyter Notebook

Before doing this, please make sure you have formed a team and know your team name!

3.1 Downloadfiles from GitHub to Pi [Recommended]

The best way to download files for lab0 is to download from GitHub directly to your Pi. You will first need a GitHub account if you do not have one already. Navigate to `https://classroom.github.com/g/S4GqSTah` in your browser to accept the lab1 assignment on GitHub. This will create a private repository for you containing the starter code for this lab. Note the name of the repository so that you can clone it in the next step.

On your Pi, `cd` to `~/EE123` and run the command

```
git clone https://github.com/EE123-students/lab1-spring2021-YOUR_TEAM_NAME lab1-spring2021
```

to clone the repository into a directory called `lab1-spring2021`.

To push files from your Pi to GitHub, use the following commands:

1. Make sure you are in your local directory by running

   ```
   cd ~/EE123/lab1-spring2021/
   ```

2. Run `git add .` to stage any modified files.

3. Run `git commit -m "TYPE A USEFUL MESSAGE HERE"` to commit the changes locally.

4. Run `git push` to push committed changes to GitHub.

3.2 Transfer files from your laptop via SCP [Optional]

Download the lab1 folder to your laptop from the GitHub repository. These files will be copied to your Raspberry Pi using the following instructions.

For Mac OS X:
From a terminal, move the lab0 notebook file to the Pi by executing

```
scp -r [path_to_lab1_folder] pi@[pi_IP_address]:~/EE123/lab1/
```

Alternatively, from a terminal on your Pi, you can use the command `wget http://file.url` to download the file.

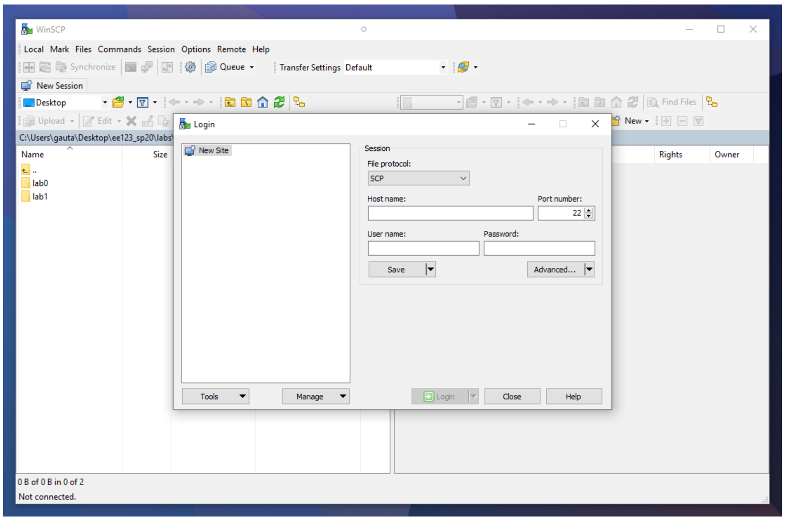
For Windows:
PuTTY comes with a command line program called PSCP which can be used to copy files (and folders with the `-r` flag). From the command prompt (search “CMD” to start), the syntax is

```
pscp [-r] source_file destination
```
.

For example, to move a folder from your PC to your Pi, type

```
pscp -r [PathToFile] pi@[pi_IP_address]:/home/pi/EE123
```

If you prefer a non-terminal interface, you can download WinSCP here. During installation, choose the "commander" view. After running the program, you should be prompted to enter login information, as in Figure 9 below.



Transfer Settings Default

C:\Users\gauta\Desktop\ee123_sp20\labs\

File protocol: SCP

Host name:

Port number: 22

User name:

Password:

0 B of 0 B in 0 of 2

Not connected.

Figure 9: WinSCP login screen

Set the file protocol to SCP, and enter your hostname, username `pi` and your password in the appropriate boxes. Once you have successfully logged in, your laptop files will appear on the left and your Raspberry Pi files will appear on the right. You can use this interface the same way you use the Windows file explorer.

3.3 Connect 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.

1. Make sure your Pi and laptop are on the same wireless network!

2. Type `hostname -I` in a terminal on the Pi to get the Pi's IP address.

3. In a browser window on your laptop, type `https://192.168.1.1:5555` (replace with your Pi's IP, but still include the `:5555` part!)

4. 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.

5. Enter the Jupyter password you set earlier into the prompt that looks like Figure 10:

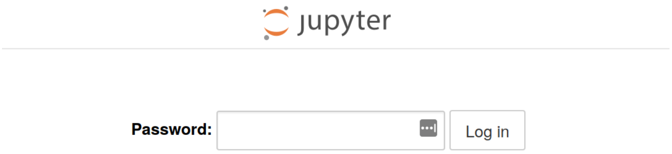


Figure 10: Jupyter Notebook login screen

That's it! You can now open lab1-dtft.ipynb and fill in the missing Python code. When you are finished with the lab, shut down the Pi by typing `sudo shutdown -h now`. Again, always shutdown the Pi using the command `sudo shutdown -h now` BEFORE disconnecting the power or you may corrupt your file system.