# Lab 0: Intro to Running Jupyter Notebook on a Raspberry Pi

Alan Dong, Gautam Gunjala, Josh Sanz,

based on previous work by

Li-Hao Yeh, Michael Chen, Nick Antipa, John Tamir, and Frank Ong

January 29, 2020

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 microSD card. We have provided microSD 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 and set up several methods of wireless connectivity: three types of Wi-fi network and optionally a Bluetooth PAN. 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!

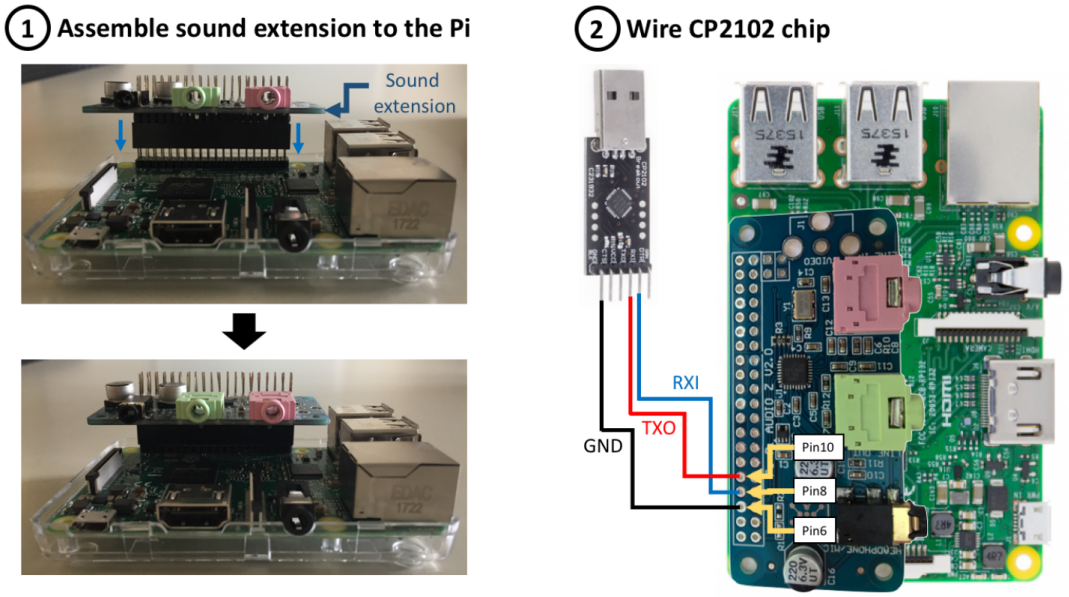


Figure 1: Raspberry Pi assembly with Fe-pi sound extension and CP2102 chip

## 2 USB-to-serial and sound extension connection

The first step of this lab is to assemble the sound extension to the Pi and establish communication between your laptop and your Pi via a serial connection over the CP2102 USB-to-UART bridge.

1. Start by downloading and installing the drivers for your appropriate OS here.
2. Next, attach the sound extension to the Pi and then use the provided jumper cables to wire the CP2102 to your Pi (see Figure 1).
   a) CP2102 RXI → Pi pin 8
   b) CP2102 TXO → Pi pin 10
   c) CP2102 GND → Pi pin 6

   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-A port. Do not power up the Pi yet.

## 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.

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.

## 3.1 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 commend, 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 connection to the Pi.

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

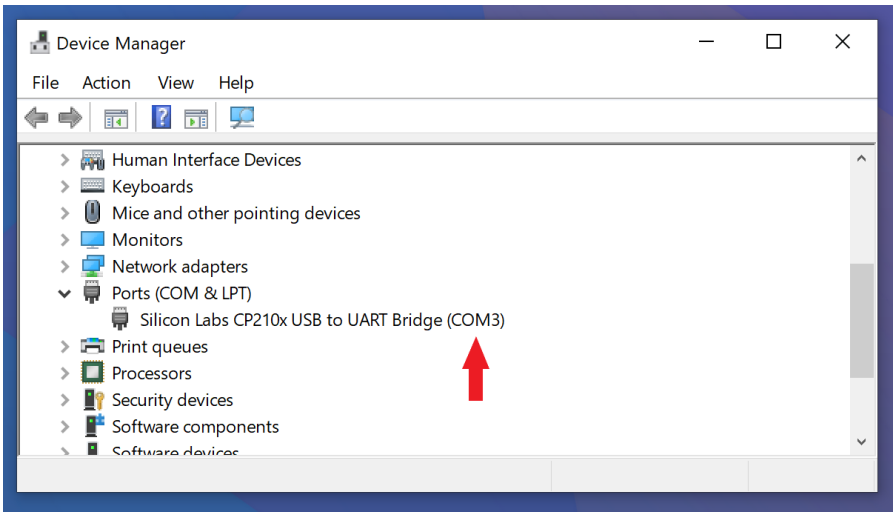


Figure 2: Device manager with recognized CP2102 USB-to-serial bridge

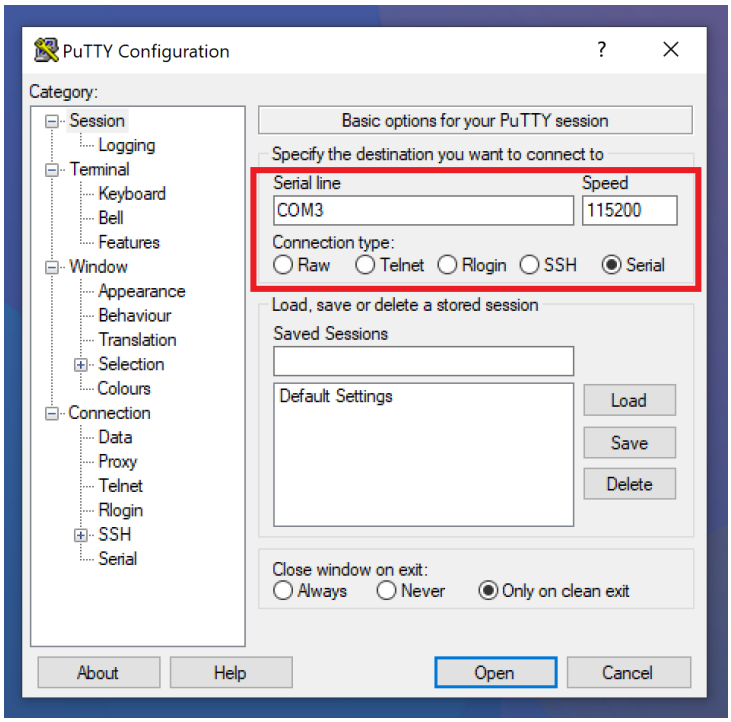


Figure 3: PuTTY setup window

### 3.3 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.

## 4 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 7, `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 terminal window returns to the login screen, 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.

# 5 Change your hostname and passwords

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 hostname of your Pi from the default.

## 5.1 Hostname

Your Pi will come with a default hostname of `raspberrypi`, but this can be confusing with so many identical hostnames around! You should see a serial number engraved into the wooden casing of your Raspberry Pi, e.g. "Serial #001". You will change your hostname from the default to e.g. `rasp001` using the serial number of your device.

1. Run `sudo raspi-config`
2. Select Option 2, `Network Options` → `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. Don't select `<Finish>`. Instead, go to the next step.

## 5.2 Login password

We then need to change your login password so others can't login to your Pi (especially when connected to a network, which you will do later).

1. From the `raspi-config` menu, select Option 1, `Change User Password`
2. Set a secure password that only you know.
3. 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!**

## 5.3 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.

# 6 Setting up networking

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 various methods for wirelessly connecting to your Pi from another computer.

## 6.1 Connecting to Wi-Fi

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, login and run `sudo raspi-config`
3. Select `Network Options` → `Wi-Fi`. 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 6.2
4. Test that you are online by pinging a website (e.g. `ping www.google.com`)

## 6.2 Connecting 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 4. 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.

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

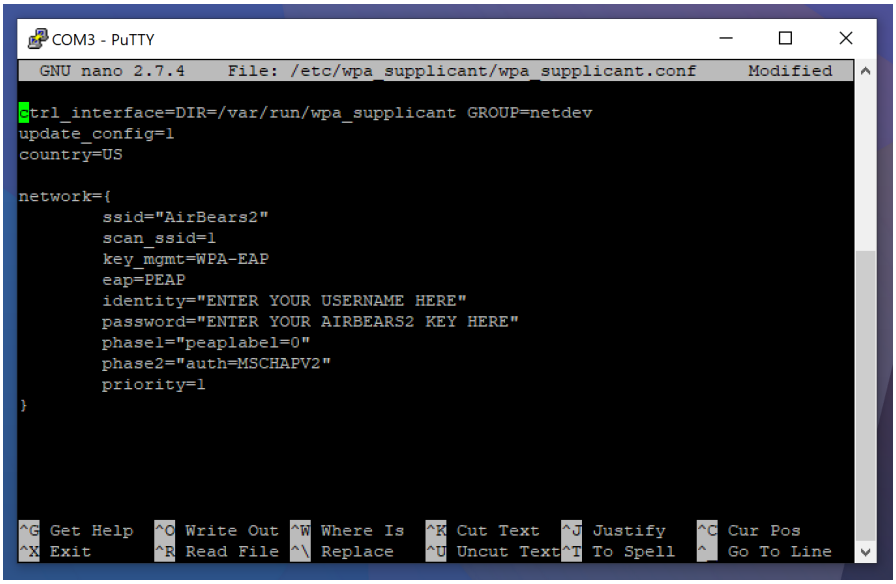


Figure 4: Plain text file containing network information

## 6.3 Ad hoc network

This method will allow you to wirelessly connect a laptop to your Pi without *any* other networking infrastructure[3]. This is handy if you want to do your DSP labs on your Pi away from your home network. Your outrageously talented teaching staff has 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 any issues, proceed to Section 6.4.

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.

## 6.4 Access Point (a.k.a. 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-secured Wi-Fi 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
   ```

[3] According to Wikipedia, a network that is ad hoc "does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks."

2. Change the SSID to something unique by editing the line
   `ssid=EE123ChangeThisName`
3. Set the password (to something secure!) by editing
   `wpa_passphrase=EE123Rocks!`
4. Save the file (`Ctrl-X, Y, Enter`)
5. Boot into managed mode using `sudo ~/managed`
6. After reboot, log in.
7. You should now see a WPA-2 secured Wi-Fi network named whatever you named it. Connect using your super secret password.

# 7 Bluetooth Personal Area Network (PAN) [Optional]

Since the Pi contains a Bluetooth module, it is possible to access your Pi from your laptop over a Bluetooth connection. This will replace the USB-to-serial bridge and, if desired, the ad hoc and managed Wi-Fi modes. This is especially convenient if you run into Wi-Fi connectivity issues. If you wish to set this up, the first thing you should do is to change the hostname of your Pi (you should have done this already in Section 5.1).

After you have successfully connected to the Pi's Bluetooth PAN, you will be able to SSH into the Pi at `10.0.0.1`.

## 7.1 Connecting to the Pi

**Instructions for Mac**:

1. Open `System Preferences` → `Bluetooth`.
2. Reboot the Pi and wait for its hostname to appear.
3. Click `Connect` to pair with the Pi.
4. `control+click` on the device and select `Connect to network`.
5. Check that you have connected successfully by running `ping 10.0.0.1` in the terminal.

**Instructions for Windows**:

1. 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 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.

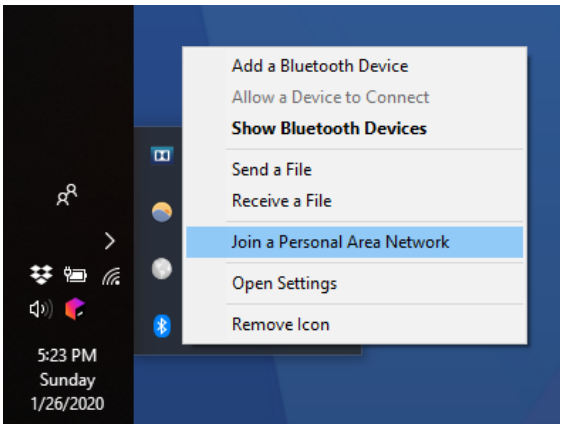


Figure 5: Navigating to Bluetooth devices from Windows taskbar (step 2b)

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.

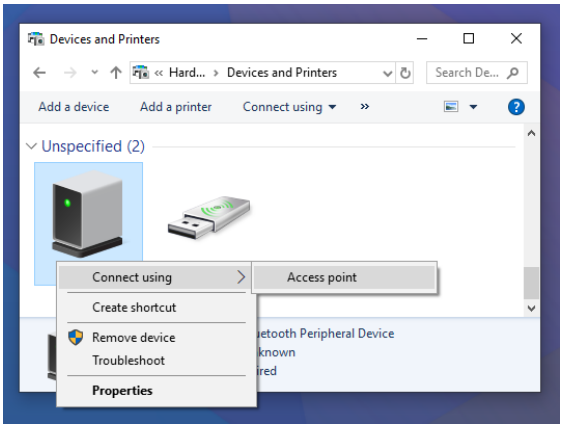


Figure 6: Establishing a Bluetooth connection to the Raspberry Pi (step 2)

3. Open PuTTY, click `Session` in the menu on the left side, and enter "10.0.0.1" in the box under `Host Name (or IP address)`. The connection type should be `SSH` by default. Once your window looks like figure 7, click `Open`.

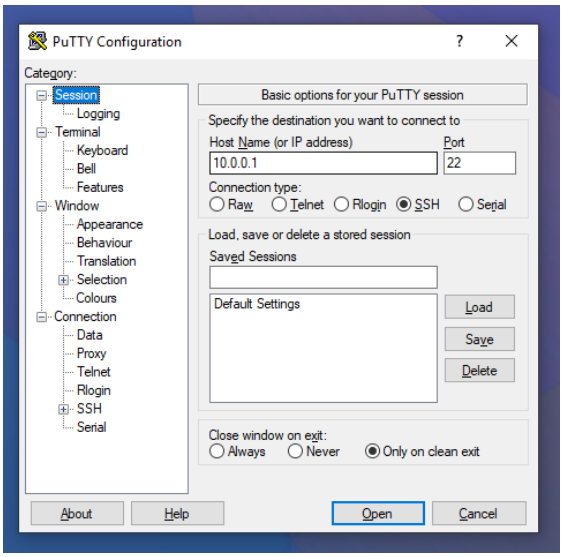


Figure 7: Access the Raspberry Pi via SSH (step 3)

4. A terminal window should open. When prompted, enter username `pi` and the password you set earlier.

# 8 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.

## 8.1 Ad hoc/managed mode

With either ad hoc or managed methods, you can connect to your Pi using a laptop without needing a separate 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 or the Bluetooth PAN if you have issues with ad hoc.

To use ad hoc or managed mode:

1. Make sure your Pi is booted into ad hoc or managed mode (see Sections 6.3 and 6.4). 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:

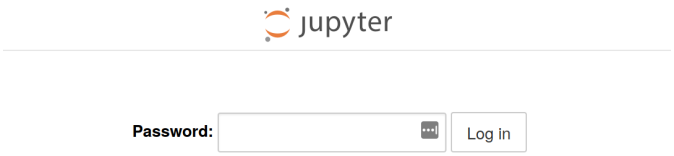


Figure 8: Jupyter Notebook login screen

## 8.2 Wi-Fi Mode

With your Pi and your laptop on the same wireless network (this is important), go to the terminal on the Pi. Type `hostname -I` to get the IP address. Follow the instructions in Section 8.1 to connect to the Jupyter notebook, 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.

## 8.3 Bluetooth PAN

The Pi always uses the IP `10.0.0.1` on the Bluetooth PAN, so you can access your Jupyter notebooks at `https://10.0.0.1:5555` while you are connected. Be warned that this connection is *much slower than Wi-Fi*, so it may be painful when doing a lot of plotting or other data-intensive interactions.

Regardless of which method you used to connect, 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. At this point, you can remove the CP2102 and use SSH to connect to the Pi. Remember which connection method you are using because the IP address you use to SSH will be different!

# 9 Transferring files to and from your device

## 9.1 Downloading files 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/a/fQCtubFy in your browser to accept the lab0 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/lab0-YOUR_GITHUB_USERNAME lab0
```

to clone the repository into a directory called `lab0`.

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/lab0/
   ```

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.

## 9.2 Transferring files from your laptop via SCP

Download the lab0 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_lab0_folder] pi@[pi_IP_address]:~/EE123/lab0/
```

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.

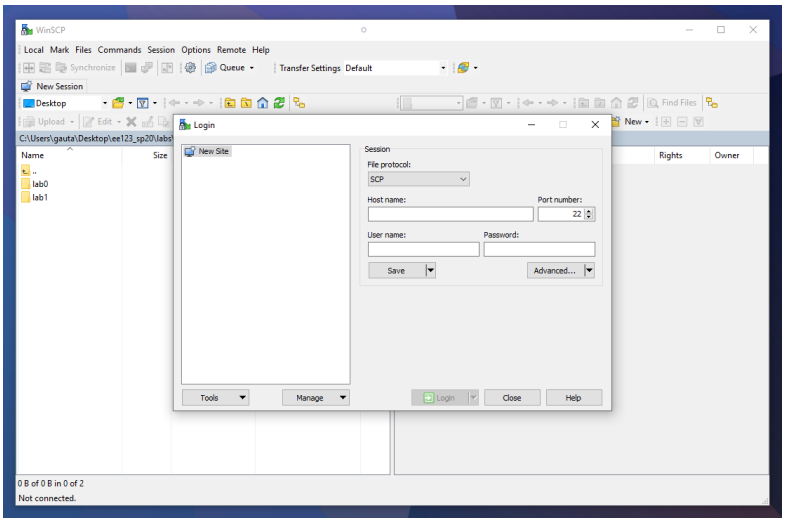


Figure 9: WinSCP login screen

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.

That's it! You can now open lab0.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.

## 10 Useful commands

1. `screen /dev/cu.SLAB_USBtoUART 115200`: connect to the Pi through CP2102 from your laptop
2. `screen -r`: resume previously created screen session
3. `sudo raspi-config`: open Pi configuration window
4. `jupyter notebook password`: change password for Jupyter notebook server
5. `sudo nano [path_of_file]`: edit the file with nano
6. `sudo ~/wifi`: switch the Pi to Wi-Fi mode
7. `sudo ~/adhoc YourSSID`: switch the Pi to ad hoc mode with SSID of EE123YourSSID
8. `sudo ~/managed`: switch the Pi to Managed mode
9. `hostname -I`: print IP address of Pi (for Wi-Fi mode)
10. `scp [path_of_file] pi@IP_ADDRESS_OF_PI:~/EE123/lab0/`: copy file to Pi's `~/EE123/lab0/` folder
11. `sudo reboot` & `sudo shutdown -r now`: reboot the Pi
12. `sudo shutdown -h now`: shutdown the Pi