Introduction to Raspberry Pi in Environmental Science

Anna Burns & Marc Los Huertos

June 22, 2020

1 Introduction

1.1 What is Rasberry Pi?

The Raspberry Pi is an tiny computer, that includes a tiny processor, a bit of memory, a slot for an SD card, and some input/output jacks, e.g. HDMI, USB, headphone, camera, and pin header for various sensors.

1.2 Why use Raspberry Pi?

The Pi has a lot of functionality and flexibility for develop monitoring of environmental parameters.

1.3 Uses in Environmental Science

1.3.1 Weather and Climate Change

1.3.2 Air Pollution Monitoring

1.3.3 Soil Water Monitoring and Irrigation Control

The automatation of irrigation is ubiquitous, however, in many cases systems don't have feedbacks that control the system. The first step relies on sensors to monitor soil moisture and temperature and then use plant growth models to control an irrigation system.

Here are some resources that describe how these can be done:

- Soil Moisture Sensors and Loggers

1.3.4 Conservation

Conservation biologists use a wide range of instruments to track and monitor wildlife (camera traps, active (radio) and passive (RIF) transmitters). In addition the use of cameras are used to evaluate plant health and diversity (spectral analysis).

1.4 Resources

1.5 Raspberry Pi

The Raspberry was created to help non-technical youth to learn computer and robotics. The Raspberry Pi was an unexpected success and now the Pi is one of the most important minaturized computers for a wide range of projects.

The RaspberryPi.org has tutorials, software updates, and example projects.

1.5.1 Video Tutorials

One good way to start is to watch the following video series that introduced the Raspberry Pi.

1.5.2 Unpacking the Raspberry Pi

When you recieve your Pi, plan on spending about 1-2 hours setting it up which include unpacking and putting the Pi together:

1. Unpack Kit Contents
2. Put Pi in case and add heat sinks

2 Communicating with Raspberry Pi

2.1 Option 1: Direct Access with the Pi Using a Monitor, Keyboard, and Mouse

If you have a spare monitor, keyboard, and mouse, follow the following steps to set up the Raspberry Pi. If you don't have these, then you'll need a ethernet cable and follow the steps in Option 4: Remote Access to Pi using SSH.

1. Connect to keyboard, mouse and monitor. Make sure the HDMI plug is in the correct mini-HDMI socket and the monitor is configured to get a signal from the port being used.
2. Insert pre-loaded SD card
3. Plug-in Pi, you'll see a rainbow screen for a minute and then a installation menu.
4. Install the Rasbian operating system only. This will take 10 minutes. Read the little windows so you know some of the resources associated with the operating system. The installations stalls at the end, where it says 100%. Be patient, it will finish on it's own and reboot.
5. Once the Rasbian OS starts, you'll see four raspberries at the top and then you'd get some prompts to set up the OS.

6. I suggest you keep the password as default for now, select the langauge, keyboard type, and time zone.

7. Next you'll need to get connected to the internet. Select your modem and enter password to connect.

8. Then you'll get a prompt to check for updates. Yes, there will be updates. This will take another 20 minutes. About 1/2 way the screen goes completely blank. This is because, by default, the Pi goes into screen saver mode without mouse or keyboard inputs. Move the mouse or type something and the screen should wake up.

9. Be sure to shut down your Raspberry Pi before unplugging it; you can use the keyboard and "Cntl-Alt-Delete" sequence to get a pop-up window to shut the Pi down or open the terminal window and type:

```
$ sudo shutdown -h now
```

10. You have successfully communicated with the Pi!

11. Unplug the Pi.

    Note: To restart the Pi, simply plug it in, unless you have a switch.

2.2 Option 2: Remote Access with Pi

It doesn't take too long (10 minutes once you get the hang of it) to be able to work remotely on your microprocessor via a tablet or laptop. More information by reading https://pythonprogramming.net/remote-access-raspberry-pi-tutorials/]

1. Begin by making sure all of your packages are up-to-date (even if you updated while setting up your OS, you will still need to do this). Do this with the code "$sudo apt-get update" and then "$sudo apt-get upgrade". This will take several minutes.

2. Enable the SSH server by typing "$if config" into the console. This will lead you to a screen with several options; select option five, "interfacing options," and then select option two for "SSH." Enable the server, and then exit this screen.

3. Note that Raspberry Pi updated in November 2016 so that SSH is disabled automatically. You will need to turn it on in order to progress.

4. Type "$ifconfig" into your console to find your Raspberry Pi's inet address under the "wlan" subcategory. This should be a string of numbers that looks something like "192.168.XXX.X.XXX." You will need this to access your Raspberry Pi remotely.

Before you proceed, look at the descriptions for options 2A and 2B below to see which is the right choice for your remote access needs.

2.2.1 Option 2A: Remote Access to the Raspberry Pi GUI

If you don't have much computer programming experience, this option may be better for you. It allows you to access the actual Pi terminal (like the one you see when you connect an HTML cable) on another computer. It is a more visually-friendly option, and allows you to directly access different apps through the Pi itself.

1. Install the package necessary to remotely access your screen: "$sudo apt-get install xrdp".
2. Connecting to Mac or Linux OS: open the terminal and add "ssh pi(at)(inet)".
3. Connecting to Windows OS: this is a bit more involved; you will need to download an SSH client.
4. The Microsoft "Remote Desktop Connection" app is available for free, and is pretty intuitive to use. Download this, and follow the instructions to set it up.
5. Once it is installed, all you will need to do is enter the address you found above into the text box (it should look something like 192.168.XXX.X.XXX).
6. A page will open for you to add your username and password (which at this point should still be "pi" and "raspberry"). Then your console will open.
7. If it won't connect: make sure your microprocessor and the computer you are attempting to accesss it from are on the same wifi;
8. make sure your microprocessor is connected and hasn't gone to sleep;
9. make sure you've properly enabled all of the packages (I had to try to run xrdp twice to get it to work the first time).

2.2.2 Option 2B: Remote Access to Raspberry Pi via SSH

This option allows you to access your Pi microprocessor through an external shell, "piggybacking" the Pi onto an already existing computer. It allows you to operate in raw code through an external frame, but does not have the visual aspects that the GUI does. If you are more comfortable with programming, or feel familiar enough with navigating the Pi, this is the right option for you.

1. For Windows Only: (if you have Mac OS or Linux, skip this step!) Make sure you have OpenSSH Client installed.
   (a) Open Windows Settings. If you are using Windows 10, click the Windows menu, and select the gear icon for Settings.
   (b) Go to settings, Apps, Apps and Features, Optional Features.

(c) Click on Add a Feature.

(d) Scroll down and add Open SSH Server.

(e) Close the Settings Windows.

2. Ethernet IP Address: 192.168.137.1

3. For Windows: (if you have Mac OS or Linux, skip this step!) First, make sure you have OpenSSH Client installed. Go to settings, Apps, Apps and Features, Optional Features, Add a Feature, and then add the client.

4. Open your Powershell (which acts sort of like an app, you can access it in the search bar) as an administrator, and then type "Ssh [username]@[identifier]" (the address that you found above, it should look something like 192.168.X.XXX). Username will be "pi" unless you have changed it during setup.

5. Open your Powershell (which acts sort of like an app, you can access it in the search bar) as an administrator, and then type "Ssh [username]@[identifier] (the IP address that you found above).

6. Once you press enter, it will ask if you trust it (say yes, you will only need to do this once).

7. Next it will ask you for your password. Note: As you type, no characters will appear on the terminal (for privacy). This is ok, once you have typed your password just press enter. If you didn't change your password during setup, it will be "raspberry".

8. The terminal should now look like the Raspberry Pi terminal that you get via HDMI, with "pi@raspberry pi: $" as the beginning of your lines. Hooray!

9. For further guidance, see the troubleshooting section.

2.3 Troubleshooting

2.3.1 Remote Access

1. Your microprocessor needs to be connected to a powersource in order to remotely access it.

2. Both the GUI and main terminal options can be rather glitchy; if the program quits, wait a moment and then log back in. It may not work on the first try, but do it again and it should connect.

3. Make sure that your computer is connected to Wifi, and that it is connected to the same Wifi network your microprocessor is on.

2.3.2 Screensaver

The screen goes black after 10 minutes of inaction. This is a bit annoying since we are often doing things that take some time to sort out, i.e. more than 10 minutes.

To turn this off you can a terminal window that allows you to type commands following the $ prompt.

Type the following command to turn off the screen saver using the following steps:

1. Type:

```
sudo apt-get install xscreensaver
```

2. If this doesn't work, make sure your sudo is up to date with

```
sudo apt-get update
```

3. Once you've done this, go to the Raspberry icon in the upper-left corner and select "preferences" from the drop down menu. Under "preferences" there is an option for "screensaver".

4. Once in screensaver, under the "Mode" dropdown you can select "disable screensaver".

5. Now your screen should stay active even if you aren't actively using it.

6. More information/different methods are available at: Pi Forum: Disable Screensaver is Raspbian

2.3.3 Recovering Commits from Github

If there is a conflict between commits and some information that was previously added to a file on R is lost, follow these steps to recover the information within previous commits.

1. Go to the project's Github page over the internet and open the page for the document you wish to recover code from.

2. On the bar before the file itself begins, select "history" on the right side of the screen.

3. Within the history tab, there should be a list of previous commits; select a commit that will have the code you are hoping to recover within it (i.e. if you were working on the code the previous Friday, select a commit from that day).

4. On the appropriate commit, select the "¡¿" option on the right of the screen to open up a historic version of the repository from that day. Note that this will direct you to the full repository from the time of the commit you select, and not just the document you are hoping to recover.

5. Go back to the document you need to recover code from, and open it. It will have the historic code that was there at the time of the commit you selected; if you chose correctly, it should have the code you were looking to recover.

6. Copy the deleted lines of code out of the historic file, and into the document on R.

7. Save the R document, commit, and push the information.

8. The Github will remain on the historic commit until you leave the project; exit the entire project (for example, by going to the home screen on Github) and then re-enter the project to enter the up-to-date portal.

2.4 Projects General-purpose input-output Examples

2.4.1 Weather and Climate

2.4.2 Air Quality

2.4.3 Sensors

Particulate Matter
NOx
O3
CO
Temperature and Humidity

2.4.4 Software

- Python Script

2.4.5 Soil Moisture

Tutorials in using a Pi to monistor soil mosture:

- Simple and Inexpensive Method
- Raspberry 3 and Capacitance Sensors – More Accurate, limited corrosion
- moisture-sensor-dfrobot?

2.4.6 Wildlife

Poacher Cam v7, Chris Kline, panthera.org