Introduction to Raspberry Pi in Environmental Science

Anna Burns & Marc Los Huertos

July 17, 2020

1 Introduction

1.1 What is Rasberry Pi?

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

1.2 Why use Raspberry Pi?

The Pi has a lot of functionality and flexibility to 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 uses 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).

2 Resources

2.1 Resources

2.2 Raspberry Pi Foundation

The Raspberry was created to help non-technical youth learn computing 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.

2.2.1 Video Tutorials

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

2.3 Unpacking the Raspberry Pi

We have three different models of Raspberry Pi, 3B version 2, 4B version X, and Zero W version X. The set up is slighly different for each.

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 if included or not previously installed

3 Setting Up Raspberry Pi

In order to set up your Raspberry Pi, you will need one of two things: access to a USB-connected mouse and keyboard as well as a monitor that allows you to connect via HDMI (TVs or desktop computers both have this capability); or a computer that can read an SD card. The first of these options is simplest; if you do not have access, use option two for the SD-card option. If you do not have access to either, contact your professor so that they can get a monitor, keyboard, and mouse to you.

3.1 Section 1: Direct Access with the Pi Using a Monitor, Keyboard, and Mouse

If you have a spare monitor, keyboard, and mouse, use the following steps to set up the Raspberry Pi. If you don't have these, but do have a slot on your computer for an SD card, then follow the instructions in Section 2.

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. If the SD care does not have the operating system (OS), then see the SOP for "Imaging the Pi 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. 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.

6. Insert pre-loaded SD card.

7. Plug-in Pi, you'll see a rainbow screen for a minute and then a installation menu.

8. 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 stall at the end, where it says 100%. Be patient, it will finish on it's own and reboot.

9. 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.

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

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

12. Then you'll get a prompt to check for updates. Yes, there will be updates. This will take another 20 minutes. About halfway 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.

13. Be sure to shut down your Raspberry Pi before unplugging it; you can use the keyboard and "Ctrl-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
```

14. You have successfully communicated with the Pi!

15. Unplug the Pi.

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

    You can now access your Pi via a monitor; however, if you would like a more portable option, read section 3 on "Remote Access on a Configured Pi" to be able to use your monitor to install the correct settings and software so that you can access your Pi from anywhere (and thus be freed from your monitor!).

3.2 Section 2: Remote Access Startup with the Pi

If you do not have access to a keyboard, mouse, and monitor, but DO have access to a laptop with an SD card slot, this option is for you. Once you go through these steps, you will be ready to begin working on your Pi remotely (and thus will not need to follow the instructions in the following section, "Remote Access on a Configured Pi").

Our Pi comes with a pre-downloaded SD card with the various operating systems already downloaded; you need to select which one you would like (Raspain), and you then need to add files to the boot on the SD card so that your Pi can connect to Wifi and the SSH is enabled. Finally, you need to find your Pi's specific IP address so that you can connect remotely, and then you can access your Pi via SSH!

NOTE: This section has not been tested, but should be shortly. Edited instructions are pending

Source:

1. Insert the Pi's SD card into your computer and access it through your computer's file navigator. It should be accessible under "BOOT" and "RECOVERY".

2. Within the RECOVERY file system, open the "OS" folder. Within OS, there should be multiple system options in different folders; delete all of them except for "Raspian". This configures your operating system to Raspian.

3. Go back to the root (or the main level) of the BOOT folder. This is where we will create the files that will allow you to operate via SSH and connect to wifi.

4. Using notepad, create a document named "wpa_supplicant.conf".

5. Insert the following code into the notepad document:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

country=Insert 2 letter ISO 3166-1 country code here

network={
 ssid="Name of your wireless LAN"
 psk="Password for your wireless LAN"
}
```

   Being sure to insert three things: your 2 letter ISO country code (if you are in the United States, this is US); your Wifi's name; and your Wifi's password. Note that there are meant to be quotation marks around your ssid and psk.

6. Within the notepad document, go into file/save as. Save the file within the boot folder for the SD card. Note that if you are on Windows, make sure that the file type is "all files" and not "txt".

7. Next, create a notepad document named "ssh".

8. Insert the following code into the document:

```
cd /Volumes/boot
touch ssh
```

9. Once again, within the notepad document go into file/save as. Save the file within the boot folder on the SD card. Note that if you are on Windows, make sure that the file type is "all files" and not "txt".

10. Double check that both of the files you created are within the boot folder, and then left-click on the folder and select "eject".

11. Remove the SD card from your computer, and reinsert it into the Pi.

12. Connect your Pi to power.

13. Open up your computer's shell (on Windows, this can be accessed under apps by typing "Windows Powershell").

14. Within the command line on the shell, find your Pi's IP address by typing "ping raspberrypi.local". As long as your computer and the Pi are on the same wifi network, this will tell you your Pi's IP address.

15. Your Pi is now configured, and now we can access it via SSH with the following instructions in your computer's shell.

16. Type

```
ssh username@IPaddress
```

where the username is pi and the IP address you just found by pinging (it should look something like this: 192.168.X.X). If you changed your username during setup, then you'll type something other than "pi".

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

18. 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".

19. The terminal should now look like the Raspberry Pi terminal that you get via HDMI, with "pi@raspberry pi: $" command prompt. Hooray!

20. Finally, you need to make sure your Pi is up to date. Do this with the following lines within the Pi shell:

```
sudo apt-get upgrade
sudo apt-get update
```

    Since this is your first time using your Pi, this process could take several minutes.

21. Now that your Pi is updated, you can follow the instructions undersection 3A of "Remote Access on a Configured Pi" to access your Pi's GUI (user interface). Instead of typing the commands into the Pi terminal in your monitor , you will do so in your computer's shell with the Pi shell piggy-backed onto it. Note that this guide takes you through a remote setup of option 2B of "Remote Access on a Configured Pi", and you will be able to follow the rest of the setup protocol without switching to a GUI. Read the following section to see which option is right for you.

3.3 Option 3: Remote Access on a Configured Pi

This section is for if you have already set up your Pi via a monitor, and would like to be able to access it via a laptop. If you have not yet configured your Raspberry Pi with the correct operating systems and settings, please refer to the previous section, "Remote Access Startup with the 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.

This requires the use of a keyboard, mouse, and monitor, then you'll be free!

More information by reading Remote Access with Raspberry Pi Tutorial or Remote Access Documentation.

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
```

    If there is an update, you will be asked to confirm the upgrade because it might take additional disk space. I don't see anyway around this, so type in Y. The update will will take several minutes.

    It might ask you to reboot, if so go for it and restart the terminal window. Either way, continue below.

2. Enable the SSH server by typing

```
sudo raspi-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. You will need to know your Raspberry Pi's IP address to connect to it. To find this, type

```
hostname -I
```

from your Raspberry Pi terminal to find your Pi's IP address. This should be a string of numbers that looks something like "192.168.XXX.XXX". You will need this to access your Raspberry Pi remotely. Find some paper and write down the number.

Before you proceed, look at the descriptions for options 3A and 3B below to see which is the right choice for your remote access needs.

3.3.1 Option 3A: 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 type

```
ssh pi@IPaddress
```

3. Connecting to Windows OS: this is a bit more involved; you might need to download an SSH client.

4. The Microsoft "Remote Desktop Connection" app is available for free, and is pretty intuitive to use. Usually, this is already installed on the computer and you can find it by searching for the name using the tool bar search icon. If not, you may need to download the application, and follow the instructions to set it up.

5. Start the probem and it will prompt you to enter the Pi's IP address you found above into the text box (it should look something like 192.168.X.X).

6. A page will open for you to add your username and password (which at this point should still be Pi's username and password, that is, "pi" and "raspberry"). Then your console will open.

7. If it won't conect: make sure your microprocessor and the computer you are attempting to accesss it from are on the same wifi, you can check by using the ping command in the command window to confirm communications.

8. Make sure your microprocessor is connected and hasn't gone to sleep;

9. Make sure you've properly enabled all of the packages. Note: You may need to run the xrdp twice to get it to work for the first time).

3.3.2 Option 3B: 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 appeal 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. Open the Command Prompt (which is command line window in windows)

3. I recommend that you "" the Pi before going to the next step. Ping is a short command use to check of the communications are working. As it turns out if I am using Pomona's VPN, I could not get it to work. So, I disconnected from the VPN and then typed

   ```
   ping 192.168.0.4
   ```

   If you get a series of packets and not a timeout error you are talking to your Pi! If not, stop – don't get frustrated and contact your contact person for help!

4. Type

```
ssh username@IPaddress
```

   where the username is pi and the IP address is the number you wrote down in step 4 in section 2.2 (itshould look something like this: 192.168.X.X). If you changed your username during setup, then you'll type something other than "pi".

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

6. 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".

7. The terminal should now look like the Raspberry Pi terminal that you get via HDMI, with "pi@raspberry pi: $" command prompt. Hooray!

8. For further guidance, see the troubleshooting section.

4 Troubleshooting

4.1 Unable establish ssh connection

if you can ping, but the "connection refused", then the pi might not have the ssh enabled. See section XXX to make sure it's enabled.

If it is enabled, double check IP address to make sure there is no typo.

4.2 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.

4.3 Headless SSH Connection

If you need to connect your Pi microprocessor to a new Wifi network but do not have access to a monitor, follow this process.

1. Make sure that your Pi is NOT connected to a power source, and remove the MicroSD card.

2. Insert the MicroSD card into a computer; the process for this is different for each computer, and you can Google how to insert a MicroSD card for the model computer you are using. It will look like a small slot (about 1 cm wide), and when you insert the card it should click in.

3. Using notepad, create a document named "wpa_supplicant.conf".

4. Insert the following code into the notepad document:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev

update_config=1

country=Insert 2 letter ISO 3166-1 country code here

network={
 ssid="Name of your wireless LAN"
 psk="Password for your wireless LAN"
}
```

   Being sure to insert three things: your 2 letter ISO country code (if you are in the United States, this is US); your Wifi's name; and your Wifi's password. Note that there are meant to be quotation marks around your ssid and psk.

5. Within the notepad document, go into file/save as. Save the file within the boot folder for the SD card. Note that if you are on Windows, make sure that the file type is "all files" and not "txt".

6. Within your file directory, left-click on the "boot" folder and select eject. You can now take out the SD card by pushing the card further in so that it un-clicks.

7. Reinsert the MicroSD card into the Pi, and continue the remote access connection in one of the methods discussed previously.

8. Note that even if you had already enabled SSH and found your pi's IP address on a previous network, connecting to a new network means that you need to re-establish SSH connections, AND that your Pi's IP address will (strangely) be different! The remote connection will NOT work with the IP address you had used on your previous network connection.

9. To find your Pi's new IP address, connect the Pi to power and then enter the shell of a computer that is on the same wifi network as the Pi. In the command line, ping your Pi by typing "ping raspberrypi.local". This should give you the Pi's IP address, which should look something like

"192.168.X.XXX". You can use this to connect to your Pi via SSH with the instructions in "Remote Access to a Configured Pi".

5 Copying Files (e.g. python) on the Pi

5.1 USB

5.2 SD card

5.3 via Internet

5.3.1 curl

curl is a command line program that can be used to get files from websites and install them directly onto your pi.

```
curl -sSL URL | bash
```

When you use this command, you will get a warning, that downloading filed from the internet is inherantly unsafe, unless you trust the source with your virtual life.

with the following arguments:

- -s silent mode
- -S show error (even when in silent mode)
- -L location, follow redirects (not sure what this means!

5.3.2 github

6 Maintaining the Pi

6.1 Monitor the Pi's Temperature

Right click on tool bar and "Add/Remove Panel Items". Then make sure the Panel Applets tab is selected, then click on the "Add" button. Add the "CPU Temerature Monitor" and close.

Follow the same steps to add the "CPU usage monitor".

6.2 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 awake even if you aren't actively using it.

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

6.2.1 MOVE THIS to GIT SOP? 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.

7 Implementing Pi Projects

7.1 Using Sensors with Pi

Now the Pi is set up we can move on to developing environmental monitoring projects.

7.2 Using a Camera with Pi