Lab 1: Setting Up the Environment

Setting up your Raspberry Pi for IoT development

In this lab, you will gain basic familiarity with the Raspberry Pi 4 Model B, choose and install an IDE that you will use for subsequent labs, and learn basic Raspbian/Linux environment commands. Specifically, you will write a program to download a weather report and save it into a text file when the Pi boots up.

Before you begin, please read the Lab Write-Up section, as you are encouraged to document your process as you carry out this lab assignment. This assignment is worth **15%** of your final grade.

Due: Friday, Oct 13, 2023

Lab Procedure

Part 1: Securing and Getting to Know Your Pi

Helpful Resources

- CanaKit Raspberry Pi 4 Quick-Start Guide booklet included in your kit
- How to install a new OS on Raspberry Pi
- How to change the password on Raspberry Pi
- How to set up Wi-Fi on Raspberry Pi
- Configure your Pi and access it through SSH
- How to configure Raspberry Pi to operate in ad-hoc mode

Procedure

Read the accompanying CanaKit Quick-Start Guide, included in your kit, to learn how to connect your Pi to peripherals and power it on. Once the Pi is turned on, do the following.

1. Follow the on-screen instructions to enter your location details, including selecting a country, language, time zone, and keyboard settings.
2. Change the root password to something unique that you won't forget. Do this before you connect to WiFi.
3. Connect your Pi to the UCSC WiFi network.
4. Update the software as prompted. This may take a few minutes.
5. Using the Secure Shell network protocol, you can also run your Pi in headless mode (without a monitor and mouse) by connecting to it through another computer. Practice accessing the Pi from your computer using the SSH connection.

6. By default, your Pi connects to a Wifi router; however, you may wish to enable it to operate in ad-hoc (or infrastructure-less) mode. This mode allows your Pi to create its own network to communicate with other Pis without using an external router. To practice this, configure your Pi to operate in ad-hoc mode. Before you modify the original file for the wifi interface, make sure you make a copy of it and name it **wifi-interface**. Name the new interface file you create as **adhoc-interface**.
7. Switch your Pi back from ad-hoc mode to infrastructure mode.

Part 2: Setting Up the Pi for Development

Helpful Resources

- IDEs for the Raspberry Pi
- OS and Python updates

Procedure

1. Check for available Raspberry Pi OS updates using the **sudo apt-get update** command.
2. Install the latest Python 3 version.
3. Research a few Integrated Development Environments (IDEs) that are available for download on the Raspberry Pi 4. Pick one IDE and install it onto your Pi, then test it by printing out "Hello World" to the output window.

Part 3: Creating a Weather Report Program

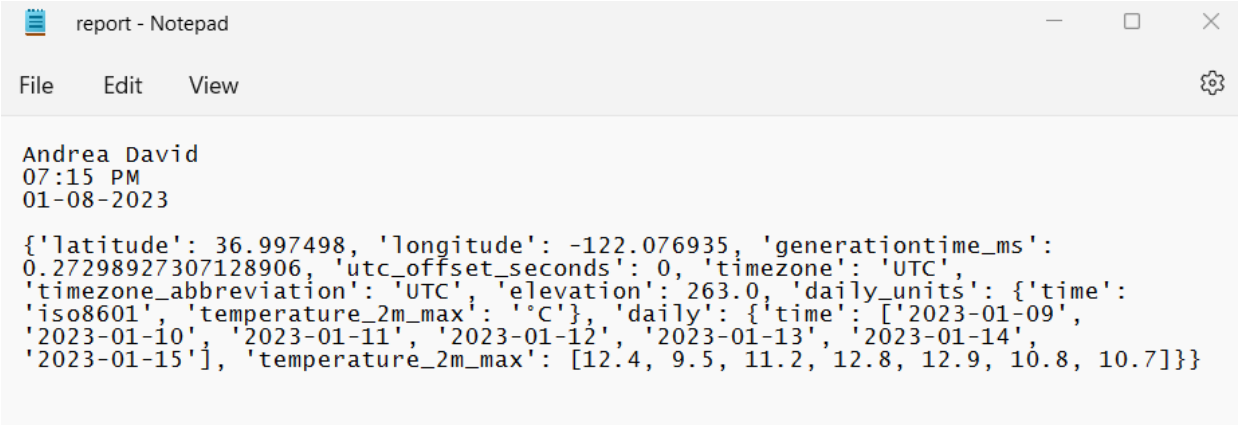
Helpful Resources

- OpenMeteoPy

Procedure

1. Create a Python script called **generate_report.py** that collects and displays the following information to an output file named **report.txt**.
    a. Using the Python script, display your full name at the top left-hand side of the output file and display the current date and time in the line below it.
    b. Your Python scripts should also retrieve information using the Open-Meteo service and output it in the text file below your name and date.
    c. In your script, you will need to enter the coordinates of the location from which you want meteorological data. To match the coordinates of UCSC, enter **37** and **-122.06** for latitude and longitude, respectively.
    d. You only need to output the daily maximum temperature at 2m at UCSC.

Your text file, **report.txt**, should look like this (with text wrap):



Part 4: Creating a Bootup Script to Run the Program at Startup

Helpful Resources

- Run a Program On Your Raspberry Pi At Startup

Procedure

1. Once you have completed your weather report program, include your Python script ***generate_reporty.py*** in the boot sequence to run automatically upon startup.

Lab Write-Up

Write-ups for your labs should provide comprehensive documentation of your work. They should be written as a tutorial or technical blog post with the goal of serving as a guide or document for someone unfamiliar with the lab assignment. Include screenshots, diagrams, or pictures where necessary to achieve this goal. You will be evaluated on the quality of your write-up and how well you meet the writing objectives described below.

Writing Objectives

As you write your paper, be sure to consider the following outline:

1. (30 points) Describe your process of initially setting up the Raspberry Pi.
    a. Provide a short description of initially setting up the Pi and connecting it to WiFi. Besides using the prompt at the initial bootup, what are other ways to change your password and set up the WiFi network?

   b. What was the process to remotely log into your Pi? Do you see yourself doing this often? Why or why not?
   c. What does the network information tell you generally? What do the interfaces represent? What specific information does each interface give you, and how is it relevant to how your device communicates?
      i. Follow-up: Compare the ad-hoc and wifi interface files. What similarities and differences do you notice? Why do you think this is?
2. (20 points) Describe your process of setting up the Pi for development.
   a. Describe your process of setting up the Pi for development.
   b. What IDE did you install, and why did you choose it over other options? How did you install them?
3. (20 points) Describe your process of creating your program, detailing each component.
   a. Provide a short description of your program, explaining each component.
   b. In your program, your code or the API you use should send a request and receive information from a server. From a high-level perspective, and maybe some research, what does this process look like? (Hint: HTTP)
4. (20 points) Describe the process of including your program in the boot sequence.
   a. Provide a short description of how you included your program in the boot sequence.
   b. Why might it be useful to include programs in the boot sequence? When might it not be useful?
5. (10 points) Conclusion: Summarize the takeaways you learned in this lab.

Submission

1. Include **generate-report.py** and **report.txt** in a file named **CSE157-Lab1.zip** and upload it to the appropriate assignment on Canvas by the check-off time.
2. Upload a PDF of your lab write-up by 11:59 pm by the deadline.

Submit as described above, then in a TA office hour or lab section, demonstrate Part 4 of the lab. The TA will only award you a grade if you have submitted your code.

Grading Scheme

Check off	5%
Write-up	10%