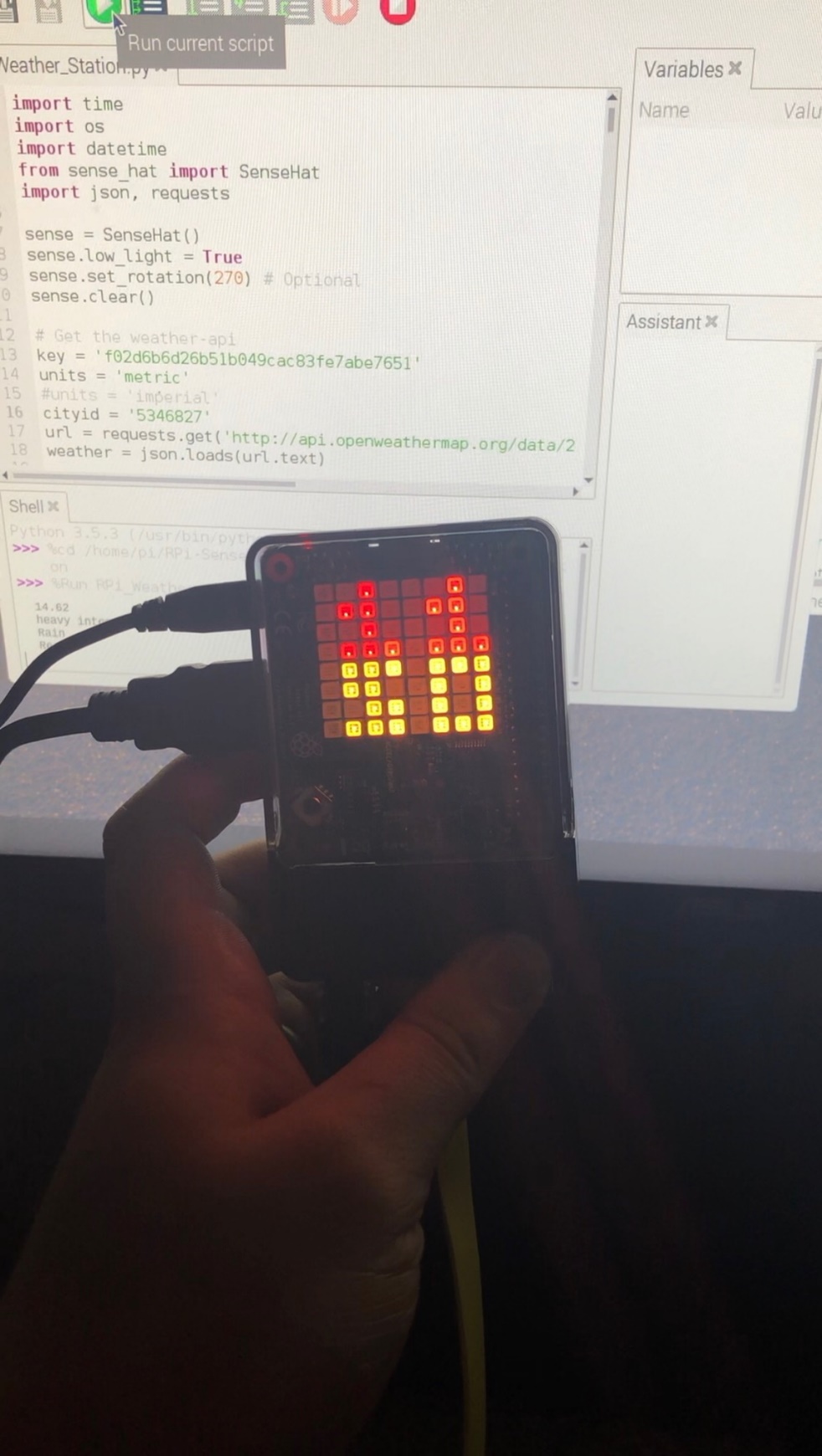


Hail, Rain, & Raspberry Pi

An In-Depth Review





# Project Overview:

Creation of Raspberry Pi Weather Station, Requiring implementation of basic coding, configuration and troubleshooting, as well as hardware implementation.

# Objective:

* Will be able to demonstrate basic knowledge of Raspberry Pi configuration
* Will learn how to implement outside technologies within coding software
* Demonstrate apprentice configuration skills within different software interfaces

# Standards:

* Fundamental networking skills
* Configuration and data insertion
* Hardware and cabling knowledge

# Requirements/Task(s):

# Acquire hardware (Raspberry Pi, Sense Hat, HDMI Cable, USB cable, Power adapter for wall, weather proof Raspberry Pi Case)

# Acquire Software, or files that will be vital as part of the project. ( Application Program Interface Key, .JSON file for location of weather reports, Raspbian, and Python)

# Create user account on <https://openweathermap.org/stations>

# Locate and note your specific application program interface key

# Now, you must download the City list, in the form of a .JSON file. This file was not viewable without an external file viewing application (File Viewer Plus)

# You must also locate your application programming interface, in the form of a URL- it can be found on the “API” section of the taskbar

# When locating your city from the list, cross reference a map, to ensure that you are using the correct city in the world based on longitude and latitude. For my project I will be using;< Escondido, Ca (92027)> <City ID: 5346827 >< Longitude: -117.086418 > <Latitude: 33.119209>

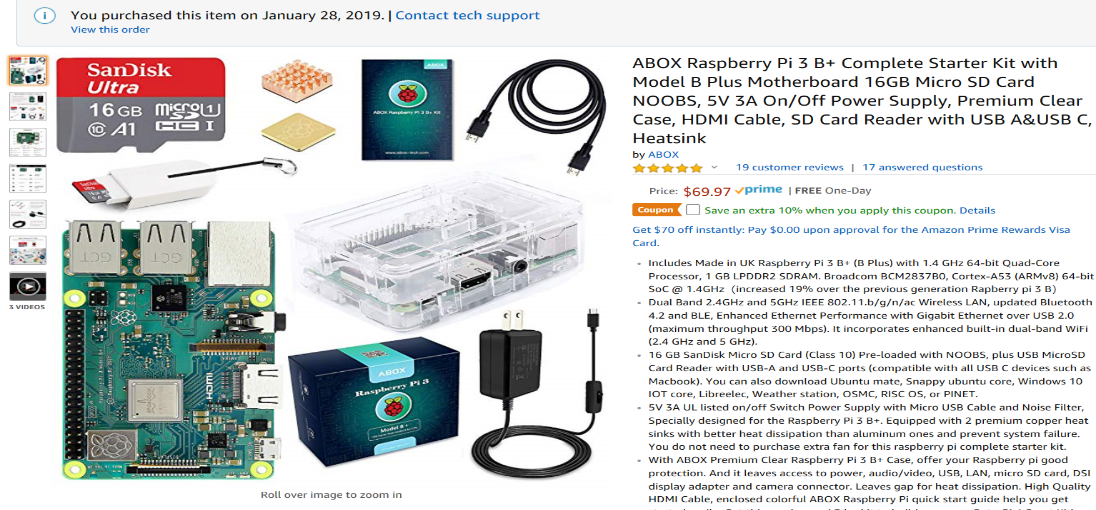
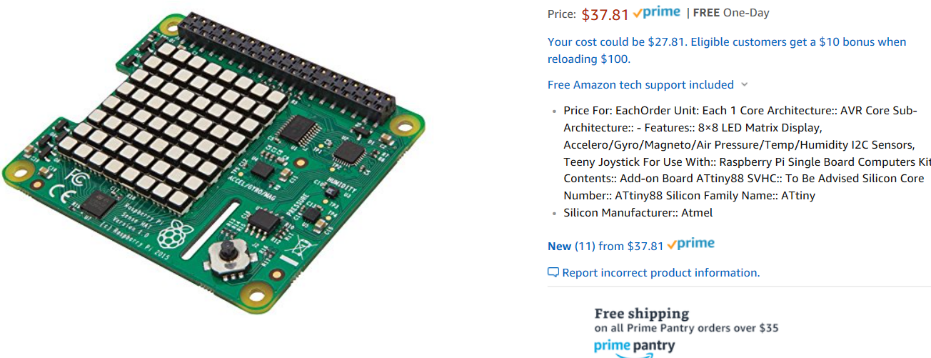
# From here, you must type the following into the PI’s terminal, in order for it to be properly implemented < sudo apt-get update && sudo apt-get upgrade >

# Then, you must access a developer file from GitHub, and Modify it to display your City ID, and your Application Program Interface Key.

# Now that you’ve done that, the final step is to chose whether you want to use the metric system, or if you’d rather view temperature in Fahrenheit.

# And BAM! just like that you’ve created your own personal Raspberry Pi Weather Station!

# Demonstration of Completion & Skills Gained

1. **Figure 1-1 and 1-2 Display necessary hardware assets outlined in my project**

**Figure 1-1** (SenseHat Add-on for Raspberry Pi, Features an LED display as well as Air pressure, Temperature, and Humidity Sensors)

**Figure 1-2** (Raspberry Pi Complete Starter Kit, Includes: Raspberry Pi 3 B+, 16GB micro SD card, NOOBS, 5V 3A Power Supply, Clear Rpi Case, HDMI cable, USB-A&USB-C cables, and an SD-card reader.

# 2. Figures 2-1 and 2-2 Display screenshots of multiple interfaces within the Openweathermap application, the URL format for the application programming interface is noted below.

# < [http://maps.openweathermap.org/maps/2.0/weather/{op}/{z}/{x}/{y}](http://maps.openweathermap.org/maps/2.0/weather/%7bop%7d/%7bz%7d/%7bx%7d/%7by%7d) >

# 

# 

**Figure 2-1** (A look inside client view of OpenWeatherMap application)

**Figure 2-2** (A look inside the application programming interface URL, and screengrab showing the chronological capabilities of the weather station.)

# 

**Figure 3-1** (File Viewer Plus 3, The viewing application I decided to implement, allows for the proper deployment of .JSON files)

# 

# 

**Figure 3-2** (Shows in the background different values for cities in the world, it comparatively shows that I have selected the correct “City of Escondido” in the .JSON file, by referencing latitude and longitude.

# 

**Figure 4-1** (A small part of the developer script required from GitHub, Circled are the values that can be changed based on geographical location, and choosing between Celsius and Fahrenheit.

# An important part to note, the City ID, and API key that you took down earlier will be necessary in the step outlined in *Figure 4-1.* Your application programming interface key will be input on line 13 whereas your City ID key will be input into line 16.

# After you have added the necessary values, run the script on python, and if everything is done correctly, then Voila! You’ve properly created an RPi weather station.

# Summarize what you learned:

Learned how to use Simple python commands, Raspbian SD card writing, basic networking and connectivity troubleshooting, and now possess a wider grasp of unknown user interface troubleshooting

**Bibliography**

1. ., OpenWeather®. "Сurrent Weather and Forecast." *Openweathermap*. N.p., 2012. Web. 29 Jan. 2019.
2. Amazon. "Online Shopping for Electronics, Apparel, Computers, Books, DVDs & More." *Amazon*. Amazon, 2019. Web. 29 Jan. 2019.
3. Bing, .. "Search Tool, Search the Web!" *Bing*. Microsoft, 2019. Web. 29 Jan. 2019.
4. Foundation, Raspberry Pi. "Teach, Learn, and Make with Raspberry Pi." *Rotate Display 90º? - Raspberry Pi Forums*. N.p., 2019. Web. 29 Jan. 2019.
5. Github. "Build Software Better, Together." *GitHub*. N.p., 2019. Web.
6. Https://templates.office.com/en-us/Project-based-learning-TM67409836. "Office Templates & Themes." *Word*. Microsoft, 2019. Web.
7. "Microsoft - Official Home Page." *Software Asset Management – Microsoft SAM*. N.p., n.d. Web. 30 Jan. 2019.
8. Raspbian , MoinMoin &, Python. "Quick Links." *FrontPage - Raspbian*. N.p., 2019. Web. 29 Jan. 2019.
9. Severinsen, Anders. "RPi Weather Station and Digital Clock." *Instructables.com*. AutoDesk Inc., 12 Oct. 2018. Web. 29 Jan. 2019.