IFB102: Mini-Project
Raspberry Pi Live Weather Station

Daniel Mendels – n11070871

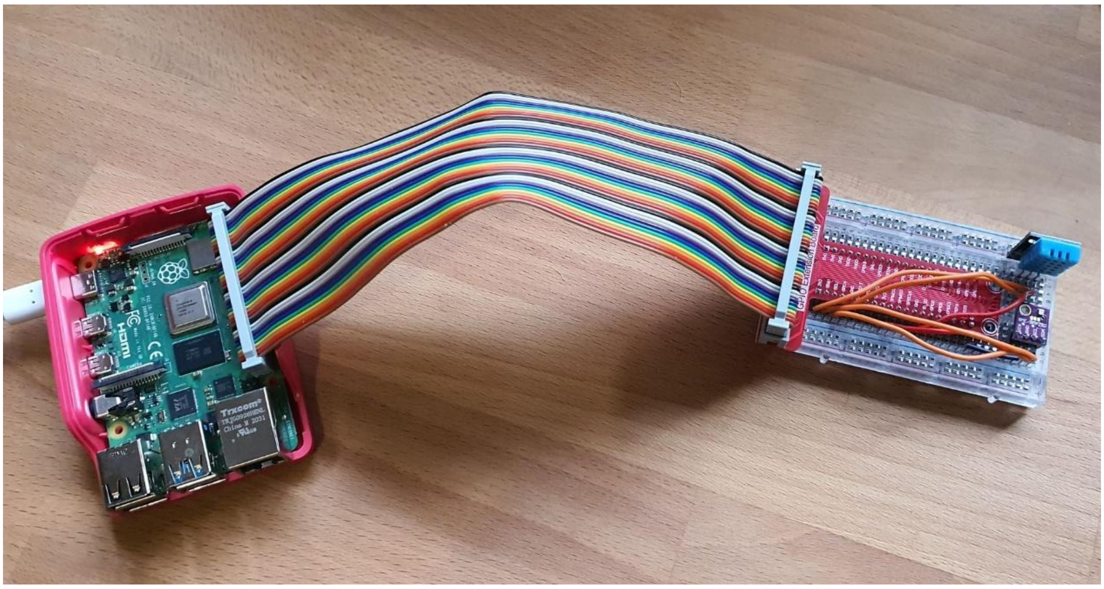


Figure 1: Raspberry Pi weather station hardware configuration

Contents

1.0 Project Objectives

This project aims to design and code a localised weather site embedded with live sensor readings, allowing users to monitor local weather conditions and predict future changes. The webpage will display current time, temperature, humidity, and pressure measurements.

Hosted on a Raspberry Pi 4, this project utilizes the Python programming language and open-source modules to read sensor data for injection into the websites HTML code.

The sensors required for this project are a DHT11 temperature and humidity sensor and a BMP280 Barometric Pressure sensor. These sensors must be wired into a breadboard connected to a GPIO extender for the Raspberry Pi, following the relevant installation manuals.

2.0 Review and Discussion of Technologies

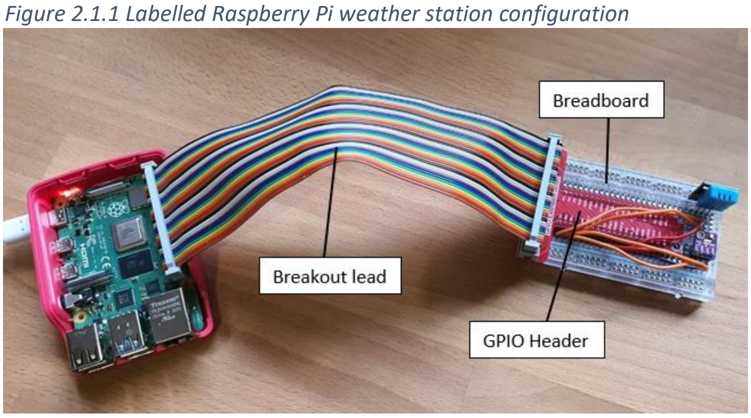
2.1 Hardware Components

GPIO Extension Board

A GPIO extension board was used to connect the sensors to the Raspberry Pi. The extension board allows for ease of installation of the two components without sacrificing efficient data transfer.

The GPIO extension board consists of 3 components: a breakout lead, header, and breadboard. The following figure depicts the Raspberry Pi and GPIO extension board configuration with all components labelled.

Figure 2.1.1 Labelled Raspberry Pi weather station configuration



All pins from the Raspberry Pi's GPIO component are directly wired into the corresponding pin on the GPIO header's integrated circuit. Labels on the GPIO header allow for ease of use and when connected to a breadboard, precise connections between the Raspberry Pi GPIO pins and sensor pins can be implemented.

Alternatives to the specific GPIO breakout lead, header and breadboard purchased for this project will make little difference. The only useful adjustment that can be made is implementing a larger breadboard to allow for additional sensors, as you are limited by the space available on the breadboard.

As explained in the course lectures, most modern computer components feature integrated circuits. These circuits, formed on small silicone chips, allow for traditionally large components to be manufactured thousands of times smaller. In this case, the GPIO header features over 40 separate wired connections in less than 30cm² (Jaycar, n.d.).

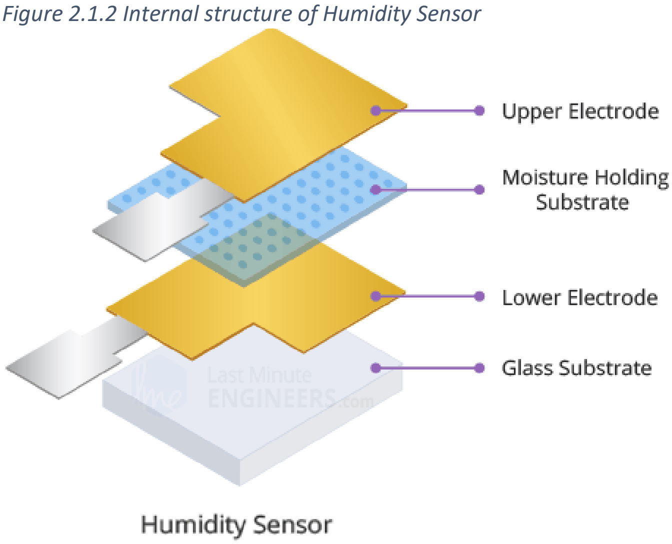
DHT11 Temperature and Humidity Sensor

To measure temperature and humidity the DHT11 was chosen. This is because it is a widely supported component which provided both temperature and humidity readings all-in-one. Additionally, prominent issues that occur during installation are well documented, with multiple peer-approved solutions found easily.

The DHT11 temperature and humidity sensor consists of two components: a humidity sensor and a thermistor. These components are wired into a microprocessor which converts the readings into 8bit data (Aosong, n.d.).

The humidity sensing component houses two electrodes separated by a moisture holding substrate (usually salt). As water vapor is absorbed by the substrate, ions are released which increase the electrodes conductivity. Changes in resistance between the two electrodes exhibit a proportional relationship to the relative humidity. Lower humidity is recorded when the resistance between the electrodes is greater, and vice versa. The microprocessor regularly measures the resistance and uses its' pre-calibrated relational data to extract the relative humidity measurement (Last Minute Engineers, n.d.).

Figure 2.1.2 Internal structure of Humidity Sensor



Note. Adapted from "Insight Into How DHT11 DHT22 Sensor Works", by Last Minute Engineers, n.d. (https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/)

The temperature measuring thermistor consists of a metallic oxide, pressed into a bead, and encased with glass. The encased material acts as a temperature dependant resistor, changing its' internal resistance in proportion to the temperature (Wavelength Electronics, 2013). Like the humidity sensor, this component has a pre-calibrated resistance to temperature conversion which is installed on the microprocessor to read the temperature.

There are many alternatives to the DHT11 as a temperature and humidity detecting device is commonly used in a wide range of projects. The most prominent alternative is the successor to the DHT11, the DHT22. This newer model of the component features a wider effective range with a more accurate tolerance. Additionally, the DHT11 has a faster sampling rate compared to the more accurate DHT22. These changes are detailed in the following table (Wavelength Electronics, 2013).

Table 2.1.1 DHT11 vs DHT22 key specifications comparison

Parameter	DHT11	DHT22
Humidity sensor effective range	20 - 80% (±5%)	0 - 100% (±2.5%)
Temperature sensor effective range	0 - 50°C (±2°C)	-40 - 80°C (±0.5°C)
Sampling rate	1 Hz (every second)	0.5 Hz (every 2 seconds)

Adapted from "Insight Into How DHT11 DHT22 Sensor Works" by Last Minute Engineers, n.d. (https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/)

For the expected operating range in Brisbane, where temperatures are consistently between 10°C - 45°C throughout the year, the drawbacks in effective temperature range of the older model are rarely exceeded. Inaccuracies may occur when humidity levels reach above 80%. However, with the limited time constraints of this project not allowing for imported components, and local suppliers not stocking the DHT22, the DHT11 will rarely fall short.

BMP280 Barometric Pressure Sensor

To accurately measure the atmospheric pressure the BMP280 was selected. The BMP280 was chosen, similarly to the DHT11, because it is a widely supported component and easy to install. Additionally, the BMP280 has well documented errors with peer-approved solutions.

The addition of a pressure sensor to the weather station project allows users to accurately predict incoming weather patterns. To understand the barometric pressure readings the following table has been provided, categorising pressure levels within their specified pressure range and the associated weather outcomes.

Table 2.1.2 Barometric pressure readings and their associated weather outcomes

Pressure Level	Pressure Range	Projected Weather
High Pressure	>1022 hPa	Clear skies and calm weather
Normal Pressure	1009 – 1022 hPa	Steady weather
Low Pressure	<1009 hPa	Warm air and rainstorms
Rapidly declining pressure in any pressure range is associated with incoming storms.		

Adapted from "How to Read a Barometer" by R. Oblack, 2020 (https://www.thoughtco.com/how-to-read-a-barometer-3444043)

The BMP280 uses a Piezo-resistive pressure sensing element to measure the atmospheric pressure to within ±1 hPa accuracy. The sensor element consists of four Piezo-resistors (whose resistance changes when strained) attached to the sides of a porous silicone diaphragm. The diaphragm bends either inwards or outwards depending on the atmospheric pressure, deforming the lattice structure, and causing a change in the band structure of the Piezo-resistors. Ultimately, this leads to a change in the resistivity of the component, which is read by the microprocessor and converted into a pressure measurement and sent to the Raspberry Pi as an 8bit data stream (Bosch Sensortec, 2015; METU MEMS Center, n.d.).

The most common alternatives to the BMP280 are the BMP180 (predecessor) and BME280 (successor). The only key difference being their absolute pressure accuracy and standard temperatures (0 - 60°C). The BMP280 and BME280 boast the most accurate readings at ±1 hPa, while the BMP180 has an accuracy of ±5 hPa (Bosch Sensortec, 2013, 2015, 2020). Another difference between the three iterations include the addition of a temperature and humidity sensor to the BME280. Since both the BME280 and BMP280 exhibit similar specifications, the most readily available iteration was implemented (BMP280).

Sensors were briefly touched on during the Week 8: Cloud Wearables IoT lecture as most IoT enabled devices use sensors to process their surroundings. One key difference being that this project does not store collected data in the cloud, instead updating the locally hosted website with the latest measurements.

2.2 Software Modules

Adafruit Libraries

Adafruit Industries provides free and open-source Python libraries to simplify interfacing with commonly used sensors. During this project, the Adafruit libraries for the DHT11 and BMP280 were installed onto the Raspberry Pi and their modules called in the weather stations backend Python code.

The Adafruit libraries provide extensive support for both the DHT11 and BMP280 through open-source code that reads the 8bit data received from the sensors and converts it into a tuple containing the recorded measurement/s. The Adafruit libraries are written and curated by professional programmers with the intent to make programming accessible and easy to learn (Adafruit, n.d.).

The only alternative to the Adafruit libraries is Pigpio, a dedicated library created to provide support for the most common Raspberry Pi GPIO sensors. However, due to the niche nature of the Python module and only one programmer actively updating the code, both the BMP280 and DHT11 no longer support interfacing via the Pigpio daemon (Pigpio, 2021).

Apache HTTP Server

Apache HTTP Server, also known as Apache webserver, is an open-source and free locally hosted webserver program. When used with the Raspberry Pi, Apache allows the user to edit and host a custom website on their local network. Apache was chosen as this projects website hosting program as it is supported on the Raspberry Pi, easy to install and had been previously covered in the course's practical exercises.

Apache establishes a connection between the local network and the viewer's browser, this connection is then used to send and receive packages. When a visitor wants to load a page, the visitor's browser sends a request to the server (in this case the Raspberry Pi). Apache responds by sending the requested files (html code, images, etc.) to the browser. The browser interprets the files, saves them to the local temporary directory and displays the webpage. The Apache program is entirely written in C and XML; interpreting requests and delivering responses instantaneously (Apache, n.d.; G., 2021).

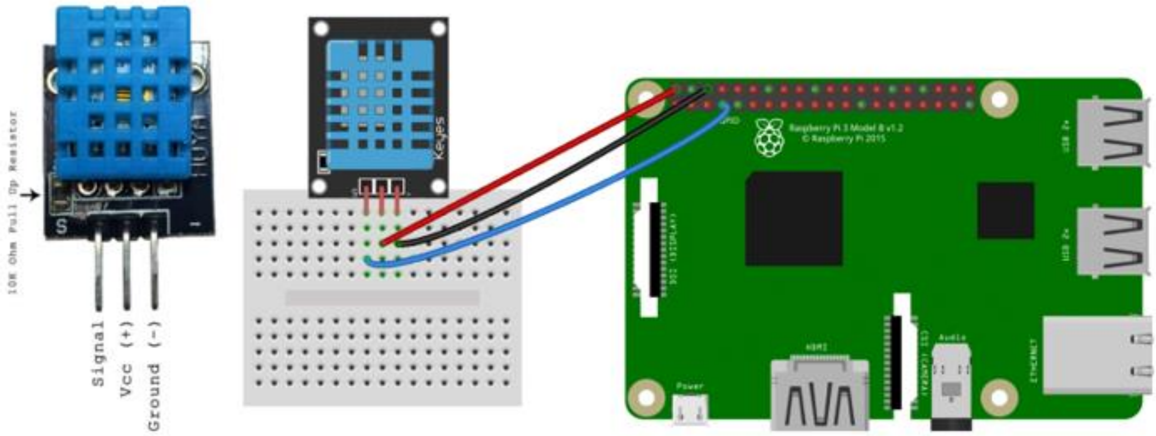
In the Week 6 practicals, the Apache webserver was briefly mentioned to host a local server with more complex HTML code than was shown in the practical lesson. Additionally, the Mini-Project Specification document links to an installation guide for the Raspberry Pi.

3.0 Design Implementation

3.1 Component Configuration

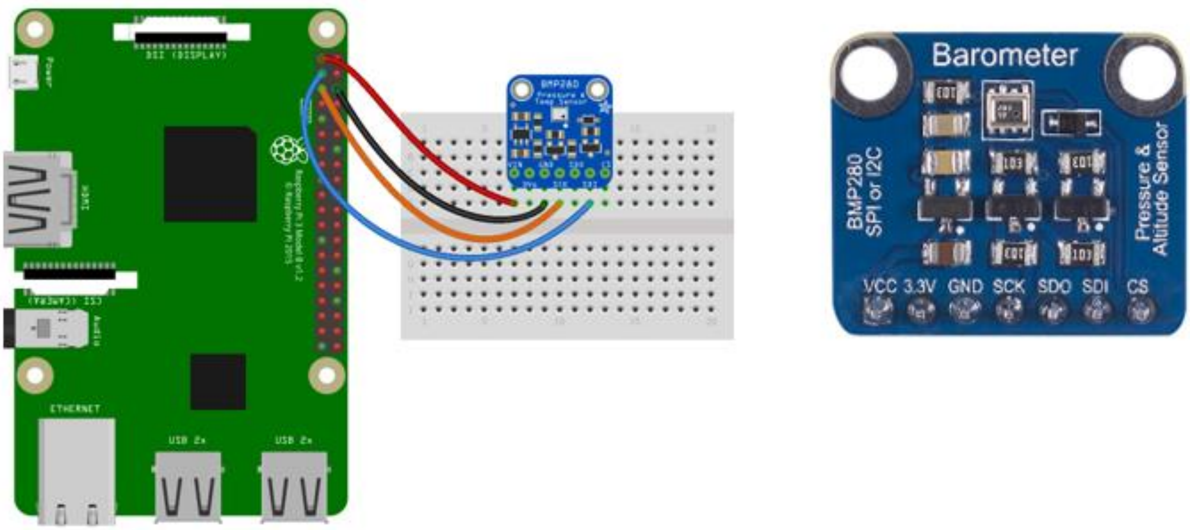
The Raspberry Pi weather station is configured to receive inputs from both the DHT11 temperature and humidity sensor and the BMP280 barometric pressure sensor. The components were wired to the 3V port as detailed below. Separate voltage in and ground pins on the GPIO extension board were used to allow for multiple sensors to be connected simultaneously.

Figure 3.1.1 DHT11 component and wiring diagram



Note. Adapted from "How to Set Up the Dht11 Humidity Sensor on the Raspberry Pi", By S. Campbell, 2020 (https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/)

Figure 3.1.2 BMP280 component and wiring diagram



Note. Adapted from "Python & CircuitPython Test", by Adafruit, n.d. (https://learn.adafruit.com/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout/circuitpython-test)

3.2 System Function

The Python file, run on start-up of the Raspberry Pi, has complete control of the system. In a continual while loop, the Python script calls the Adafruit module to read the current sensor data. This data is returned as a tuple and stored as variables for injection into the HTML code. After updating the HTML code, the Python script writes to the Apache server directory. When a request is made by the user's browser, Apache sends the latest HTML file to the browser for the browser to process and display (see Figure 3.4). This Python script repeats every 30 seconds. The following systems diagram details the exact path the sensor data takes before reaching the webpage.

Figure 3.2.1 Raspberry Pi live weather station systems diagram

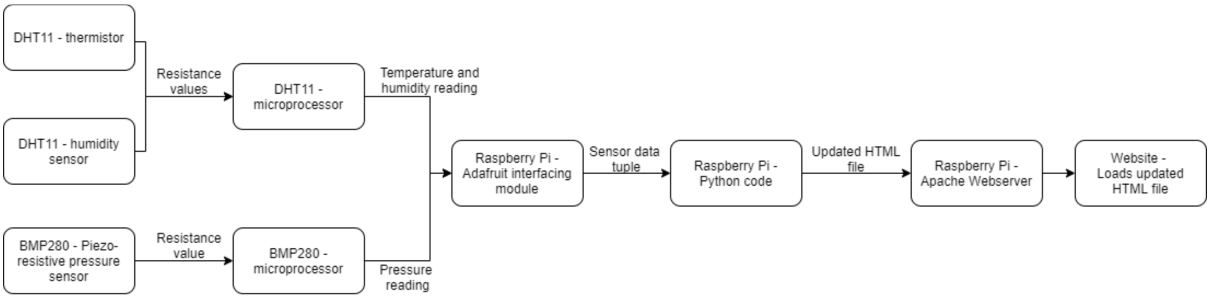
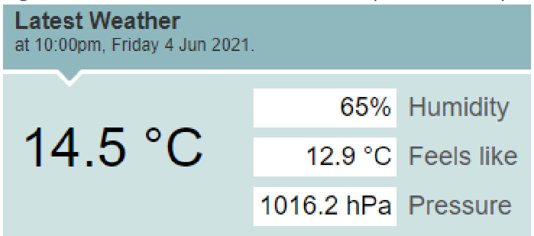


Figure 3.2.2 Raspberry Pi weather station webpage

3.3 Results

The accuracy of the data recorded by the DHT11 and BMP280 sensors were validated by comparing results displayed on the webpage with results from the Australian Bureau of Meteorology and a physical thermometer.

Figure 3.3.1 BOM Brisbane weather report at 10:00pm July 4, 2021

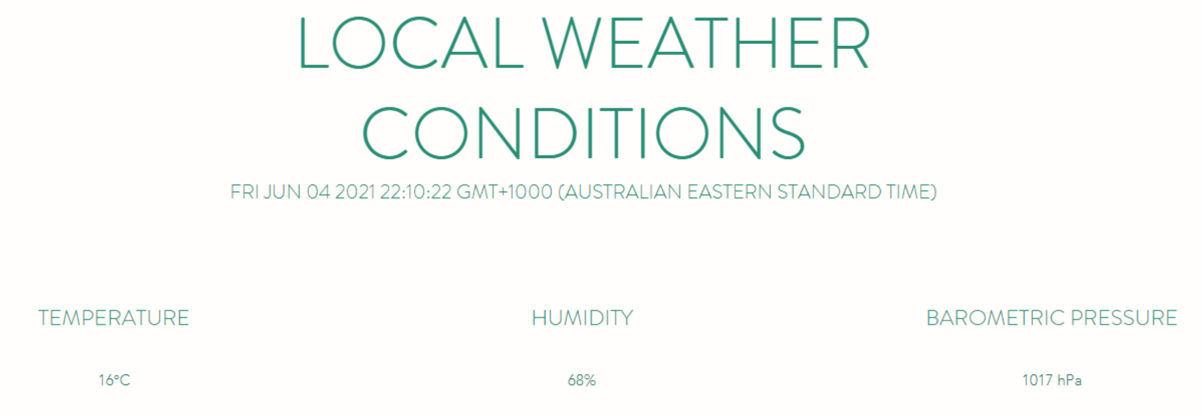


Note. From "Observations for Brisbane" by Australian Bureau of Meteorology, 2021, (http://www.bom.gov.au/places/qld/st-lucia/observations/brisbane/)

Temperature readings from both sources are shown below. At the time, the BOM reported a temperature of 14.5°C, while the Raspberry Pi weather station reported 16°C. Since the surrounding environment is known to play a large role in the local temperature, a physical thermometer was placed next to the weather station to confirm the local temperature as 16°C.

The pressure and humidity are less affected by the surrounding environment (assuming physical tampering of the weather station does not occur, e.g. breathing on the sensor), small changes in these measurements occur over a much larger area. Corroboration between the BOM report and the weather station webpage further proves the sensor data's accuracy.

Figure 3.3.2 Raspberry Pi weather station readings at 10:10pm July 4, 2021



3.4 Challenges

When implementing a project, many issues and setbacks arise. This project faced many pitfalls, requiring countless hours of extra work to ensure the system operational.

The first major problem arose when the Raspberry Pi failed to connect with the DHT11 temperature sensor. The first step in diagnosing the issue was to check that the sensor had an active voltage. A voltmeter was connected to the positive and negative terminals of the component and the voltage read 3.3V, the minimum required to power the component. Next, Python code from the component supplier and numerous other online sources was installed and failed to read the sensor. Ultimately, it was concluded that the sensor was faulty, and a replacement was required. After replacing the DHT11, the initial Python code worked successfully, and data was recorded live from the sensor.

The next major problem arose when attempting to inject the sensor data into the HTML code. In Python, there are four ways to insert a variable into a string. Three of which involve placing an identifier within the string to mark where a variable would be injected. These involved curly brackets or parentheses. As HTML code often contained these arguments, the Python interpreter was unable to discern the correct location for the variables and ultimately failed.

The final and less documented option adopted string concatenation, involving splitting a string, inserting the variable manually and then continuing the string afterwards. This final method proved useful when the placeholder HTML code was replaced with much more complex code.

3.5 Improvements and Future Directions

A significant improvement to this project would include storing the recorded data and providing live analyses. Recorded data could be displayed on live graphs based on the user-chosen period (24hrs – 7 days). Analysing the change in pressure over time is particularly important to predict future weather conditions. Real life applications of this data can be seen in weather reporting as high and low pressure systems move throughout the world, causing predictable weather events.

This improvement would allow users to generate highly-specified weather reports that are not currently detailed in national weather forecasting services.

4.0 References

Adafruit. (n.d.). About Us. Retrieved 4 June, 2021, from https://www.adafruit.com/about

Adafruit. (n.d.). Python & CircuitPython Test. Retrieved 4 June, 2021, from https://learn.adafruit.com/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout/circuitpython-test

Aosong. (n.d.). DHT11 Product Manual. Retrieved 4 June, 2021, from www.aosong.com

Apache. (n.d.). About the Apache HTTP Server Project. Retrieved 4 June, 2021, from https://httpd.apache.org/ABOUT_APACHE.html

Australian Bureau of Meteorology. (2021). Observations for Brisbane. Retrieved 4 June, 2021, from http://www.bom.gov.au/places/qld/st-lucia/observations/brisbane/

Bosch Sensortec. (2013). BMP180 Data Sheet. Retrieved 4 June, 2021, from https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf

Bosch Sensortec. (2015). BMP280 Data Sheet. Retrieved 4 June, 2021, from https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf

Bosch Sensortec. (2020). BME280 Data Sheet. Retrieved 4 June, 2021, from https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf

Campbell, S. (2020). How to Set Up the DHT11 Humidity Sensor on the Raspberry Pi. Retrieved 4 June, 2021, from https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/

G., D. (2021). What is Apache? An In-Depth Overview of Apache Web Server. Retrieved 4 June, 2021, from https://www.hostinger.com/tutorials/what-is-apache

Jaycar. (n.d.). Raspberry Pi GPIO Breakout Lead and Header for breadboard. Retrieved 4 June, 2021, from https://www.jaycar.com.au/raspberry-pi-gpio-breakout-lead-and-header-for-breadboard/p/XC9042

Last Minute Engineers. (n.d.). Insight Into How DHT11 DHT22 Sensors Work & Interface With Arduino. Retrieved 4 June, 2021, from https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/

METU MEMS Center. (n.d.). Piezoresistive Pressure and Temperature Sensor Cluster. Retrieved 4 June, 2021, from http://www.microsystems.metu.edu.tr/piezops/piezops.html

Oblack, R. (2020). How to Read a Barometer. Retrieved 4 June, 2021, from https://www.thoughtco.com/how-to-read-a-barometer-3444043

Pigpio. (2021). The pigpio library. Retrieved 4 June, 2021, from http://abyz.me.uk/rpi/pigpio/index.html

Wavelength Electronics. (2013). Thermistor Basics. Retrieved 4 June, 2021, from https://www.teamwavelength.com/thermistor-basics/

5.0 Demonstration Video

https://youtu.be/mHtYFayok5M