

GNG1103
Design Project User and Product Manual

CLIMATE SENSOR USER MANUAL

Submitted by:

[Design Algorithm, D3]

[Zach Léger, 300119396]

[Olivia Carnegie, 300189614]

[Michaela Jones, 300045591]

[Ogechi Ahunanya, 300135925]

[Julio Midence, 300170486]

[April 11, 2021]

University of Ottawa

Table of Contents

| | |
|---|-----------|
| Introduction | 1 |
| Overview | 2 |
| Cautions & Warnings | 3 |
| Getting started | 3 |
| Set-up Considerations | 5 |
| User Access Considerations | 7 |
| Accessing the System | 7 |
| System Organization & Navigation | 8 |
| 3.4.1 Hardware and Connection | 8 |
| 3.4.2 Code | 8 |
| 3.4.3 Protection | 8 |
| Exiting the System | 9 |
| Using the System | 10 |
| Arduino Uno/Microcontroller | 10 |
| Arduino Code | 11 |
| Connection Wires | 12 |
| Grove AHT20 | 13 |
| Troubleshooting & Support | 14 |
| Maintenance | 14 |
| Support | 14 |
| Product Documentation | 15 |
| <Subsystem 1 of prototype> | 15 |
| BOM (Bill of Materials) | 15 |
| Equipment list | 16 |
| Instructions | 17 |
| Testing & Validation | 19 |
| 6.2.1 Impact Resistance Tests | 19 |
| 6.2.2 Sensor Performance Tests | 20 |
| 6.2.3 Waterproof Fabric Tampering Tests | 21 |
| 6.2.4 Waterproofing Tests | 23 |
| Conclusions and Recommendations for Future Work | 24 |

| | |
|--------------------------------------|-----------|
| Bibliography | 25 |
| APPENDIX I: Design Files | 27 |
| APPENDIX II: Other Appendices | 28 |

List of Figures

Figure 1. Climate Sensor

Figure 2. Arduino Uno connection & casing

Figure 3. Sensor connection & casing

Figure 4. Sample code

Figure 5. Arduino Uno (*microcontroller*)

Figure 6. Arduino Case

Figure 7. Grove AHT 20 Sensor

Figure 8. Sensor Casing

Figure 9. Waterproof Mesh

Figure 10. Arduino Uno Components

Figure 11. Output of Data

Figure 12. Grove 4 Pin Male Connector wire

Figure 13. Climate Sensor

Figure 14. Sensor Casing and CAD Model

Figure 15. Arduino Casing

Figure 16. Sensor Casing on Drone Lid

Figure 17. Wire Holes in Sensor Casing

Figure 18. Graph showing the temperature and humidity values over time in a cold environment.

Figure 19. Graph showing the temperature and humidity values over time in a hot and humid environment.

Figure 20. Graph showing the temperature and humidity readings over time of the sensor without waterproof fabric coverage.

Figure 21. Graph showing the temperature and humidity readings over time of the sensor with waterproof fabric coverage.

Figure 22. Water poured on top of the waterproof fabric.

Figure 23. Empty bowl after letting the water sit on the waterproof fabric for a long period of time.

1 Introduction

This User and Product Manual (UPM) provides the information necessary for food delivery drones to effectively use the Climate Sensor and for prototype documentation.

The Climate Sensor was developed for JAMZ who desires to deliver food by drone in rural areas. JAMZ determined that they need to monitor the climate inside the delivery box. The assumptions made were that the delivery box was 1ft x 1ft x 1ft and that the extreme temperature delivered food can attain -17°C to 74°C . This UPM will go through the set-up of the Climate Sensor, how to use the Climate Sensor, how to troubleshoot and obtain support if necessary, documentation of the conception of the Climate Sensor and Validation of the feasibility of the Climate Sensor. The intended audience is individuals who wish to monitor the climate (temperature and humidity) inside a delivery box.

2 Overview

The problem at hand is that JAMZ requires a way to determine and alert the flight operator when the temperature and/or humidity values go out of optimal range ($\pm 10\%$ room temperature and humidity). The solution to this problem is critical so JAMZ is aware when food being delivered is no longer in optimal condition and stores the occurrences as insurance to customers. Furthermore, the solution can be used to improve the conditions for the food in the delivery box in the future.

The fundamental needs of the user are readable and understandable data, flag data, dimension and compactibility and light weight add-on. It is essential that the data is easily readable and understandable so the flight operator, who is monitoring multiple drones, can quickly determine if the temperature and/or humidity values are not in range. This ties into the users choice for flag data, which only gives warning when values are out of range, so that the flight operator is not overwhelmed with a lot of incoming data. The dimensions and weight are important so that the aerodynamics of the drone are not impacted when the add-on is implemented to the drone.

Our product has a sensor that can collect humidity and temperature values at whatever frequency the user desires (can be faster than the desired 2 Hz speed) which can be easily edited. The code of our product can be easily manipulated to alert at different temperature and humidity ranges. Furthermore, the installation is quick and easy to follow. The sensor is able to measure a large range of temperature and humidity values outside of typical food conditions to ensure data accuracy. The product comes with waterproof casing to ensure durability.

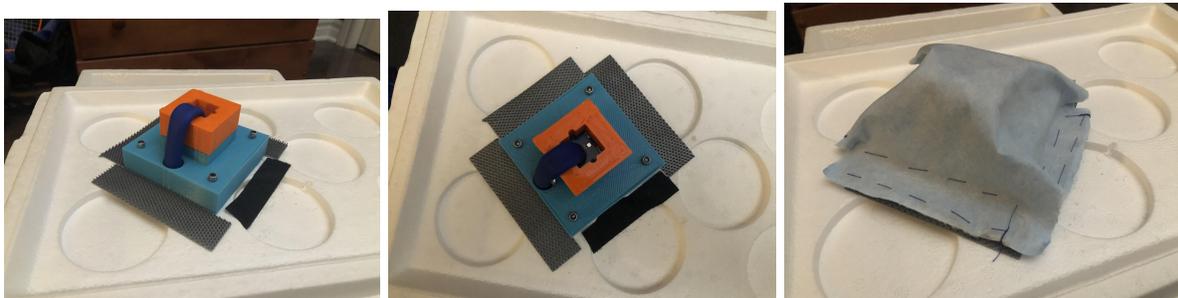


Figure 1. Climate Sensor (Ahunanya et al. 2021)

The key features of the product are: the included waterproof casing to ensure durability, easy-to-edit code for the user's desired conditions, accurate data, quick and easy installation, temperature readings from -40°C to 85°C and humidity readings from 0% - 100% relative humidity.

The architecture of the system consists of an unified temperature and humidity sensor that gathers data at the users desired speed with the help of a controller, Arduino Uno, that sends it to another controller, Raspberry Pi, which sends the tabulated data to the flight operator.

2.1 Cautions & Warnings

DO NOT use the Climate Sensor in unintended environments. The intended temperature environment is from -40°C to 85°C . The intended humidity environment is from 0% - 100% relative humidity.

3 Getting started

In order to initiate the system, all hardware parts, including the microcontroller and the sensor must be properly connected with wires. Once everything has been properly connected we recommend applying all safety and protection features, including casings and waterproof coverings. From there both the microcontroller and the sensor can be placed wherever desired and then the code can be run.



Figure 2. Arduino Uno connection & casing (Ahunanya et al. 2021)

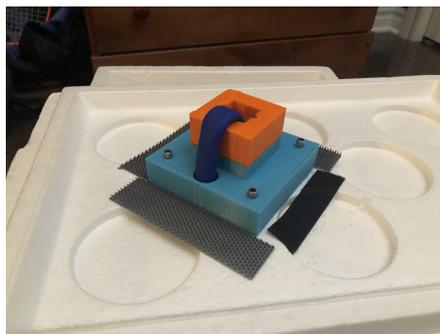


Figure 3. Sensor connection & casing (Ahunanya et al. 2021)

While the code runs, the sensor will gather information about the environment's humidity and temperature which will be transferred via connection wires and inputted into the arduino (Ahunanya et al. 2021). Once imputed the information will be outputted via the arduino. The output will then be displayed according to the way in which the code specifies (Ahunanya et al. 2021).

In order to exit the system the code can be stopped from running and for much smaller scale applications the usb cord from a computer can simply be unplugged.

Features of this system that must be considered before running this code include... range and warning statement customization as well as rate of data received. Before running the code temperature ranges can be altered to fall anywhere between -40 and 85 degrees celsius (Ahunanya et al. 2021). Warning statements can be customized to display both, one or neither of the aspects that is out of range (humidity or temperature). Lastly, that rate of data can be sped up to a maximum of 2 hertz and slowed down to any value (Ahunanya et al. 2021).

The screenshot shows a serial monitor window titled 'COM3'. It displays a series of data points and warning messages. The data points consist of a timestamp, temperature in degrees Celsius, and humidity percentage. The warning messages are 'WARNING, HUMIDITY IS TOO HIGH' and 'WARNING, TEMPERATURE AND HUMIDITY TOO HIGH'. The data points start at 01:22.50s and end at 01:44.00s. The temperature values range from 24.65°C to 24.79°C, and the humidity values range from 60.49% to 76.01%.

| Timestamp | Temperature (°C) | Humidity (%) | Warning |
|-----------|------------------|--------------|--|
| 01:22.50s | | | |
| 01:23.00s | | | |
| 01:23.50s | | | |
| 01:24.00s | | | |
| 01:24.50s | | | |
| 01:25.00s | | | |
| 01:25.50s | | | |
| 01:26.00s | | | |
| 01:26.50s | | | |
| 01:27.00s | | | |
| 01:27.50s | | | |
| 01:28.00s | | | |
| 01:28.50s | | | |
| 01:29.00s | | | |
| 01:29.50s | | | |
| 01:30.00s | | | |
| 01:30.50s | 24.65°C | 60.49% | WARNING, HUMIDITY IS TOO HIGH |
| 01:31.00s | 24.65°C | 61.25% | WARNING, HUMIDITY IS TOO HIGH |
| 01:31.50s | 24.69°C | 62.37% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:32.00s | 24.70°C | 63.28% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:32.50s | 24.69°C | 64.11% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:33.00s | 24.68°C | 64.88% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:33.50s | 24.70°C | 65.68% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:34.00s | 24.70°C | 66.42% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:34.50s | 24.70°C | 66.99% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:35.00s | 24.70°C | 67.55% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:35.50s | 24.71°C | 68.13% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:36.00s | 24.70°C | 68.72% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:36.50s | 24.70°C | 69.26% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:37.00s | 24.71°C | 69.81% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:37.50s | 24.72°C | 70.39% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:38.00s | 24.71°C | 70.88% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:38.50s | 24.73°C | 71.36% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:39.00s | 24.74°C | 71.85% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:39.50s | 24.74°C | 72.34% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:40.00s | 24.75°C | 72.84% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:40.50s | 24.76°C | 73.26% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:41.00s | 24.75°C | 73.72% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:41.50s | 24.78°C | 74.13% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:42.00s | 24.79°C | 74.54% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:42.50s | 24.77°C | 74.92% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:43.00s | 24.79°C | 75.30% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:43.50s | 24.78°C | 75.65% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |
| 01:44.00s | 24.79°C | 76.01% | WARNING, TEMPERATURE AND HUMIDITY TOO HIGH |

At the bottom of the window, there are two checkboxes: 'Autoscroll' (checked) and 'Show timestamp' (unchecked).

Figure 3. Sample code (Ahunanya et al. 2021)

3.1 Set-up Considerations

The Arduino Uno is a microcontroller that is able to read or pick up on certain inputs and then turn it into an output. In the case of this subsystem, the input the arduino will be receiving is from the Grove AHT 20 sensor (Ahunanya et al. 2021). With help from some simple coding an output will be produced.



Figure 5. Arduino Uno (microcontroller)

Since this subsystem is primarily made for drone delivery a casing for the arduino is required. In order to protect it from damage due to shaking and/or falling the arduino must be placed in the following case before the microcontroller is placed in its desired spot (Ahunanya et al. 2021).



Figure 6. Arduino Case (Makerbot, 2015)

Next the Grove AHT 20 sensor will be responsible for providing the arduino with the appropriate information for input. The pins on the sensor are what will sense humidity and temperature values.



Figure 7. Grove AHT 20 Sensor (Zuo, 2020)

Since this subsystem is primarily made for drone delivery a casing for the sensor is also required. In order to protect the sensor from and physical damage the AHT 20 sensor must be placed in the following case. The provided case also has 4 small holes so that the sensor and case can be screwed into a desired place (Ahunanya et al. 2021).

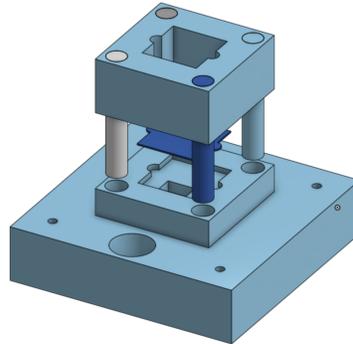


Figure 8. Sensor Casing (Ahunanya et al. 2021)

Before any cases are put on or cases are screwed in, it is crucial that the provided wires are put in the correct locations in both the sensor and the arduino. Once the correct locations are found a rubber wire covering should be placed over all wires to protect them from any physical strain, water or general damage. Lastly, it is important to ensure that the velcro waterproof mesh is securely attached in order to fully protect the subsystem.



Figure 9. Waterproof Mesh (Ahunanya et al. 2021)

3.2 User Access Considerations

This subsystem has been primarily designed for drone usage, in particular drone food delivery use. Although it could be used in many other instances. This subsystem has been designed to track if an environment stays within a specified humidity and temperature range. This

“desire range” is easily changeable within the code. Although this specific sensor, Grove AHT 20, is only capable of detecting values between -40 and 85 degrees celsius(Ahunanya et al. 2021).

Other than that restriction, that warning system can be very customizable. Currently, the subsystem default is to alert the user when temperature values are out of this range: 10% from STP (25 degree celsius) and/or when humidity values are out of this range: 50-55% (Ahunanya et al. 2021). The default warning provides the user with both the humidity and temperature when one or both are out of range at a particular time stamp, though this can be altered and customized within the code.

3.3 Accessing the System

In order to begin setting up this subsystem, it is first essential to ensure all equipment is obtained. Please ensure that you have an Arduino Uno, a Grove AHT 20, attachment wires, access to provided code, screws (if required), as well as both cases, waterproof mesh covering and rubber wire covering (for additional protection).

Next, begin by altering the code if needed (require alternate ranges or warning statements). Once any changes are made, start placing the appropriate wires in the microcontroller and sensor. We recommend at this point the code should be run to and tested to confirm all values and functions are working in the desired way.

Once the system is confirmed to be working how desired we would also suggest putting the rubber wire mesh on at this point, followed by both cases. After that, the microcontroller can be placed where desired, while the sensor can also be placed where desired and additionally screwed into place. Lastly we recommend the velcro waterproof mesh is placed on the sensor and case. At this point the code can be run and the subsystem is ready to go!

3.4 System Organization & Navigation

The system itself can be broken down into about three main sections; hardware and connection, code, and casing and protection.

3.4.1 Hardware and Connection

The hardware and connection system refers to the physical portion of this subsystem, this includes anything surrounding the Arduino Uno, the Grove AHT 20 as well as the connection between them.

The main function of this system is to ensure that the proper information is accurately being detected by the sensor, quickly being transferred via the wires, and last being inputted into the microcontroller in order to be outputted in the desired way. A feature of this sensor is that

inputs and outputs can be delivered at a fast rate of 2 hertz, meaning that the sensor can go as fast as that or any amount slower.

3.4.2 Code

The code system refers to all aspects implemented while programming the subsystem. The main function of the code is turn the output provided by the microcontroller into desired readable data and warning statements. A feature of this code is that it was designed to be customized. Desired ranges within -40 and 85 degrees celsius as well as the extent of warning statements can all be easily altered.

3.4.3 Protection

The protection system refers to any casing or steps taken to protect the hardware and connection system. The main function of this system is obviously to protect but also to ensure nothing surrounding the system has the opportunity to take away from the accuracy of the data.

The arduino and sensor cases are used to protect them from any shaking, dropping or damage to the sensor pins. The rubber wire covering is used to protect the wire from any direct water as well as just damage over time. The rubber wire covering also acts as a form of organization, to have all wires in one place. Lastly the waterproof mesh to be placed on the sensor and case it shields the sensor from any water, which if exposed could skew results and accuracy (Ahunanya et al. 2021).

3.5 Exiting the System

In order to properly exit the system the code can simply be turned off and/or stopped from running. For smaller scaled uses the arduino can also be disconnected from its power source.

4 Using the System

The climate sensor made by Design Algorithm is very simple to implement. The main purpose of the climate sensor is to gather temperature and humidity readings from the inside of the delivery box. As mentioned before, the implementation of the design is quite simple. There are three components that play an important role. The first component is the Arduino Uno which is the microcontroller being used. Then comes the Grove AHT20. This is the device that gathers the temperature and humidity of its surroundings. Finally, the connector wires are used to transmit the data from the AHT20 to the Arduino Uno.

The following subsections provide detailed, step-by-step instructions on how to use the various functions or features of the Climate Sensor.

4.1 Arduino Uno/Microcontroller

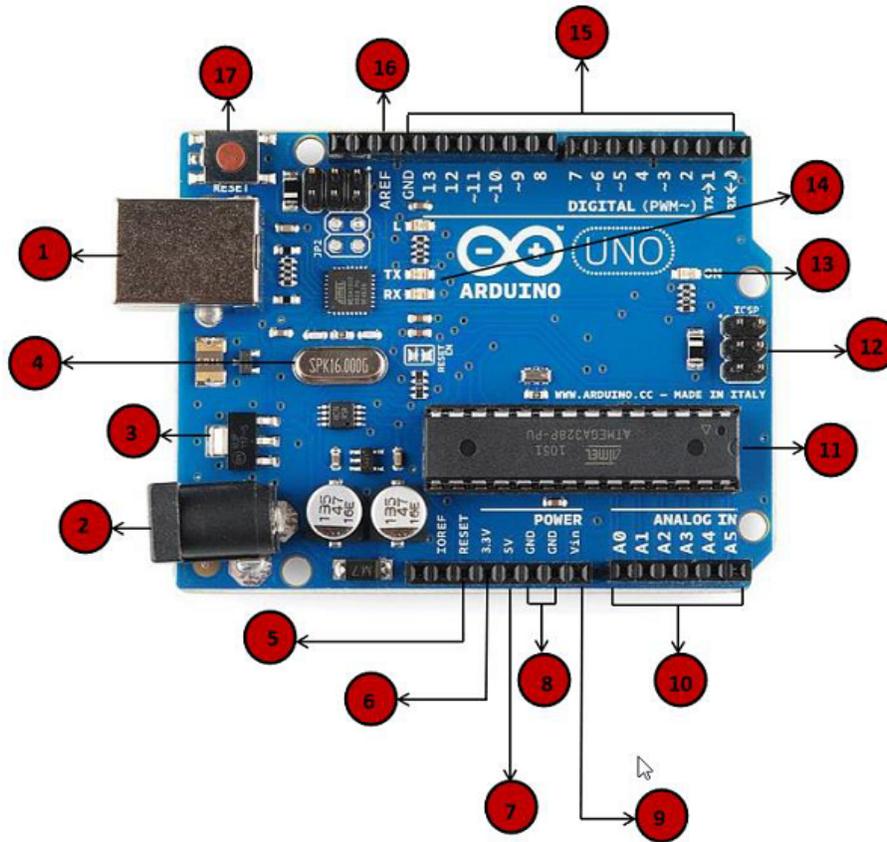


Figure 10. Arduino Uno Components (TutorialsPoint)

Figure 9 breaks down all the different components of the Arduino Uno. For the climate sensor there are some areas that need to be highlighted.

The Arduino Uno is responsible for receiving the data from the AHT20 sensor and outputting it however it has been programmed. Section 1 of the figure points to the way to power and program the Arduino. A USB cable connects the Arduino to a computer. The computer powers the microcontroller and is able to program it using the Arduino IDE. Section 7 & 8 are used to power any sensors connected to the Arduino. Section 16 is responsible for the gathering of the data from the sensor. The implementation of all of these sections will be explained later on during the documentation.

4.1.1 Arduino Code

```

COM3
0:0:22.50s
0:0:23.00s
0:0:23.50s
0:0:24.00s
0:0:24.50s
0:0:25.00s
0:0:25.50s
0:0:26.00s
0:0:26.50s
0:0:27.00s
0:0:27.50s
0:0:28.00s
0:0:28.50s
0:0:29.00s
0:0:29.50s
0:0:30.00s
0:0:30.50s      24.65°C      60.49%      WARNING, HUMIDITY IS TOO HIGH
0:0:31.00s      24.65°C      61.25%      WARNING, HUMIDITY IS TOO HIGH
0:0:31.50s      24.69°C      62.37%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:32.00s      24.70°C      63.28%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:32.50s      24.69°C      64.11%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:33.00s      24.68°C      64.88%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:33.50s      24.70°C      65.68%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:34.00s      24.70°C      66.42%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:34.50s      24.70°C      66.99%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:35.00s      24.70°C      67.55%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:35.50s      24.71°C      68.13%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:36.00s      24.70°C      68.72%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:36.50s      24.70°C      69.26%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:37.00s      24.71°C      69.81%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:37.50s      24.72°C      70.39%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:38.00s      24.71°C      70.88%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:38.50s      24.73°C      71.36%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:39.00s      24.74°C      71.85%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:39.50s      24.74°C      72.38%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:40.00s      24.75°C      72.84%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:40.50s      24.76°C      73.26%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:41.00s      24.75°C      73.72%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:41.50s      24.78°C      74.13%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:42.00s      24.77°C      74.54%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:42.50s      24.77°C      74.92%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:43.00s      24.79°C      75.30%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:43.50s      24.78°C      75.65%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
0:0:44.00s      24.79°C      76.01%      WARNING, TEMPERATURE AND HUMIDITY TOO HIGH
 Autoscroll  Show timestamp

```

Figure 11. Output of Data (Ahunanya et al. 2021)

Figure 10 shows how the data gathered from the sensor will be outputted.

The layout of the outputted data can be easily changed. The code has been made very human readable and includes lots of comments to guide the person looking at it. In order to program the Arduino and tell it what to do, the Arduino IDE must be used.

4.2 Connection Wires

Once the Arduino has been programmed, connection wires are used to connect the AHT20 sensor to the Arduino. The grove 4 pin connector wire is used to connect the Sensor and the microcontroller.



Figure 12. Grove 4 Pin Male Connector wire (Ahunanya et al. 2021)

Figure 11 shows the wires used to connect the components of the design.

The white end of the wire is the part that connects to the AHT20 sensor. The black pins are connected to the Arduino. The location of where the black pins are being connected will be explained during the discussion of the AHT20 sensor.

4.3 Grove AHT20

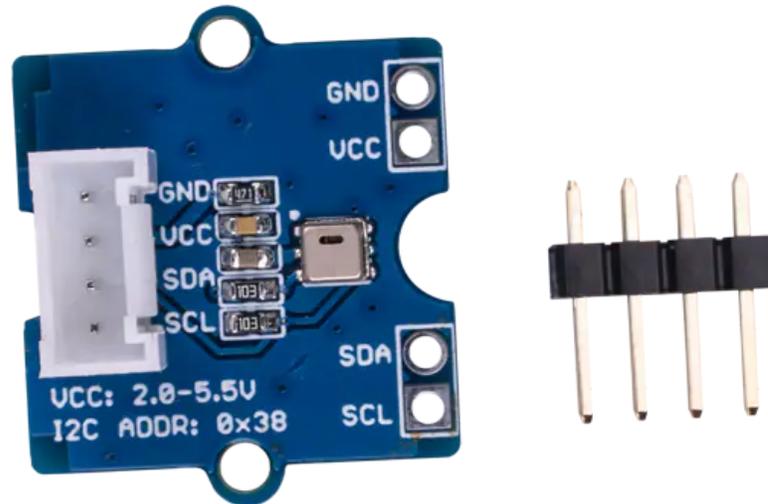


Figure 13. Climate Sensor (Ahunanya et al. 2021)

Figure 12 shows the sensor used in our design. The Grove AHT20 was used because of its simple implementation and reliability.

The white end of the connector wire is connected like a puzzle piece into the white section of the sensor. In order to connect the wire to the Arduino successfully, the pins have to be chosen carefully. The wire connected to “GND” has to be connected to either of the two “GND” pins on the Arduino. “VCC” corresponds to the “5V” pin on the Arduino. “SDA” corresponds to the “SDA” pin, and “SCL” corresponds to the “SCL” pin of the Arduino.

5 Troubleshooting & Support

There are a couple of aspects that need to be taken into account when it comes to the failure of the system. Any damage to either the microcontroller, wires or sensor are of utmost importance. All three of them are dependent on each other. With no Arduino Uno, the values can not be gathered and outputted. Without the AHT20 sensor, no temperature and humidity readings can be done. Lastly without the wires, no data can be transmitted from the sensor to the microcontroller.

5.1 Maintenance

In order to avoid failure there are a couple of simple steps that can be taken. The most important one would be to make sure all the wires are connected in the correct place. In section 4, specifically 4.3 it says how to properly connect all the wires. The state the wires are in could play a role in the transmission of data. The wires are protected by a rubber tubing, but making sure the wires themselves are not ripped or damaged is very important.

5.2 Support

The people below are the ones who made the final design. Their contact information will be added just in case any issues arise regarding the user manual.

Julio Midence: jmide091@uottawa.ca (Electronics and Software)

6 Product Documentation

A lot was learned during the prototyping and testing process of this subsystem and we were all able to use what we learned in order to improve our design and come up with the final product. Once we were able to print out and test the AHT20 climate sensor and Arduino Uno casings, we had a better understanding of what elements from that design worked well and what elements needed to be improved upon. We were also able to test the code very frequently and improve it to better fit our needs every time we did so.

Changes to the original case designs were made for the final prototype so that everything would fit together tightly with very little chance of anything coming apart or breaking. The problem we noticed with the first and second prototypes of the climate sensor was how fragile it was and how the different pieces could come apart a little too easily. This was obviously not acceptable since its entire purpose is to house and protect the sensor. The sides of the case as well as the connection pegs were made to be thicker and stronger in order to solve these issues using the same plastic printing material.

In the final prototype we were also able to solve the waterproofing issue. It took a few weeks to finally decide on the best way to go about waterproofing the subsystem, but in the end we found that a rubber hose connected to the sensor and the arduino to hold the wires and a sheet of polyethylene material covering any other exposed components would be the best solution. No other materials were tested with the prototypes. The main concern with covering the climate sensor with a different material was that it wouldn't perform up to the expected standard, however, after further testing was done it was found to not have an effect on the data collected.

These improvements to the subsystem means that consistent and accurate data can be given to the flight operator while the subsystem is also strongly resistant to the elements and any accidental damages that could occur.

6.1 <Subsystem 1 of prototype>

6.1.1 BOM (Bill of Materials)

| Materials | Link to Materials | Price (CAD) | Quantity Needed | Function/Purpose | Did we already have it? |
|----------------------------|--|-------------|-----------------|---|-------------------------|
| Grove AHT20 climate sensor | Link to climate sensor | \$6.66 | 1 | -The temperature and humidity sensor will be attached to the inside of the lid of the delivery package. | No |

| | | | | | |
|-------------------------------------|---|---------|------------------------------|---|-----|
| | | | | -The sensor will be connected to an Arduino using the wires below. The sensor will be used to gather data and transmit it to the drone operator. | |
| Grove 4PIN Male Jumper Wires (5pkg) | Link to jumper wires | \$3.94 | 1 | -These are the wires that will run from the Arduino to the climate sensor -The wires will be used to power and gather data from the sensor | No |
| Wire Tubing | Link to wire tubing | \$6.99 | 1 (cut to 20cm long) | -The flexible tubing will protect the wires from possible damages and keep them all together | Yes |
| M3 screws / wire | Link to wire | \$11.99 | 4 pieces | -The screws will hold the AHT20 climate sensor casing secure to the box lid | No |
| Polyethylene Waterproof Fabric | Link to waterproof material | \$11.99 | 1 sheet (cut to 20cm x 20cm) | -The waterproof fabric will guarantee the protection of the sensor and all other covered components from any condensation from hot food or from any unexpected events involving water | No |
| Velcro Ribbon | Link to velcro | \$3.99 | 4 strips (20cm long) | -This will be used to secure to Polyethylene waterproof fabric in place on the lid | Yes |

6.1.2 Equipment list

Equipment list

- 3D printer
- Filament for the 3D printer
- Arduino Uno
- USB cable
- Grove AHT20 climate sensor
- Grove 4PIN Male Jumper Wires

- Velcro ribbon (and possibly a needle and thread)
- Waterproof material (polyethylene)
- M3 Screws
- Rubber hose for wire covering
- CAD modeling software (ex: Onshape)

6.1.3 Instructions

1. Purchase or acquire Arduino Uno, Grove AHT20 climate sensor, as well as all other necessary materials and equipment.
2. Find appropriate CAD modeling software (such as Onshape).
3. Design protective casing for both the Arduino Uno microcontroller and the Grove AHT20 climate sensor using desired CAD modeling software.

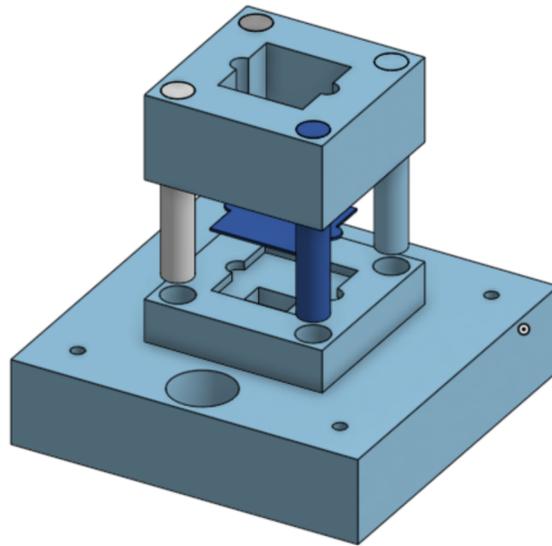


Figure 14. Sensor Casing and [CAD Model](#) (Ahunanya et al. 2021)



Figure 15. Arduino Casing (Makerbot, 2015)

4. Print the protective casings you have designed using a 3D printer and an appropriate printing material.
5. Place the Arduino Uno and the Grove AHT20 climate sensor in their respective casings and secure them into place within the casing.
6. Find the placement of the sensor casing on the lid of the box and drill a hole for the wires in the correct location to the size of the rubber hose so that it fits tightly.
7. Attach the Grove AHT20 climate sensor casing to the lid of the box using the M3 wire.

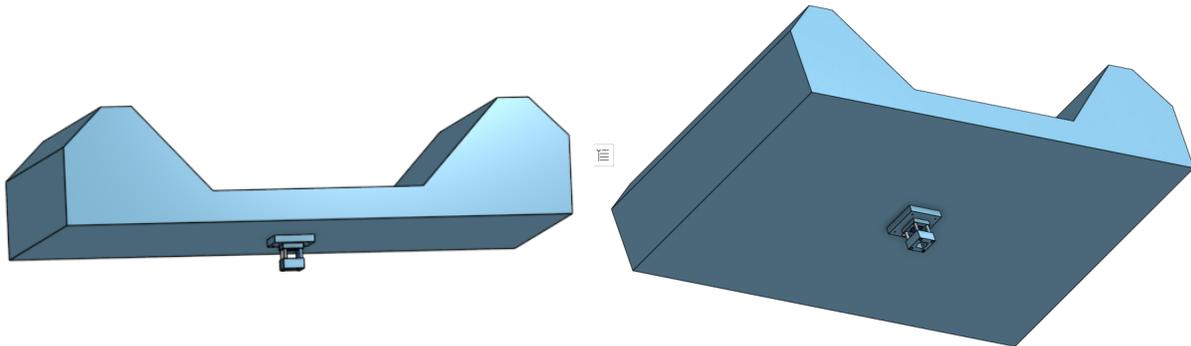


Figure 16. Sensor Casing on Drone Lid (Ahunanya et al. 2021)

8. Feed the rubber hose through the hole you have just created in the lid of the box and the hole created for it in the sensor casing.

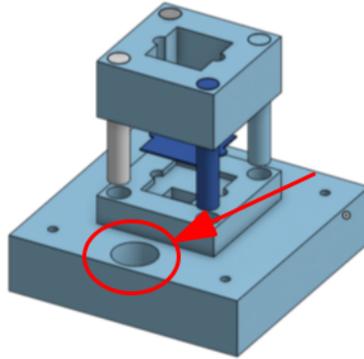


Figure 17. Wire Holes in Sensor Casing (Ahunanya et al. 2021)

9. Attach the wires to the climate sensor.
10. Feed wires through the rubber hose and secure the hose to the Grove AHT20 climate sensor at the wire attachment point.
11. Attach the wires to the appropriate pins of the Arduino Uno (which will be located on the opposite side of the box to the climate sensor). This information can be found in section 4.1.
12. For testing purposes you can now simply connect the arduino to your computer using a USB cable. For actual use, you will now need to connect the Arduino Uno to a DC converter in order to bring the voltage down to 5V.
13. Finally, run the code and observe your results.

6.2 Testing & Validation

For the climate sensor add-on, many tests were conducted on the prototype, such as testing the casings' impact resistance, the sensor's performance in different climates, the effect of the waterproof fabric on the data collection and how well the waterproof fabric works.

6.2.1 Impact Resistance Tests

For the impact resistance tests, they mostly consisted of multiple drop tests done at different heights. For the arduino casing, the top portion of the case would always pop off after feeling the impact of the drop. This issue was not too important since the arduino was going to be stored in another casing compartment of the drone. For the sensor casing, it performed well in the drop tests. The top and bottom pieces of the casing never came apart after any of the drop tests. Also, the pieces of the sensor casing were pretty thick and strong, so the pieces themselves never took any structural damages.

6.2.2 Sensor Performance Tests

To test the performance of the sensor, we put the sensor in extreme environments to see how well it would read the temperature and humidity values. We already knew that the sensor worked at a normal climate (room temperature climate) as shown in **Section 6.2.3**. For the first test, we put the sensor in a freezer to simulate a cold environment. The results are shown in the figure below.

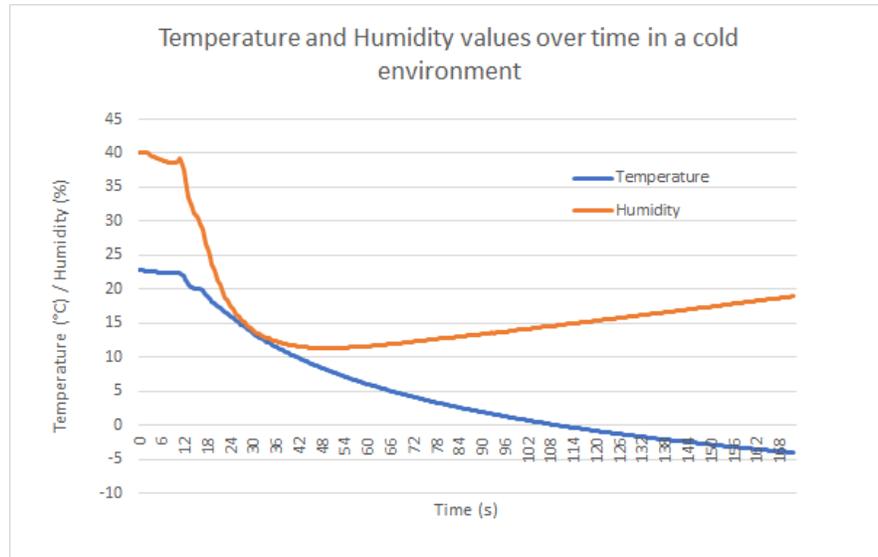


Figure 18. Graph showing the temperature and humidity values over time in a cold environment.

As shown in the figure above, the temperature and humidity values drop fairly quickly. However, we did notice that the temperature and humidity values got less accurate as the temperature got colder. This is probably due to the very cold temperatures messing with the components. Nonetheless, we did not worry too much since we knew that it would be highly unlikely for the temperature of the box to get that cold. Next, we put our sensor in a very hot and humid environment. We cared more about the result of this test since this climate would be more likely to happen with hot beverages or food. The results of the test are shown in the figure below.

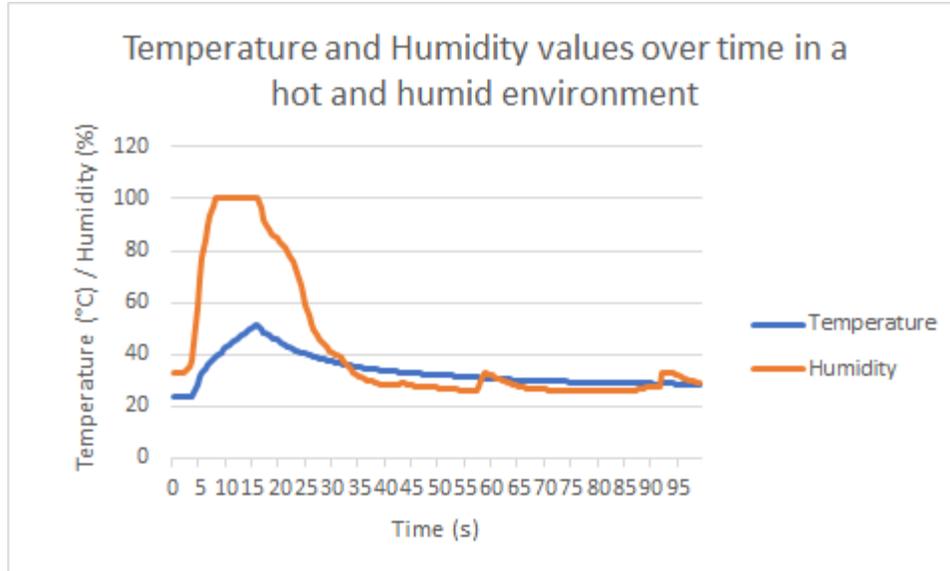


Figure 19. Graph showing the temperature and humidity values over time in a hot and humid environment.

As shown in the figure above, the values of temperature and humidity increased very quickly when we put our sensor in the hot and humid environment (stuffed it inside a hot pack). We also noticed that once we took out the sensor, it measured the normal values relatively fast, meaning that the sensor was able to detect sudden changes in temperature and humidity. Overall the sensor handled the extreme climates very well, giving us full confidence that it would be able to measure the temperature and humidity of any type of climate that would be in the food box.

6.2.3 Waterproof Fabric Tampering Tests

The next aspect we wanted to test was if our waterproof fabric would tamper with the temperature and humidity readings of the sensor. To conduct this test, we simply measured the temperature and humidity readings of a normal climate (room temperature climate) with and without the fabric covering the sensor and sensor casing. In this experiment, we were really looking to see if the fabric made would retain any heat or moisture making the values increase. The results are shown in the figures below.

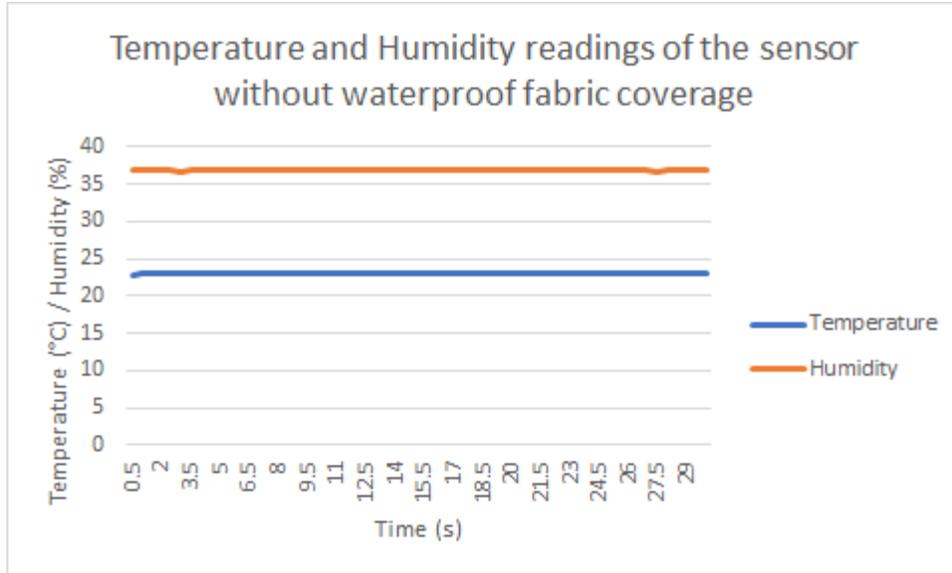


Figure 20. Graph showing the temperature and humidity readings over time of the sensor without waterproof fabric coverage.

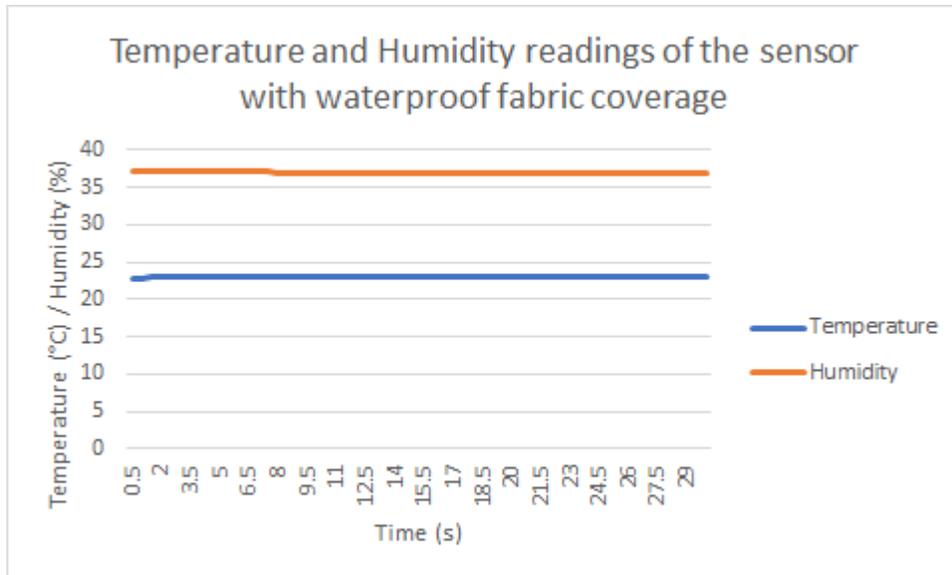


Figure 21. Graph showing the temperature and humidity readings over time of the sensor with waterproof fabric coverage.

As shown in the figures above, the temperature humidity readings are almost the exact same over time. Based on our test, the waterproof fabric had no impact on the sensor readings. This makes

sense since the fabric is very thin meaning that it can't retain heat or moisture very well, but because of its polyethylene coating, it is able to repel water.

6.2.4 Waterproofing Tests

To make sure the fabric we used is actually waterproof, we poured water on top of it and we observed if the water seeped through it. To do this, we tapped a piece of the fabric on top of a clear bowl and then poured water on the fabric. We then let the water sit on the piece of fabric for a long time (around an hour) and we checked if any water collected in the bowl. Our result is shown in the figures below.

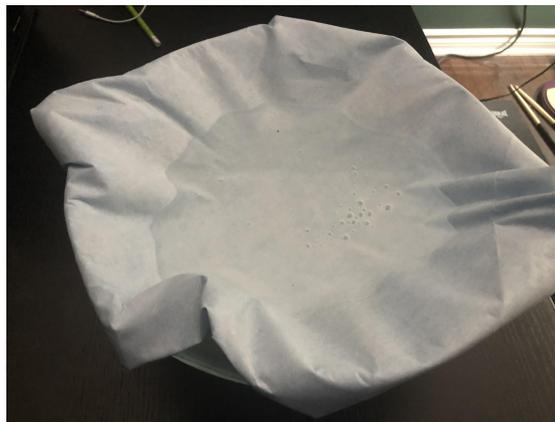


Figure 22. Water poured on top of the waterproof fabric.



Figure 23. Empty bowl after letting the water sit on the waterproof fabric for a long period of time.

As shown in the figures above, there was no water that seeped or leaked through the waterproof fabric. This is due to its polyethylene coating that repels water. Overall, our team had full confidence in this fabric to protect our components from water damage.

7 Conclusions and Recommendations for Future Work

Over the course of developing the Climate Sensor, multiple prototypes have been made and many lessons have been learned. To develop the climate sensor, we used the design process to determine client needs, define the problem, benchmark, ideate, prototype and test. We discovered that 3D printing is not always 100% accurate, especially when the details are small. Furthermore, clients change their minds about their wants and needs so it is important to be prepared for that over the course of the design process. Furthermore, we learned that it is important to consider other possible scenarios, ask “What if?”, because we found that sometimes those other scenarios happen when it is least expected and it is critical to be prepared.

The most productive avenues for future work are the reaction of the sensor to new environments and a better way to attach the waterproofing fabric to the sensor. If we had more time to work on the project, we would attempt making the waterproofing covering a permanent attachment to the sensor casing. We abandoned sending colour coded data to the flight operator due to lack of time which would have made the data a lot easier to understand.

8 Bibliography

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable B: Need Identification and Problem Statement* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable C: Design Criteria and Target Specifications* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable D: Conceptual Design* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable E: Project Schedule & Cost* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable F: Prototype I and Customer Feedback* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable G: Prototype II and Customer Feedback* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable H: Prototype III and Customer Feedback* [Published privately]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable I: Design Day Presentation* [Unpublished presentation]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *Project Deliverable J: Project Presentation* [Unpublished presentation]. GNG 1103 University of Ottawa.

Ahunanya, O., Carnegie, O., Jones, M., Léger, Z., & Midence Rosario, J. (2021). *B.O.M. (Bill of Materials)* [Unpublished presentation]. GNG 1103 University of Ottawa.

MakerBot. (2015). *Arduino Uno Snug Case by Esquilo*. Thingiverse.
<https://www.thingiverse.com/thing:994827>.

TutorialsPoint. (n.d.). *Arduino - Board Description*. Tutorialspoint.
https://www.tutorialspoint.com/arduino/arduino_board_description.htm.

Zuo, B. (2020). *Grove - AHT20 I2C Industrial Grade Temperature & Humidity sensor*.
<https://wiki.seeedstudio.com/Grove-AHT20-I2C-Industrial-Grade-Temperature%26Humidity-Sensor/>.

(Ahunanya et al. 2021)

APPENDICES

9 APPENDIX I: Design Files

Summarize the relationship of this document to other relevant documents. Provide identifying information for all documents used to arrive at and/or referenced within this document (e.g., related and/or companion documents, prerequisite documents, relevant technical documentation, etc.).

Include all design files in MakerRepo. Also provide the MakerRepo link to your project.

Table 3. Referenced Documents

| Document Name | Document Location and/or URL | Issuance Date |
|---------------|------------------------------|---------------|
| | | |
| | | |
| | | |
| | | |
| | | |

10 APPENDIX II: Other Appendices

You can include other critical and important work here. Maybe they are not important in the structure of this document but need to be included.