JAMZ Temperature Sensor: Conceptual Design

Ogechi Ahunanya, Olivia Carnegie, Michaela Jones, Zach Léger, Julio Midence

February 21, 2021

Abstract

The team is working on a climate sensor for an automated drone delivery system. Each team member has individually generated one subsystem for the climate sensor with three functional solutions. The concepts each team member generated were discussed and combined into three functional solutions for the climate sensor.

Table of Contents

1 Introduction

Our group has decided to tackle the challenge of creating a climate sensor for the packaging of the automated drone delivery service offered by JAMZ. There are three main areas we as a team have decided to divide up the add on system of the climate sensor. The first one would be the materials needed to implement our climate sensor. This includes a microcontroller and its other electronic components, and the materials needed for any protection. The second one would be the placing and fitting of the microcontroller and the temperature and humidity sensor. The third one is all about the non visual aspect of the system. In other words, the programming side.

2 Protection of Sensor

2.1 Covering and Materials

For the temperature sensor cover, we thought of designs and materials that would properly cover and protect the sensor itself within the food delivery box. For all designs the wires will connect to the sensor and run through the bottom of the case to then connect to the arduino. The wires will also run through a tube for extra protection. Designs were made as either a permanent attachment to a reusable delivery box or as an easily removable attachment to a single use delivery box.

2.1.1 Covering Concept #1

As the first concept, we assume that the sensor will be a permanent attachment to the delivery box and no clients will need to worry about detaching it. The bottom of the sensor will slip into a silicone case that will leave the top portion of the sensor completely visible.

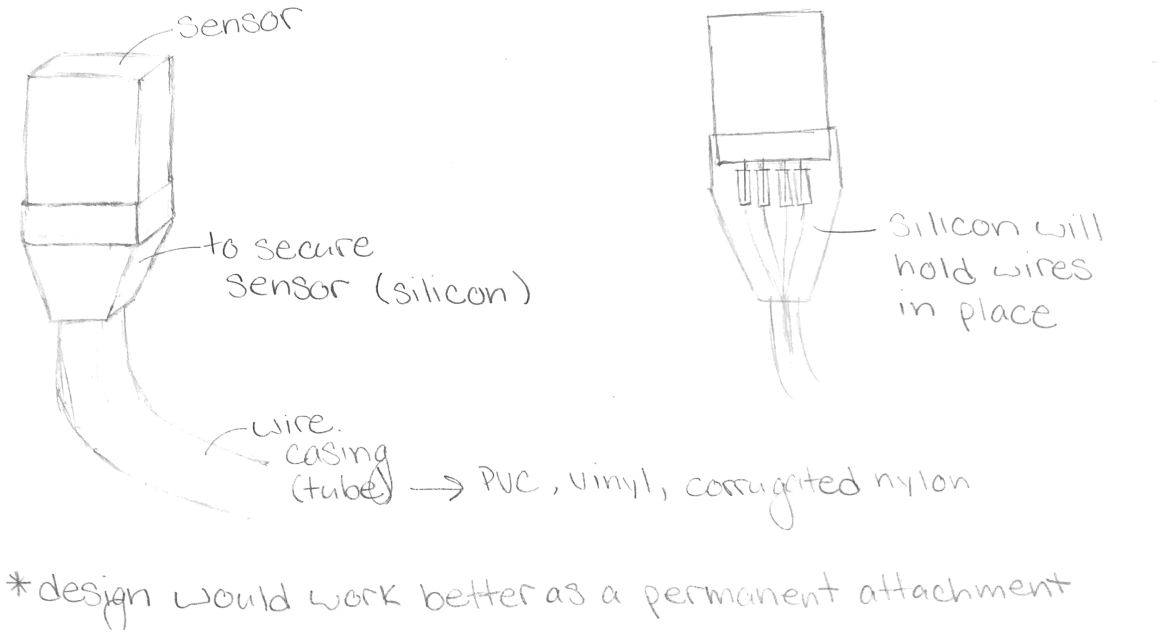


Figure 1. Sketch of a permanent sensor attachment to the delivery box.

2.1.2 Covering Concept #2

As the second concept, we assume that the sensor will be detachable from the delivery box. This case is made of a more durable hard plastic and the sensor will be held in the case by screws.

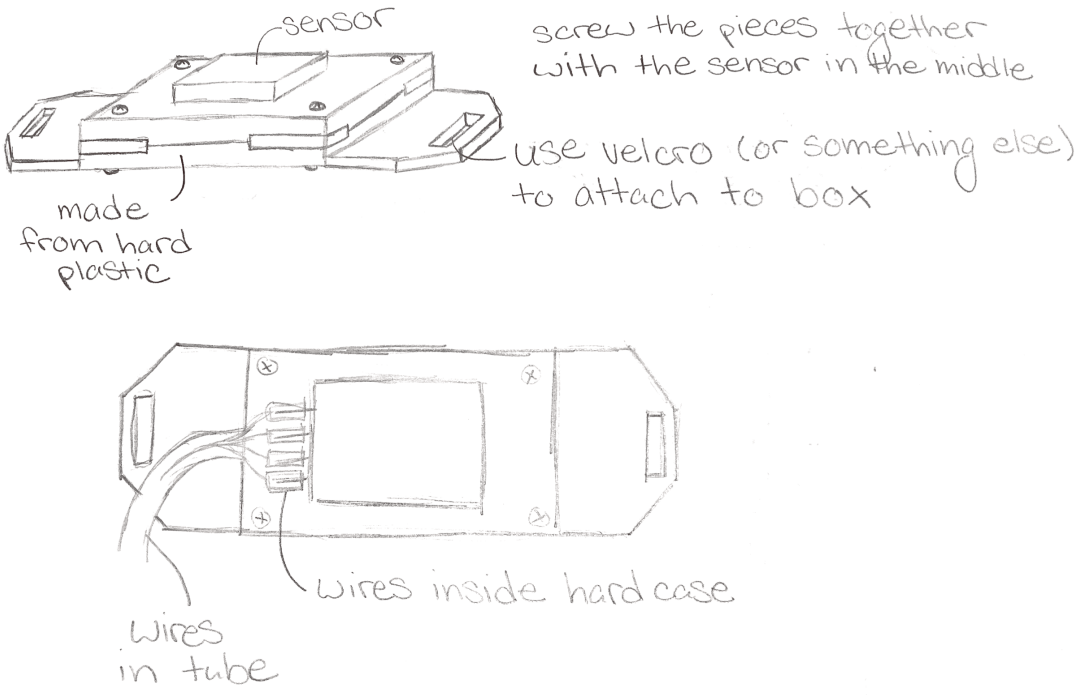


Figure 2. Sketch of a detachable sensor attachment to the delivery box.

2.1.3 Covering Concept #3

As the third concept, we assume that the sensor will be detachable from the delivery box. The sensor will be able to slip into the flexible silicone case that will then be attached to the delivery box.

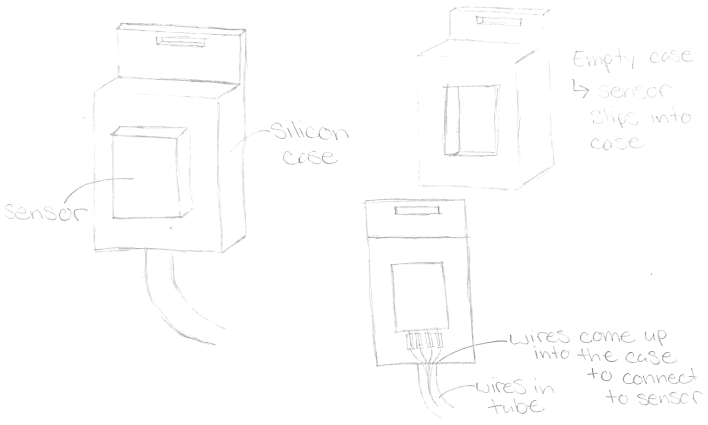


Figure 3. Sketch of a detachable sensor attachment to the delivery box.

2.1.4 Covering Concept #4

As the fourth concept, the sensor could be detachable or a permanent attachment to the delivery box. The sensor will fit into the bottom of the hard plastic case and the top can then come down and snap into place to keep it protected.

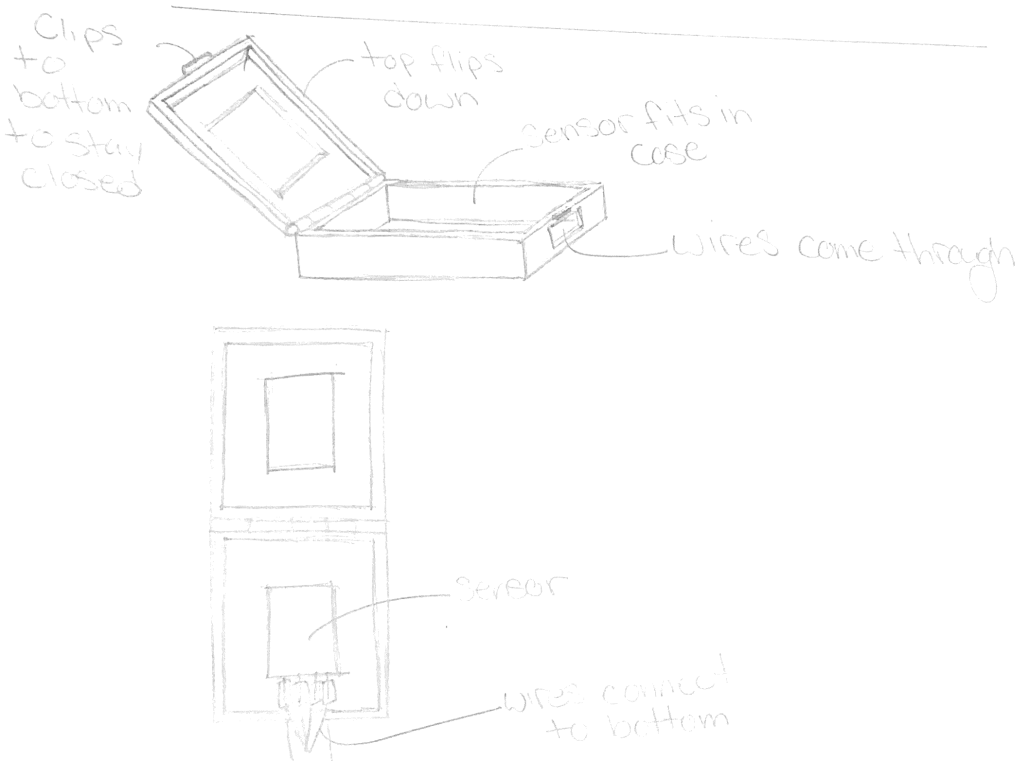


Figure 4. Sketch of a detachable and/or permanent sensor attachment to the delivery box.

2.2 Final Protection of Sensor Concept

For the final sensor covering design concept, we think that a combination of Covering Concept #2 and Covering Concept #4 would be the ideal solution. The ability to work as a detachable or permanent attachment provides flexibility in the delivery box design and the hard plastic case will provide greater and longer lasting protection to the sensor.

3 Fitting and placement of sensor

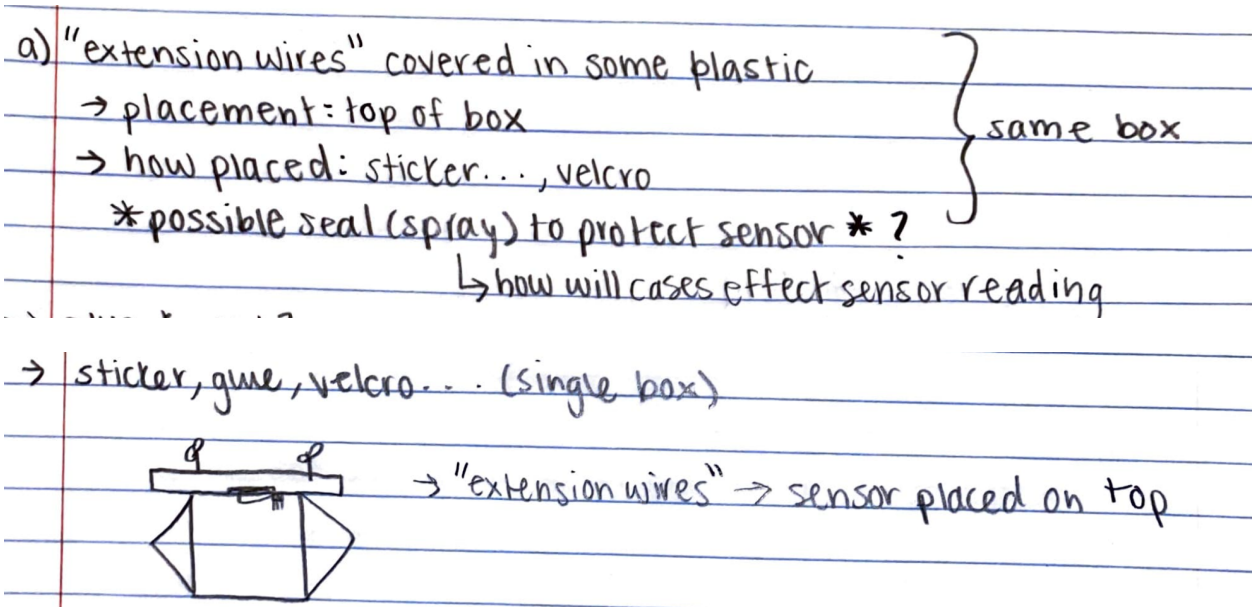
3.1 Fitting of sensor into box

For ways to fit the temperature sensor into the food delivery box, we considered two possible scenarios. The first scenario would be if JAMZ decided to have a box attachment to their drone, meaning that the same box would be used for each food delivery and no box detachment would

be required. The second option would be if JAMZ decided to have no same box attachment to their drone, meaning that each food delivery would attach their own box.

3.1.1 Sensor Fitting Concept #1

As a first concept, we decided to assume that a single box would be attached to the drone. From that we thought that the sensor could be extended from the Arduino to the box via "extension" type wires through a small hole. The placement of the sensor could be on the top of the box, stuck by some sort of adhesive or velcro (for easier detachment).



Figures 5 & 6. Rough work of concept #1: Extension Wire + sticker/velcro

3.1.2 Sensor Fitting Concept #2

As a second concept, we decided to consider both scenarios… From that we decided to run off similarly to our first concept but this time make it a bit more user friendly. In this concept we would still want the temperature sensor to run out of the Arduino via extension type cords, but this time have them combine into one big cord due to a protective plastic. From there we decided the end of the cord could be viewed as a plug (with the sensor inside). This idea would require each food vendor to plug in the drone before take off. This concept could be used in either of the two scenarios.

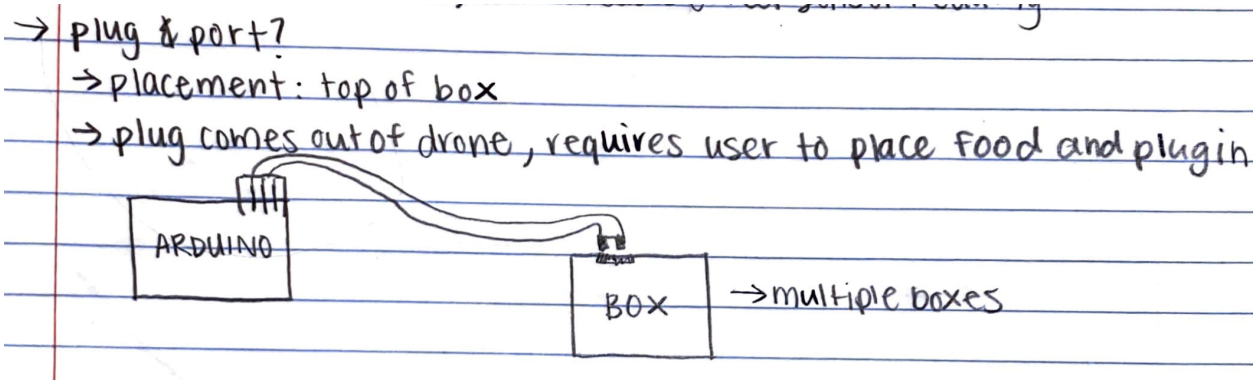


Figure 7. Rough work of concept #2: Plug & Port

3.1.3 Sensor Fitting Concept #3

As a final concept, we decided to again consider both possible scenarios. In this concept each box would have to be designed with a hole for a sensor. The sensor would have to be placed accordingly to fit in the box when it is placed on the drone. This would act as a puzzle piece type placement and could be used in either scenario. To make it more user friendly it might be useful to have a clicking sound when the box is properly placed, similar to when you put the lid of a tupperware one.

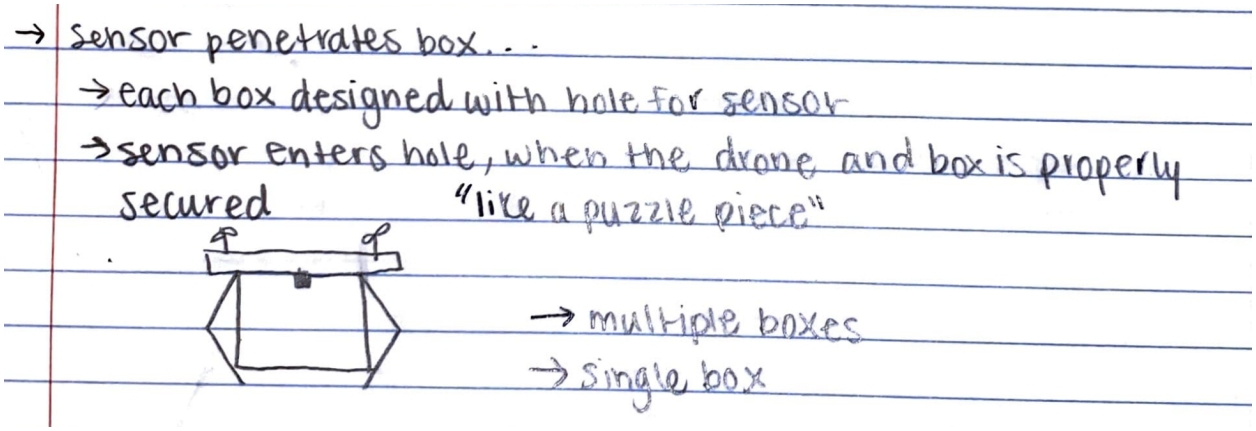


Figure 8. Rough work of concept #3: Sensor Penetrates

3.2 Attachment of the Add-On to the Drone

The attachment of the add-on system is constrained by the aerodynamics of the drone and its ability to fly well in the air. Taking this into account, it would be best to attach the add-on to the sides of the drone since it is lightweight and compact. To ensure the add-on stays secured to the drone in flight, a strong adhesive such epoxy (an example being Gorilla Glue) to the casing of the add-on where the add-on would be placed. Epoxy is resistant to high temperatures, solvents, UV light and impact which are qualities needed for something attached to a drone in flight. Super Glue can also be used to attach the add-on to the drone however it has a very low shear

stress value meaning that it will not resist off angle stress which may be present during flight and operation.

3.2.1 Attachment Concept #1

The first attachment concept is simply center placing it on top of the drone which allows for the drone to be more balanced in flight. This surface seems to be unobstructed and free for placement of an add-on which is ideal since we do not want to disturb any of the other features of the drone.

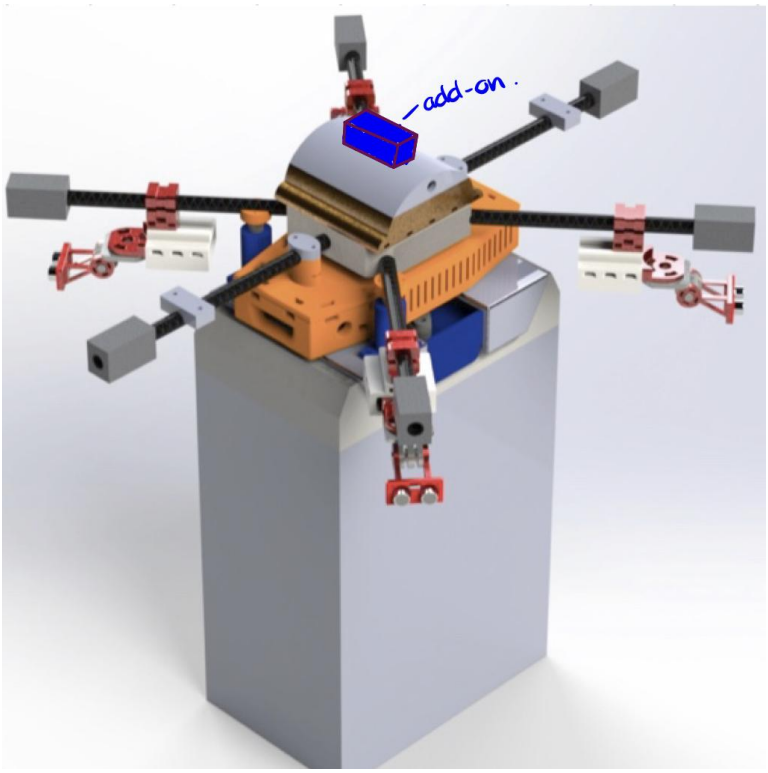


Figure 9. Sketch of the add-on attached to the center/top of the drone.

3.2.2 Attachment Concept #2

The second attachment concept entails adhering the add-on to the upper side of the drone. The surface is flat which is ideal for attachment and is unobstructed which allows enough room for our add-on system. furthermore, the add-on is less exposed in the event of a fall, rendering it less likely to break due to fall impact.

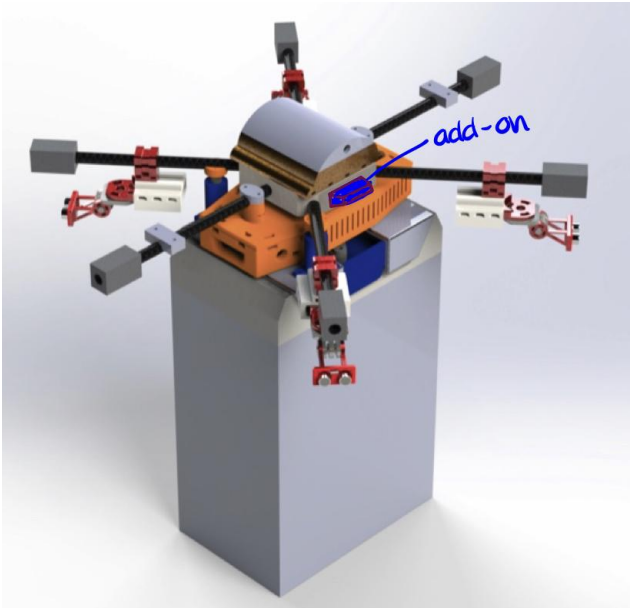


Figure 10. Sketch of the add-on attached to the upper side of the drone.

3.3 Final Fitting and Attachment Concept

For this portion of the design, we believe that Sensor Fitting Concept #3 and Attachment Concept #2 would be best. We think that Sensor Fitting Concept #3 would be the most user friendly and most applicable to either or the two possible scenarios. We think that Attachment Concept #2 since the surface is flat which makes it optimal to adhere to the add-on since it also has a flat surface. Furthermore, the add on would be less exposed to damage compared to Attachment Concept #1 since there are other large components surrounding it.

4 Programming & Electronics

The code being written heavily depends on the electronics being used. Our team has chosen to work with the arduino uno and the DHT22 climate sensor. The DHT22 is able to measure humidity and temperature. In order for the arduino to read the data from the sensor, a DHT library must be included in the Arduino IDE. Instead of having an unfinished code with uncertain functions, the team has decided to have a simple code which will read and print out the sensor's values. One of JAMZ's biggest priorities is to gather consistent data from the add on systems. Our team will give them consistent data of the temperature and humidity by keeping the code simple and doing various test runs.

4.1 Arduino Code

The simple code which will give consistent data has been finished. If the code is modified and begins outputting inconsistent data, the team will always have the code below to go back to:

```
/*
 * In order to read from the DHT22 sensor, you must include the DHT library
 */
#include <DHT.h>

/*
 * Some constants have to be declared. 1st, the digital pin the sensor is connected to.
 * Make sure to change this constant depending on the pin being used. In this code the pin
 * will be pin 8, but in theory any digital pin on the arduino works. 2nd, the type of DHT sensor
 */
 #define DHTPIN 8
 #define DHTTYPE DHT22
 DHT dht(DHTPIN, DHTTYPE);

 /*
  * Declaring variables needed. The variables will of type float for more accuracy.
  * The two variables are "humValue" for the humidity and "tempValue" for the temperature.
  */
  float tempValue;
  float humValue;

/*
 * The typical baud or bits per second rate used is 9600. This value will be used in the code
 */
void setup()
{
  Serial.begin(9600);
  dht.begin();
```

```
}


/*
 * The data from the sensor will be read and printed every 3 seconds or 3000 milliseconds.
 * The two variables have to be initialized with their according values
 */
void loop()
{
  delay(3000);
  tempValue = dht.readTemperature();
  humValue = dht.readHumidity();
  Serial.print("Temperature: ");
  Serial.print(tempValue);
  Serial.print(" °C \t Humidity: ");
  Serial.print(humValue);
  Serial.println(" %");
}
```

4.2 DHT22 Climate Sensor

There are a couple of key items regarding the functionality of the DHT22 sensor. The response time for the sensor is two seconds. Our team has taken it into consideration which is why the code only requires for the values to be read and printed every three seconds. For optimal performance with the arduino uno, the sensor needs to have a pull up resistor. In order to achieve this, our team has taken into account the use of a DHT22 sensor which is mounted on a PCB. The PCB has a built in pull up resistor and eliminates the third pin of the sensor that is never connected to the arduino.

4.3 Data Display Concepts

For data collected from the sensor, we designed ways to display this data on the operator's computer. The goal of these data display concepts is to display the collected data in a way that is clear, understandable and organized so that whoever views the data will understand it.

4.3.1 Data Display Concept #1

As our first concept, we came up with a graph that can show the relation between temperature and time or humidity and time. In the graph there will also be a preset range (defined by two dotted lines) so that the person viewing it will see when the temperature or humidity gets out of the wanted range. Also, the colour of the line would change depending if the temperature/humidity is in or out of range.

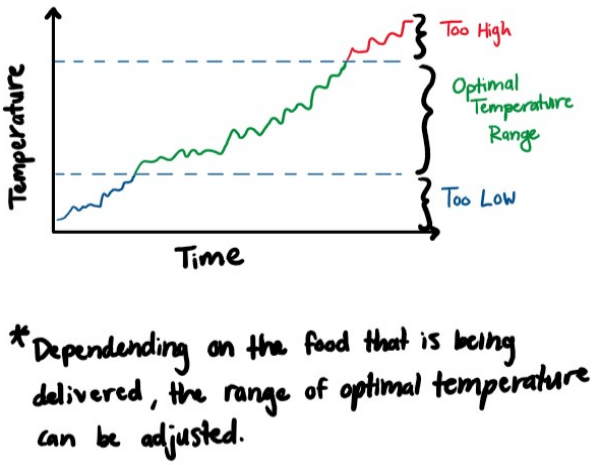


Figure 11. Sketch of Concept #1 for a temperature/time relation graph.

4.3.2 Data Display Concept #2

For our second concept, we came up with a continuous table from departure to arrival that will display the time, temperature and humidity as well as a status columns that indicate if the temperature and/or humidity is in range, getting out of range or out of range.

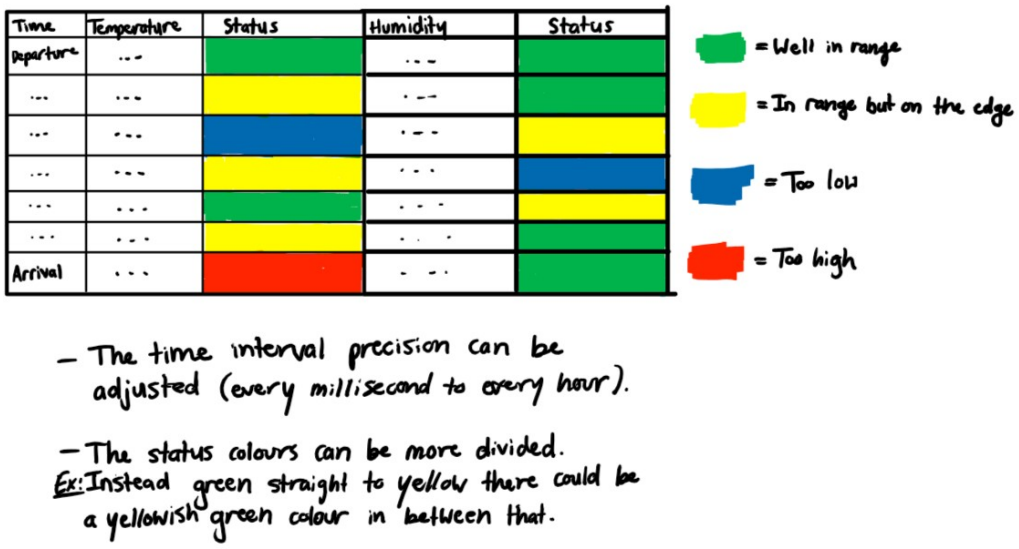


Figure 12. Sketch of Concept #2 for a simplified table.

4.3.3 Data Display Concept #3

For our third concept, we followed the same idea as Concept #2, but we modified the way that we display the status columns. One idea was to replace the solid colours by coloured lettering. So instead of a solid green colour, the table would say "In Range", in green lettering. The other idea was to take out the status columns and give the colour to the data. So if 52% is in range, it would be displayed in green.

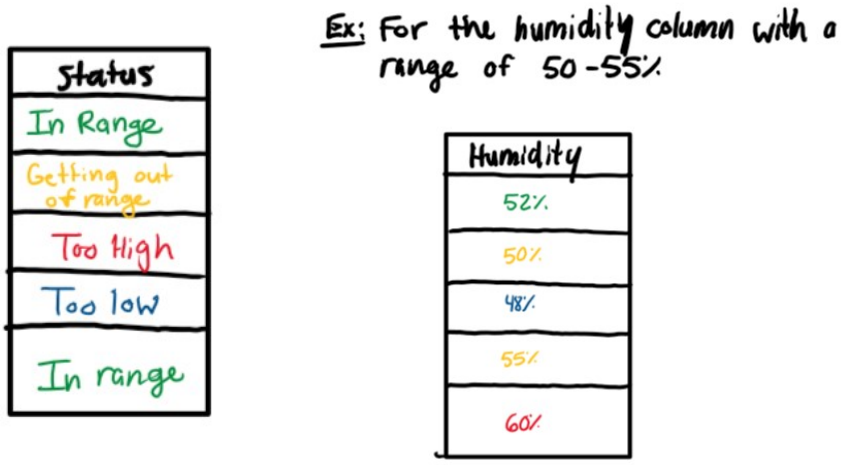


Figure 13. Sketches of Concept #3 for how the status of the data can be displayed.

4.4 Final Programming and Electronics solution

Based on the code and its limitations, the concept for the data display that would work best with the code is Concept #2 or some version of it. How the arduino IDE operates is by having a setup and forever looping function. In the setup function the code would print the titles of the columns and in the looping function, the code would print the results underneath the titles. The code would have if statements in order to bypass the colour coding of the results. The if statements would change the print message of the results if a certain value has been reached. This would work as a warning system.

5 Conclusions and Recommendations

In this report, our team has come up with various concepts and have combined them to form one main solution. The combination of concepts were selected so that each selected concept could work well together. For the final sensor covering design concept, we believe a combination of Covering Concept #2 and Covering Concept #4 would be the best. For the fitting and attachment of our sensor, our team decided that Sensor Fitting Concept #3, Attachment Concept #2 would work the best together. Finally, for the programming and electronics our team decided that with the coding concept we have, Data Display Concept #2 would be ideal to display the data.

6 References

1. *DHT11/DHT22 sensor with Arduino*. (2019, April 25). https://randomnerdtutorials.com/complete-guide-for-dht11dht22-humidity-and-temperature-sensor-with-arduino/#:~:text=Open%20your%20Arduino%20IDE%20and,Sensor%E2%80%9D%20in%20the%20search%20box