Weather Box

Tina Maurer, EE, Anthony Mendez, CSE, Stephan Kim, CSE, and Christian Norton, CSE

Abstract—**As part of pre-flight preparations, drone operators must check the local weather conditions to ensure a safe and successful flight. While commercial weather stations can effectively collect data for a specified area at the macroscale, weather conditions in that area at the microscale can vary greatly. Since flight conditions can be greatly affected by these constraints, drone operators need a more accurate localized weather map reading for the area of flight. Weather Box will create this localized map in a network of battery powered sensor modules to provide drone users with the required information via a website and application. Our product will allow operators to quickly decide whether the conditions are suitable for safe drone flight.**

I. Introduction

WEATHER forecasts for certain areas are not the most accurate since weather stations could be placed miles apart from each other. Interpolation between stations can be used to measure weather in areas between stations, but depending on the distance this method can be incredibly unreliable.

A. Significance

Weather has a massive role in many modern infrastructures and systems, either directly or indirectly. That is why it is so important to monitor and record weather data. Weather can affect a huge range of systems. Ranging from things like transportation scheduling, the operation of power systems, and even the ability for public services to enforce the law or give assistance. Every year, weather delays cost airliners and customers several billion dollars [1] [2]. Power systems can be shut down due to issues like icing or heavy wind [3]. These shutdowns can cost as anywhere between 30 billion dollars and 130 billion dollars annually [4]. Wind farms often run their turbines based off of wind and weather predictions to optimize their output to their costs of operation. Its shown that officers perform fewer stop and frisks during colder and wetter weather [5]. In many cases, when law enforcement and public services want to use drones in their operations, they cannot because the weather is too rough for the drone to operate safely.

Overall, weather can have a huge impact on the performance and function of many modern infrastructures and systems. In many cases, a far more accurate and small-scale rendering of weather can have a massive impact on the decisions and implementations of these systems. Airliners can have fewer delays as they can see exactly where an approaching storm is. Power grid developers can put more infrastructure in place to reduce the damage in locations they know will have harsher conditions. Officers, EMS, and firefighters can be better equipped to handle, or avoid, harsher weather conditions during operation and to see what their options are during an operation.

B. Context and Existing Products

There are a few products on the market that do help increase the resolution of weather forecasts. One of the most common is the weather balloon. Although weather balloons are very easy to deploy, they only give a single vertical line of measurements, can only be used once, have a chance of the payload never being recovered, and have limits on where they could be deployed. Another solution would be to have people on the ground with various sensors to take measurements themselves. Although this could be incredibly accurate, it is highly inefficient since it requires people to be present on site and record values at various points when that might not even be entirely necessary. Refer to *Appendix Section A* for more details.

C. Societal Impacts

One of our main areas of concern is the public safety aspect of drone flight. In some cases, it is much too dangerous to fly a drone outside due to weather conditions such as wind speeds. Since weather forecasts cannot give an accurate reading of the weather at a very small area, our system would help drone operators better judge if an area is safe to fly through. In addition, our system can help weather stations provide a more accurate forecast to specific areas.

D. Requirements Analysis and Specifications

We would like each Weather Box enclosure to be portable, able to accurately measure wind speed, temperature, humidity, atmospheric pressure, air quality, and dust, and send data remotely.

Requirement	Specification	Value
Portable	weight	< 1lb
	Battery powered	24hr battery life
Accurate measurements	Wind speed	Accurate with 95% confidence
	temperature	
	humidity	
	Atmospheric pressure	
	Air quality	
	dust	
Send data remotely	Connect to Wi-Fi and send data to website	2.4 GHz band to connect to Wi-Fi

Table 1: Requirements and Specifications

II. Design

A. Overview

For our solution we will have 3 independent, battery-powered enclosures that will send weather data via Wi-Fi to our website. Below is the block diagram of our proposed solution.

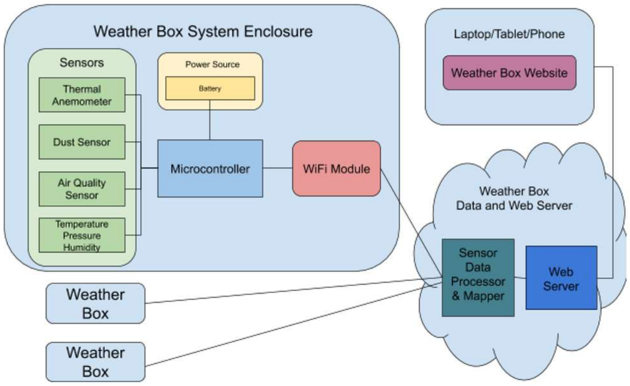


Figure 2: Block Diagram of our Solution

B. Power Source

One of the specifications of our network of weather sensor packages is a battery life of at least 24 hours. In order to meet this specification, we are using a +12V battery stack of AA batteries. This creates the voltage line necessary for the thermal anemometer, as it requires a supply voltage of 9 to 12V. By creating the battery stack, we allow ourselves a buffer for the batteries in case they drop below the ideal stack voltage. From the +12V stack, we use a LDO, or low dropout regulator, which is a linear regulator that steps the +12V down to create a +5V line. We then use a similar LDO to again step down the voltage and create a +3.3V line. With this power circuitry, we are able to create all the power lines necessary to run all the electronics of the sensor package. One consideration that we have for the upcoming development of a printed circuit board is changing the regulation circuitry from linear voltage regulators to switching voltage regulators. Though the linear regulators we are currently using can handle up to 1.5A and allow us to meet the current power specification (as we are drawing approximately 110mA and have a battery life of about 28 hours), a switching regulator will allow us to further exceed our specification in this area and optimize our sensor packages.

C. Microcontroller

We are using an STM32 microcontroller in which the firmware is written in C using ST Microelectronics' own HAL API. For this block, we needed to learn the HAL API as well as the locations and functions of different pins on our microcontroller. This part was like Computer Systems Lab, where we had to read datasheets on our microcontroller and figure out the different quirks of our specific one. Our microcontroller takes the data from the sensors utilizing multiple protocols for serial communication, and then sends the data via the Wi-Fi module to our website. Before the data is sent, the microcontroller does some processing on the data to make the data more meaningful, and then formats the data into the JSON format for our website to read. We tested our firmware using STM's own CubeIDE which is an Eclipse-based IDE and used the debugger to check values coming in from the sensors.

D. Sensors

We have four different sensors connected to our microcontroller. Much like the microcontroller, the techniques used to create this block are very similar to what we did in Computer Systems Lab, where we would read the various datasheets of each sensor and figure out how to get our microcontroller to interact with the different sensors. For wind speed, we are using a Modern Device Thermal Anemometer which uses two thermistors and compares how much power it takes for the exposed thermistor to reach the same temperature as the idle one and gives an output voltage based off of that. The output voltage is then input into a formula in order to get wind speed. For MDR, this sensor's output voltage is read via an external ADC that utilizes the SPI protocol. For MDR, we are also using a Bosch BME280, which is a 3-in-1 temperature, pressure, and humidity sensor. This sensor utilizes the I2C protocol. Our other two sensors are the SHARP GP2Y1010AU0F dust sensor and the ams CCS811B-JOPD500 air quality sensor. These two will be implemented for CDR. For testing, we will set up different environments to simulate changes in weather and compare measurements from our system to other sensors in the same conditions. For more on testing, refer to *Appendix Section B.*

E. Wi-Fi Module

For enabling Wi-Fi connectivity, we are using a development board from SparkFun which utilizes, Esspressif Systems' ESP8266 chip for Wi-Fi connectivity. For this, we had to learn how AT commands for modems worked and send them to the Wi-Fi module via UART from the microcontroller. In addition, we had to learn how to format an HTTP POST request in order to send data to our website. In order to implement this block, we had to utilize knowledge from our Computer Networks course for the HTTP messages as well as techniques from our various programming courses in learning the commands to program the module. For testing this, we sent some data to our website and checked the website's logs to see if the request would go through.

F. Weather Box Data, Web Server, and Website

An integral part to our project is the Web Server, which hosts the communications with the SQL database, Weather Box systems, and displaying the website. For our hosting provider, we are using Microsoft Azure. Microsoft Azure provides $100 credit for students, along with some free resources. The web server is programmed in Python, utilizing the Flask library to perform web server functionalities. The website shows a map of the Weather Boxes on the map using the Google Maps API in JavaScript. The Maps API has marker objects and information window objects. We place the markers where the Weather Boxes are, and the information windows attached to the markers. The way the website retrieves information for each system is as follows.

The JavaScript running on the website will ping the server every 10 seconds with a HTTP GET data request. The Python Flask web server would receive this request and retrieve the most recent Weather Data for each system. The server then programs this data into a JSON format that could be easily understood by the JavaScript. The JavaScript client receives the JSON message, parses it for the weather data, formats the printout, and displays it on the information window for each system. For retrieving the information from each Weather Box, it is as follows.

The Weather Boxes are currently programmed to measure and send data every 10 seconds. This can be adjusted later. The Weather Boxes pings the server for a HTTP POST data request. The server acknowledges, the Weather Boxes microcontrollers. The microcontroller then measures the data from each of its sensors, and formats it into a JSON message. The JSON message is then sent to the server. The web server parses the message and stores it into the SQL database with a timestamp.

III. Project Management

Status of MDR Goals	
Status	Goal
We were able to build two fixed point development sensor boards that measure temperature, air pressure, humidity, and wind speed.	Two fixed point development sensor boards that measure temperature, air pressure, humidity, and wind speed.
We were able to measure temperature below 30°F but have not been able to test in an environment greater the 80°F mark.	Temperature between 30°F - 80°F with 95% confidence
We have been able to measure air pressure but not measure below or above these marks.	Air pressure between 1.00atm - 0.96atm with 95% confidence
We have been able to measure humidity but not at these marks we set.	Humidity between 0% - 100% with 95% confidence
We have been able to measure wind speed but not 50 mph mark.	Wind speed between 0mph - 50mph with 95% confidence
The sensor package is able to be powered by a battery, the microcontroller is still powered by the laptop.	The sensor package will be powered by a battery
The website displays a map with each sensor system, and displays the data points of each system.	Map displaying the location and data points of each sensor package

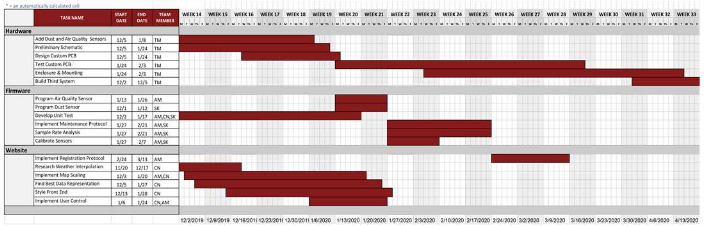
Table 3: Status of MDR Goals we promised in PDR.

We were able to accomplish all our MDR deliverables. We built two sensor packages that measure our data and communicates with the web server. The web server saves this data and sends it to the website. The website creates a map of each data point. In terms of MDR, we have met all our goals and are done. For CDR, we still have more to be do.

We need to begin researching an interpolation algorithm, a process to registering each system and changing their stored positions. There also needs to be more user control and better data representation on the front end.

On the firmware side, we need to program the air quality sensor, dust sensor, design a maintenance protocol, and perform more tests and calibrations for each sensor. Ideally, we would program the sensors early, so we have time to calibrate and test them.

For the hardware development, a preliminary schematic is currently being drawn up for the MDR prototype of the sensor package. The power circuitry will then be refined to optimize the battery life of the weather box. From there, we will need to complete layout and a first revision of the custom PCB, with circuitry to flash the MCU as it will no longer be on a Nucleo development board. The completed first revision PCB will be done before CDR, so that testing and possible debugging can be done with enough time to create a second revision PCB if necessary.


* = an automatically calculated cell

	TASK NAME	START DATE	END DATE	TEAM MEMBER
Hardware				
	Add Dust and Air Quality Sensors	12/5	1/8	TM
	Preliminary Schematic	12/5	1/24	TM
	Design Custom PCB	12/5	1/24	TM
	Test Custom PCB	1/24	2/3	TM
	Enclosure & Mounting	1/24	2/3	TM
	Build Third System	12/2	12/5	TM
Firmware				
	Program Air Quality Sensor	1/13	1/26	AM
	Program Dust Sensor	12/1	1/12	SK
	Develop Unit Test	12/2	1/17	AM,CN,SK
	Implement Maintenance Protocol	1/27	2/21	AM,SK
	Sample Rate Analysis	1/27	2/21	AM,SK
	Calibrate Sensors	1/27	2/7	AM,SK
Website				
	Implement Registration Protocol	2/24	3/13	AM
	Research Weather Interpolation	11/20	12/17	CN
	Implement Map Scaling	12/3	1/20	AM,CN
	Find Best Data Representation	12/5	1/27	CN
	Style Front End	12/13	1/28	CN
	Implement User Control	1/6	1/24	CN,AM

On the hardware side, Tina is the lead. Anthony is a general software lead. Stephan oversees the firmware, and Christian is responsible for the website and researching an interpolation algorithm. Tina will be the Team Manager until CDR. Anthony, Christian, and Stephan will be assisting each other if need be for general software issues. Tina is solely responsible for the hardware and power systems on each board, as well as designing the enclosures for each system.

IV. Conclusion

Currently, we have met our MDR goals and are working on getting our prototype closer to an actual product. We had to work many more hours than expected in order to get certain components working, especially since we made our MDR prototype with our STM32 microcontroller instead of an Arduino or an Arduino-like board. However, we plan to have a prototype of our custom PCB, enclosure for the PCB and sensors, all the firmware completed, and an interpolation algorithm running on our server for CDR. On the web server side, we plan to have our first iteration of a weather interpolation algorithm up and running. We are also planning to implement a registration and relocation protocol for a Weather Box. So, each time we add a new Weather Box system or need to change its location, we do not have to write SQL queries manually and can do it all from the website. And finally, we will test the systems more often and attempt to measure each data point with high accuracy.

Acknowledgment

We would like to thank Professor Zink for providing us with advice and assistance as our advisor throughout the semester. We also like to thank our evaluators, Professor Tessier, and Professor Pishro-Nik for providing feedback and advice to us after our presentations. We would also like to extend our thanks to the M5 staff allowing us to use their tools and components.

References

[1] J. Erdman, “The Most Weather-Delayed Major Airports in the U.S.,” *The Weather Channel*, 21-Nov-2018. [Online]. Available: https://weather.com/travel/news/2018-11-19-most-weather-delayed-major-us-airports. [Accessed: 18-Dec-2019].

[2] T. A. Press, “Flight delays are costing airlines serious money,” *Mashable*, 10-Dec-2014. [Online]. Available: https://mashable.com/2014/12/10/cost-of-delayed-flights/. [Accessed: 18-Dec-2019].

[3] Y. Liu, “Power Grid Operation Weather Security: NCAR Research Applications Laboratory,” *RAL*, 2019. [Online]. Available: https://ral.ucar.edu/projects/power-grid-operation-weather-security#nogo. [Accessed: 18-Dec-2019].

[4] K. H. Lacommare and J. H. Eto, “Understanding the cost of power interruptions to U.S. electricity consumers,” *ERNEST ORLANDO LAWRENCE BERKELEY NATIONAL LABORATORY*, Jan. 2004.

[5] M. P. J. Ashby and L. Tompson, “Does a good cop really never get wet? The impact of weather on stop and frisk,” Jun. 2018.

Appendix

A. Design Alternatives

When looking into the subject of monitoring weather at a micro-scale, there is a range of products that satisfy some of the individual aspects that one would be looking for, like portability, scale, resolution, or data storage and analysis. However, many of these products do not cover all the aspects that someone would want out of them. Our design process focused on trying to deliver a product that had as many of these aspects as possible within our price range.

One of the early designs that we had conceived used LIDAR (Light Detection and Ranging) to measure the rate that particles in the air moved in order to create a map of wind speed, rainfall and dust density. This would have been used in conjunction with other sensors in order to generate a full map of weather conditions. However, this design came with many draw backs. LIDAR systems are not cheap and to get a basic unit that had the resolution needed for such calculations was out of our budget. In addition to this, systems that already implement LIDAR for similar applications, like those used in windfarms for measuring wind speed, often come in very large packages, usually needing to be towed by a truck. These two major restraints were the reason this design was ultimately scrapped. Unfortunately, because of this, it was clear that having multiple three-dimensional maps of wind was out of our price range.

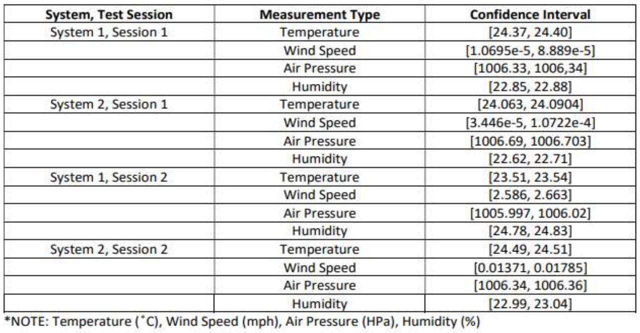
We had looked at many existing sensor packages at different price ranges and noticed that although they had many of the sensors that we were looking for, they lacked the ability to access the data from alternative locations. This was the main criteria we wanted to have implemented in our project. Because of this many of our design choices came down to budgeting our sensor packages with our power systems and communication systems. This is what ultimately aided us in making the major design choices in our project. Some sensors were dropped in favor of having higher quality sensors and communication systems. It was this process of optimization that lead to our final design.

For the most part, our design process was swapping out almost equivalent sensors while also trying to optimize the cost of operation with of the sensors with the necessary power systems to run them. For example, many of the sensors that we had dropped or swapped were due to the fact that they required far too much power and would not be able to have as long of a battery life as we would want. In other cases, sensors or parts were changed due to the cost of them being to much and putting us slightly out of the given range we wanted for each sensor. This happened a few times while we were deciding the best Wi-Fi module to use for our project.

B. Testing Methods

Our current testing methods consist of starting a Wi-Fi hotspot on a laptop or phone. The Weather Boxes would connect to this Wi-Fi hotspot, and to send their data every 10 seconds. One this setup is done, if we want to test the wind speed, we grabbed a fan and let it run over the wind sensor for about 20 minutes. After, we turn of the fan and perform another 20-minute session with no wind. To test the temperature sensor and overall operating temperature, we went outside for about an hour where it was about 5 degrees Celsius. The Weather Box performed well and did not suffer from any issues.

In order to analyze our results from the various testing sessions, we calculated confidence intervals of 95% for each of the testing sessions. This displayed the accuracy of our sensor packages for each session. We completed both a few shorter tests (around 20 minutes) and a longer test to show that we met the specification for our battery life. This shows that our project can both accurately measure weather data as well as be able to meet our power/battery specification. Below is the test data we collected for each test session, shown in Table 4. Furthermore, in Figure 3, we show the endurance testing of one of the weather boxes. During this task, one of the sensor packages was able to take measurements every 10 seconds for approximately 35.5 hours.



System, Test Session	Measurement Type	Confidence Interval
System 1, Session 1	Temperature	[24.37, 24.40]
	Wind Speed	[1.0695e-5, 8.889e-5]
	Air Pressure	[1006.33, 1006,34]
	Humidity	[22.85, 22.88]
System 2, Session 1	Temperature	[24.063, 24.0904]
	Wind Speed	[3.446e-5, 1.0722e-4]
	Air Pressure	[1006.69, 1006.703]
	Humidity	[22.62, 22.71]
System 1, Session 2	Temperature	[23.51, 23.54]
	Wind Speed	[2.586, 2.663]
	Air Pressure	[1005.997, 1006.02]
	Humidity	[24.78, 24.83]
System 2, Session 2	Temperature	[24.49, 24.51]
	Wind Speed	[0.01371, 0.01785]
	Air Pressure	[1006.34, 1006.36]
	Humidity	[22.99, 23.04]

*NOTE: Temperature (˚C), Wind Speed (mph), Air Pressure (HPa), Humidity (%)

Table 4: 95% Confidence Intervals

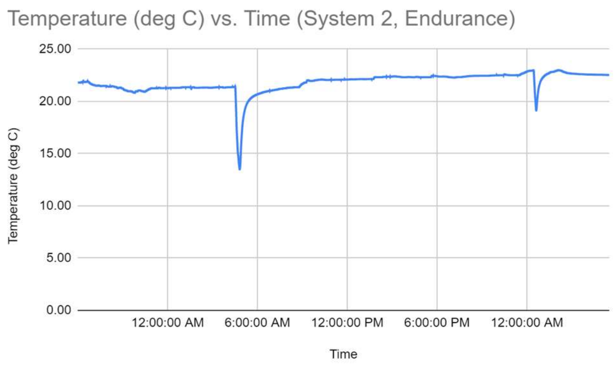


Table 5: Endurance Testing for System 2

C. Team Organization

Anthony is the Software Lead since he has the most work experience in Software Engineering. Tina is the Hardware Lead since she is the only Electrical Engineer in the group, but also because she has research and work experience under her belt. Stephan oversees most the firmware but receives assistance from Anthony in programming some modules. Christian oversees the webserver and website but receives assistance from Anthony in programming as well.

Overall, the team is working well together. We have met our MDR deadlines, we have two working systems, that measure data somewhat accurately.

Relative to this project, each team member has their own expertise. Anthony has a good understanding of the overall system and how they should communicate with each other. Tina expertise is in anything hardware related: power systems, components, PCB design, and enclosure design. Stephan has in charge of most of the firmware and programmed the Wi-Fi module and the wind sensor with the ADC. Christian has programmed parts of the webserver and website.

While everyone has their own respective domains, we always assist one another when one is falling behind on their work. This semester, Christian had a surgery and was unavailable for about a month. So, we had to fill in for him and complete some of his website and webserver responsibilities.

We have a good communication system set up right now.

D. Beyond the Classroom

A. Anthony

Part of my responsibility was setting up a few crucial parts of this project. The first was setting up a Microsoft Azure project to host our website. I used Azure before at an internship but never dealt with creation of projects or resources, which is something I had to learn. Not only that, but I also had to learn how to setup the Python Flask website, the Google Cloud account to access the Google Maps JavaScript API and set up the STM32 project in CubeIDE.

In connection with my current experience as a professional, there are a few domains. In setting up the Microsoft Azure project, I used Azure in my time with Citrix as a Software Engineering Intern, so my experience there helped me with this project. I have experience programming Microcontrollers from class and my time with Zebra Technologies.

B. Stephan

My main responsibility for this project is the firmware. So far, I have written the firmware for our ADC, anemometer, and our Wi-Fi module. This project has allowed me to further develop my skills with embedded systems and C, which I first developed in our Computer Systems Engineering class. I also had to learn the AT Command set for modems in order to program our Wi-Fi module. The main skill that I have learned is to create code that does not exist elsewhere since I could not find any other project that utilizes the exact same hardware configuration. For this project, I essentially had to write my own driver to program our Wi-Fi module via our microcontroller by serially sending commands specific to our project. Some helpful resources have been the different datasheets as well as looking at some examples utilizing similar hardware as us.

I see connections with this project and my potential future career since this project has exercised my skills in both writing firmware and debugging it. By writing code that doesn't exist, I have exercised my skills not only as a programmer, but also as an engineer by creating something that does not exist.

C. Tina

As the hardware lead of this project, I have needed to both make all major hardware decisions for the group and begin designing our custom PCB. Thus far, this project has allowed me to further develop my board-level design skills as well as learn how to take into consideration both software and mechanical constraints when making hardware decisions. An example of this is my design of the power circuitry and decisions on using an external ADC component. With the power circuitry, I had to create and test specifications such as weight and capacity of the batteries, power consumption of the regulators, and how I can better refine this before the custom PCB is ordered.

I have already been able to see connections between this project and my professional life. At my internship this past summer at Globus Medical I was able to use these types of board level design skills to work on a navigation board, ensuring that my board handled issues such as power line placement, sensitivity and ratings of components used, and mechanical considerations of size and cabling. Being able to quickly prototype and test my ideas has been especially helpful this semester and I know will continue to help me in my future career endeavors.

D. Christian

My main responsibility for our project is developing our website for rendering the stored data on. I have had a little experience with basic HTML and CSS programming before, but this project has really helped me flush out my understanding. I have also learned a lot of java script and python for this project. Assisting Anthony in developing some of the back has really taught me a lot about the Microsoft Azure web services.

I have also learned a lot about implementing sensors and custom hardware with the microcontroller. Spending time in the lab while helping trouble shooting the micro-controller, analog digital converter, and sensors has really flushed out my ability to use the equipment available in lab.

As a professional, as much as I do enjoy the firmware and hardware side of the project, I mainly see myself working in a software development. I think many of the skills I have learned will apply to that field well.